HABILITATION À DIRIGER DES RECHERCHES
DE
L'UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE (USTL)

HABILITATION À DIRIGER DES RECHERCHES

Spécialité : **Mathématiques Appliquées**

présentée par

Bruno Michel Jedynak

Le jeu des 20 questions et autres contributions en statistique mathématique et computationnelle

Rapporteurs :	M.	Patrick	**Bouthemy**	INRIA Rennes
	M.	Alfredo	**Hero**	University of Michigan
	M.	Alain	**Trouvé**	ENS Cachan

Soutenue le **27 Mai 2014** devant le jury composé de

M.	Christophe	**Biernacki**	USTL	Président
M.	Patrick	**Bouthemy**	INRIA Rennes	Rapporteur
M.	Alfredo	**Hero**	University of Michigan	Rapporteur
M.	Alain	**Trouvé**	ENS Cachan	Rapporteur
M.	Avner	**Bar-Hen**	Université Paris V	Examinateur
M.	Mohamed	**Daoudi**	Telecom Lille	Examinateur

À ma femme, Nathalie.

Contents

Remerciements

Merci à Nathalie, ma femme, et la mère de mes enfants, je te dédie cette habilitation.
Merci à mes enfants: Laura, Leonard et Jacob.
Merci à mes parents: Sylvie et Pierre. Merci à Mamie.
Donald Geman, merci pour avoir eu confiance en moi depuis le premier jour.
Merci à la petite équipe qui s'est organis'ee pour m'amener à soutenir cette habilitation: Nathalie Jedynak, Avner Bar-Hen et Pierre Jedynak.
Merci à mes rapporteurs: Alfredo Hero, Patrick bouthemy, et Alain Trouvé. Merci a Christophe Biernacki, mon garant, Avner Bar-Hen et Mohamed Daoudi, mes examinateurs.
Merci a ceux qui m'ont accueilli et m'ont aidé à développer mon travail: Donald Geman, Marie-Claude Viano, Jean-Louis Bon, Yali Amit, Daniel Naiman, Michael Miller, Jerry Prince et Avner Bar-Hen.
Je souhaite aussi dire un grand merci a mes co-autheurs: Francois Fleuret, Yali amit, Mohamed Daoudi, Mohamed Obeid, Huicheng Zheng, Yali Amit, Sanjeev Khudanpur, Ali Yazgan, Damianos Karakos, Craig Stark, Hailiang Huang, Joel Bader, Tilak Ratnanather, Neeraja Penumetcha, Andre Levchenko, Joshuah Vogelstein, Liam Paninski, Stephanie Davis, Sanjay Jain, Camille Vidal, Martin Pomper, Bill Bishai, Raphael Sznitman, Laurent Younes, Peter Frazier, Li Chen, Graham Beck, Andrew Lang, Bo Liu, elyse Katz, Yanwei Zhang, Brad Wyman, David Raunig, Brian Caffo, Jerry Prince, Pierre Jedynak, John Bogovic, Bennett Landman, Sarah Ying, Russ taylor, richa Rogerio, Gregory Hager, Pascal Fua, Alexander Crits-Christoph, Jocelyne diRuggiero, Yulia Gel, Murat bilgel, Susan Resnick, Heeyeon Im, Jonathan Flombaun, Ehsan Jahangiri, Erdem Yoruk, Rene Vidal, Saumya Gurbani, et Bahman Afsari.
Enfin, merci à mes professeurs inoubliables: Madame Soulas (cours préparatoire à Suresnes), Monsieur Buffet (septième à Suresnes), Monsieur Audigier (première à Neuilly), Jacques Roubaud (première année de DEUG a Nanterre), Robert Azencott, Adrien Douadi (license a Orsay), Yehuda Rav, Pascal Massart , Jean Bretagnolle (maîtrise à Orsay), Marie Duflo et Christian Leonard (DEA Orsay).

Chapter 1

Resumé en français

Mon travail de recherche depuis l'obtention de ma thèse de doctorat s'organise autour des thémes ci-dessous.

1.1 Le “Jeu des 20 questions" et ses applications

Le mécanisme d'acquisition d'information, par exemple la localisation d'un object dans une image est considéré d'un point de vue Bayésien. De manière éventuellement séquentielle et adaptative, des régions de l'image sont considérées, des fonctions des pixels sont calculées, et la distribution sur la position de l'object est remise à jour. J'ai étudié des stratégies optimales pour l'espérance de l'Entropie de Shannon sur la position de l'object après un nombre fixé d'itérations. J'ai aussi contribué au développement algorithmique de cette méthodologie pour la détection de visages, le suivi d'outils chirurgicaux dans des séquences d'images et le contrôle d'un microscope électronique.

Publications: [7, 1, 8, 9, 33, 18, 20, 35, 6, 32, 34, 41].

1.2 Les méthodes de maximum d'entropie sur la moyenne

Je propose dans ce travail une méthode originale pour l'estimation des paramètres $p_1, \ldots, p_k$ d'une loi Multinomiale dans la situation où le nombre de paramètres k, ainsi que le nombre d'observations n sont grands. Par exemple, $k = 100.000$ et $n = 1.000.000$. Dans un premier temps, un ensemble de lois Multinomiales qui sont "proches" des observations sont identifiées. Dans un second temps, je sélectionne dans cet ensemble la loi Multinomiale qui est la plus proche (Kulback) d'une loi choisie par défaut. Cette méthode est utilisée pour estimer une probabilité sur les mots de la langue anglaise. D'autre part, j'ai utilisé et evalué des méthodes d'estimation de paramétres de mesures de Gibbs et j'ai appliqué ces méthodes à la détection de la peau humaine dans des images en couleur.

Publications: [22, 23, 24, 25, 44, 43, 17]

1.3 Les méthodes statistiques en imagerie

J'ai proposé une formulation originale pour le problème de détection d'amers en imagerie médicale basée sur l'apprentissage statistique. J'ai formulé le problème en terme d'estimation Bayésienne d'une déformation. J'ai évalué cette méthode pour des images de résonnance magnétique du cerveau. Par ailleurs, j'ai proposé une méthode de mise en correspondance de formes adaptée a l'imagerie médicale des poumons pour l'étude de la tuberculose.
Publications: [39, 15, 12, 14, 13, 38, 5, 31, 29, 37].

1.4 Les méthodes statistiques pour l'étude clinique de la maladie d'Alzheimer

La base de données ADNI (Alzheimer's Disease Neuroimaging Initiative) est conçue pour étudier la maladie d'Alzheimer (AD). Cette base comporte plusieurs milliers de sujets observés pendant plusieurs années avec de multiples biomarqueurs incluant l'imagerie structurale et fonctionnelle, le comptage de protéines et de nombreux tests neurologiques. J'ai proposé une méthode statistique permettant de combiner des biomarqueurs hétérogènes afin de fournir un indice ou score d'Alzheimer pour chaque sujet. J'ai contribué ainsi à décrire le processus évolutif d'AD.
Publications: [26, 27].

1.5 Travail collaboratif en ingénierie biomédicale

Je regroupe ici des travaux variés où j'ai contribué en temps qu'expert en probabilités, statistique et imagerie.
Publications: [19, 11, 40, 2, 30, 3].

Chapter 2

Summary of research activities

2.1 Pattern theory, Bayesian modelling and computer vision

2.1.1 Pattern theory

Pattern Theory was initiated by Ulf Grenander about thirty years ago. The aim is to analyze patterns from a statistical point of view in all "signals" generated by the world, whether they be visual, acoustical, textual, molecular (e.g., DNA strings), neural, etc. Patterns are described using hidden variables, together with their probability distributions, whereas signals, or relevant functions of the signals, are modeled conditionally on the hidden variables. In principle, the detection of patterns in noisy and ambiguous samples can then be achieved by the use of Bayes's rule. An overview of pattern theory as a mathematical theory of perception was presented during the International Congress of Mathematics in 2002; see [28].

There are enormous difficulties in realizing the pattern theory program. Initially, I was inspired by problems arising in computer vision. Indeed, computer vision offers an overwhelmingly rich source of challenging questions. How can a system identify an object in an image in the presence of clutter and occlusion? Locally, the existence of the object is always ambiguous, whereas globally it seems unambiguous! Humans identify faces and skin very efficiently from still images. A trained radiologist can precisely identify brain structures from magnetic resonance images. Can one design efficient computer vision systems that replicate these capabilities ?

In each situation, one starts with data and seeks concepts in the sense of interpretations. It is then necessary to use a quantitative measure of information that can be applied to a large class of objects. Shannon information theory provides some of the theoretical foundations. However, the classical goals of information theory – coding and compression – are not the same as in computer vision, where the questions are of a statistical nature, mostly estimation. There are many interesting connections between statistics and information theory. One of particular interest and simplicity is related to the problem of Bayesian classification.

2.1.2 Bayesian analysis and conditional entropy

In Bayesian classification, there is a finite set of objects or interpretations of interest, denoted $\mathcal{Y}$. The data, often multidimensional, lives in a set $\mathcal{X}$. Assuming a suitable probability structure and a binary loss function, the best guess for the object, having observed a data point $x \in \mathcal{X}$, is $\hat{Y}(x)$, the mode of the posterior probability $P(Y|X = x)$. Now the expected error of this classifier, denoted e^*, is very closely related to the conditional entropy $H(Y|X)$ (in base 2) which measures the expected amount of information that X provides about Y. On one hand, a concavity argument provides $e^* \leq cH(Y|X)$; for some positive constant $c(|\mathcal{Y}|)$ and, on the other hand, Fano's inequality yields a reciprocal bound: $H(Y|X) \leq H(e^*) + e^* \log(|\mathcal{Y}| - 1)$. So, it is sufficient and necessary to control $H(Y|X)$ in order to control e^*. The situation generalizes to other Bayesian loss functions as the quadratic loss or the L_1 loss as long as Y is countable. In each case, it is sufficient and necessary to control $H(Y|X)$ in order to control the risk of the Bayesian estimator. In the case of regression, when Y is a continuous random variable or vector, the situation is different. In the case of the quadratic loss, controlling the conditional differential entropy $H(Y|X)$ is necessary but not sufficient. Indeed, using the maximum entropy principle, Th. 17.2.3 in [4] and Jensen inequality,

$$E\left[||Y - E[Y|X]||^2\right] \geq \frac{d}{2\pi e} 2^{\frac{2}{d}H(Y|X)} \tag{2.1}$$

where Y is of dimension d. As a consequence, a large value of the conditional entropy $H(Y|X)$ implies a large Bayesian quadratic risk (left side of (2.1)). However, as small value for the conditional entropy does not necessarily implies a small value for the Bayesian quadratic risk. We provide two examples. First, consider the situation where the conditional distribution of Y given $X = x$, for each x, is a mixture, with equal weights, of two Normal distributions with different means but same variance σ^2. Choosing a sequence of values σ_n^2 such that $\lim_{n\to+\infty} \sigma_n^2 = 0$, we have $\lim_{n\to\infty} H(Y|X) = -\infty$ but the Bayesian quadratic risk remains away from zero. In this case, some regularity of the posterior distribution of Y given $X = x$ would be needed in order to guaranty a reciprocal in (2.1). As a second example, consider the case where Y is two-dimensional, i.e $Y = (Y_1, Y_2)$ but X provides information on Y_1 only. In this case $H(Y|X) = H(Y_1|X, Y_2) + H(Y_2|X) = H(Y_1|X, Y_2) + H(Y_2)$ where the first equality is the chain rule for the conditional entropy and the second comes from the independence of X and Y_2. Now, if $H(Y_1|X, Y_2)$ is arbitrarily small (i.e negative and large in absolute value), so is $H(Y|X)$. However, $H(Y_2|X) = H(Y_2)$ which means that the component of the Bayesian quadratic risk associated with the second coordinate (i.e Y_2) is not controlled.

In what follows, we will consider algorithms which sequentially reduce the Shannon entropy of the posterior.

2.2 20 questions

This work was done at JHU in collaboration with Raphael Sznitman, Peter Frazier and Han Weidong. It pursue ideas originally presented by Donald Geman.

2.2.1 Iterative questioning as a model for perception

Let us consider computer vision, or machine perception, as an efficient mechanism aimed at reducing uncertainty; as do others. A concrete example is as follows: consider the task of locating a front facing standard size face within an image, this location being by definition fully characterized by the pixel location of the nose. As in Bayesian statistics, this location is described by a random variable, which distribution over the set of pixel locations has large entropy. We assume that a collection of unit cost questions are available. Each question is parameterized with the coordinates of a sub-image. The answer which is a numerical value is obtained by computing a function of the image values within this sub-image and is modeled as a noisy answer to the question "does the face belong to this sub-image?" Which sub-images should then be chosen and in which order such that one would detect the face while minimizing the average or worse case number of queries? More formally, we now describe the basic theoretical framework of "20 questions" in which our research is conducted:

1. There is some unknown target, or collection of targets, that we would like to locate. Let Y denote the location of this (these) target(s), and let $\mathcal{Y}$ denote the set of values that Y can take. For example, Y might be the pose of an object in a scene, as recorded in an image (we call this task "object localization").

2. Before the search begins, there is a Bayesian prior distribution p_0 on the unknown object or quantity. Thus, we model Y as being a random variable or vector drawn at random from the distribution p_0. In object localization, for example, this prior incorporates information that certain poses of an object are more common than others.

3. We may ask questions, perform tests, or search in particular locations, to gather information. However, the answers to these questions may be obscured by noise.

   Formally, we model a question as a subset $A \subset \mathcal{Y}$, whose truthfull response is given by $Z(A) = 1_A(Y)$, where 1_A is the indicator function of the set A. We do not observe $Z(A)$ directly, instead observing a noisy version $X(A)$ of $Z(A)$.

   For example, in object localization, A would be a sub-image, and $X(A)$ would be obtained by running a statistical filter over the image whose output distribution depends on whether $Y \in A$ or not.

   We ask a sequence of questions, $A_1, \ldots, A_N$ in this way, observing answers $X_n = X(A_n), n = 1, \ldots, N$, where the choice of question A_{n+1} depends on all the previous questions and their responses, $A_1, X_1, \ldots, A_n, X_n$.

4. Once questioning ceases, we estimate the location of Y as well as possible, given the information that we have collected.

Our ability to perform the final estimation task well depends critically upon the questions that we have asked. If we ask poor questions, then we will need a much larger number of questions to perform well. If we ask good questions, then **we can provide high-quality estimates with limited data and limited time.**

Thus, **the central goal of this research is to design algorithms for adaptively deciding which questions to ask next**, to provide optimal or near-optimal average-case performance under the prior.

2.2.2 Main results

As questions are chosen, asked and answered, the prior distribution p_0 over the target Y is updated according to Bayes rule, providing $p_1, p_2, \ldots$. Notice that even when there is no noise in the answers and when the questions are chosen deterministically, p_n is in general a random sequence, as a function of Y. We notate $\mathcal{F}_n$ the history of the first n questions and answers under a valid policy, that is a policy for which the choice of the next question depends only on the questions and answers obtained so far. We consider next the $\mathcal{F}_n$ measurable sequence H_n which is the entropy of p_n. It is convenient to use the following notations: $H_n = H(p_n) = H(Y|\mathcal{F}_n)$. Then,

$$\begin{aligned} E\left[H_{n+1}|\mathcal{F}_n\right] &= H\left(Y|\mathcal{F}_n, X_{n+1}\right) && (2.2)\\ &= H\left(Y, X_{n+1}|\mathcal{F}_n\right) - H\left(X_{n+1}|\mathcal{F}_n\right) && (2.3)\\ &= H(Y|\mathcal{F}_n) - \left(H(X_{n+1}|\mathcal{F}_n) - H(X_{n+1}|\mathcal{F}_n, Y)\right) && (2.4)\\ &= H_n - I\left(Y, X_{n+1}|\mathcal{F}_n\right) && (2.5)\\ & && (2.6) \end{aligned}$$

The right hand side of (2.2) is the conditional entropy of Y given the random variable X_{n+1} and the history $\mathcal{F}_n$. Equations (2.3)and (2.4) are direct applications of the chain rule for the joint entropy. Equation (2.5) comes from the definition of $I\left(Y, X_{n+1}|\mathcal{F}_n\right)$, the mutual information between Y and X_{n+1} given the history $\mathcal{F}_n$. Since the mutual information is a non negative quantity, we see from (2.5) that the entropy process H_n decreases in average, under any valid policy. Now, recall that we defined X_{n+1} as a noisy version of Z_{n+1} which itself is a binary variable: the indicator of some set $A_{n+1} \subset \mathcal{Y}$. More precisely, we assume that given the history $\mathcal{F}_n$, $Y \mapsto Z_{n+1} \mapsto X_{n+1}$ is a Markov chain. This in turns implies,

$$\begin{aligned} E\left[H_{n+1}|\mathcal{F}_n\right] &\geq H_n - I\left(Y, Z_{n+1}|\mathcal{F}_n\right) && (2.7)\\ &\geq H_n - H(Z_{n+1}|\mathcal{F}_n) && (2.8)\\ &\geq H_n - 1 && (2.9) \end{aligned}$$

where (2.7) is the *data processing* inequality, Th. 2.8.1 in [4]. (2.8) is obtained by bounding from above the mutual information and (2.9) since X_{n+1} is a binary random variable. Now, taking the expected value on both sides of (2.9), we obtain

$$E[H_n] \geq H_0 - n \qquad (2.10)$$

which gives its name to the game of 20 questions. Indeed, consider a situation where Y is uniformly distributed over the first 2^{20} (which is close to $1,000,000$) positive integers. How many binary questions are needed in average to guess Y, assuming truthful answers? We plug $E[H_n] = 0$ in (2.12) obtaining $n \geq H_0 = \log_2 2^{20} = 20$. Moreover, this lower bound is achievable using for example the dichotomy policy.

Another situation of interest occurs when the noise process X_n is a memoryless channel, that is when $X_1, \ldots, X_n$ are conditionally independent given Y. We have shown in [18] (Chapter 3) that in this case the bound in (2.9) can be improved to

$$\mathrm{E}\left[H_{n+1}|\mathcal{F}_n\right] \geq H_0 - C \tag{2.11}$$

where $0 \leq C \leq 1$ is a *channel capacity* which can be computed explicitly as function of the noise model.

As was noted in [36], combining (2.11) with (2.1),

$$\mathrm{E}\left[\|Y - \mathrm{E}\left[Y|\mathcal{F}_n\right]\|^2\right] \geq \frac{d2^{2\frac{H_0}{d}}}{2\pi e} 2^{-2n\frac{C}{d}} \tag{2.12}$$

provides a lower bound on the quadratic efficiency of any policy.

Equation (2.5) suggest a specific policy, namely the *entropy pursuit* which consist in, having observed a specific history $\mathcal{F}_n$, choosing at step $n+1$

$$A_{n+1} = \arg\max_A I\left(Y, X_{n+1}(A)|\mathcal{F}_n\right) \tag{2.13}$$

We show in [18] that under certain conditions, this policy is actually globally optimal in minimizing the expected entropy $\mathrm{E}[H_N]$ over an horizon of N questions for any $N \geq 1$. We further present the dyadic policy which is also optimal but not adaptive i.e. all the questions can be asked and answered in parallel. Now, the performance of an optimal policy is remarkable. Indeed, under an optimal policy

$$\mathrm{E}[H_n] = H_0 - nC \tag{2.14}$$

The expected entropy of the target decreases linearly with the number of questions. In other words, the same amount of information (measured by the Shannon entropy) is obtained in average at each step. As long as C is not too small, this is a motivating result for application and further theoretical explorations. Chapter 3 presents this theory and Chapter 4 present an extension to multiple targets. Note that the later is a preliminary draft. Section 2.3 and Chapters 5 and 6 provide applications of this framework for various tasks in computer vision.

I plan to further study problems in optimal search, detection, and interrogation in a Bayesian decision-making framework, developing search algorithms with both theoretical guarantees and strong empirical performance. Using the 20 questions mathematical framework I plan to study specific problems in computer vision focusing on target tracking and detection, in screening, in simulation optimization, in machine learning and in experimental psychology for understanding visual search.

2.3 Application of iterative questioning to computer vision

2.3.1 Road Tracking

This work was done while I was a PhD student under the supervision of Donald Geman. As such it not part of the HDR. However, since it has been important in framing future work, it is briefly described here.

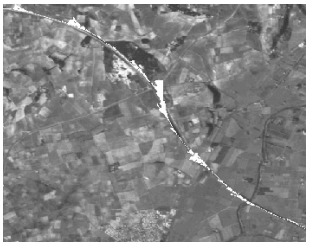


Figure 2.1: An algorithm for tracking roads

We present a new approach for tracking roads from satellite images, and thereby illustrate a general computational strategy ("active testing") for tracking 1D structures and other recognition tasks in computer vision. Our approach is related to work in active vision on "where to look next" and motivated by the "divide-and-conquer" strategy of parlor games such as "Twenty Questions." We choose "tests" (matched filters for short road segments) one at a time in order to remove as much uncertainty as possible about the "true hypothesis" (road position) given the results of the previous tests.The tests are chosen on-line based on a statistical model for the joint distribution of tests and hypotheses. The problem of minimizing uncertainty (measured by entropy) is formulated in simple and explicit analytical terms. To execute this entropy testing rule we then alternate between data collection and optimization: At each iteration new image data are examined and a new entropy minimization problem is solved (exactly), resulting in a new image location to inspect, and so forth. We report experiments using panchromatic SPOT satellite imagery with a ground resolution of ten meters: Given a starting point and starting direction, we are able to rapidly track highways in southern France over distances on the order of one hundred kilometers without manual intervention. Road tracking consists of identifying a road in a remotely sensed image, starting with a pixel on the road in the image and a direction, both manually selected; see [7].

2.3.2 Outlier Detection and Asymptotic Properties of the Road Tracking Algorithm

This work was initiated at USTL and continued at JHU in collaboration with Damianos Karakos.

Our motivation for this paper originates in the work on road tracking described above. Below a certain clutter level, that algorithm could track a road accurately, but suddenly, with increased clutter, tracking would become impossible.

We consider the problem of detecting a target in the presence of background clutter. We study an ultra-simplified model, introduced in [42], where a phenomenon of phase transition is observed: there are $M + 1$ sequences of independent discrete random variables, each sequence being of length N, and all sequences have components with the same probability mass function p_0 except for one sequence, the target, whose elements have probability mass function p_1. We focus on asymptotic bounds of performance, and we show that the error of the maximum likelihood estimator for the target converges to 0 or to 1, depending on the behavior of the fundamental quantity $M2^{-N\,D(p_1,p_0)}$, where $D(.,.)$

is the Kulback-Leibler divergence. Moreover, we describe a target detector for the case where p_0 and p_1 are unknown, and we prove that it has the same phase transition behavior as in the case of known distributions. See [20] and Chapter 9

2.3.3 Face Detection

This work was done during my post-doc at the University of Chicago in collaboration with Yali Amit.

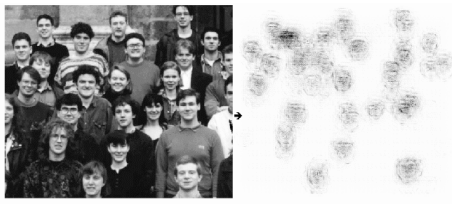


Figure 2.2: Face detection. The amount of processing as a function of the location in the image

Face detection consists in identifying the locations of the faces, if any, in an image. It is a necessary step for performing face recognition from unconstrained images. Here the class variable takes only two values corresponding to the presence or absence of a face in a sub-image. This apparent simplicity hides a complex mixture of situations when a face is present, corresponding to instances of pose, identity and lighting, not to mention the enormous variations in the nature of the cluttered background. It is actually surprising that any statistically meaningful performance could be achieved. Detection is done in two stages: (i)"focusing", during which a relatively small number of regions-of-interest are identified, minimizing computation and false negatives at the temporary expense of false positives; and (2) "intensive classification", during which a selected region-of-interest is labeled face or background based on multiple decision trees and normalized data. In contrast to most detection algorithms, the processing is then very highly concentrated in the regions near faces and near false positives, as can be seen in Figure 2.2. See [1] and Chapter 5. Unfortunately, such a computational design does not emerge naturally from greedy entropy. We studied this phenomenon in a more general context as described in the following subsection.

2.3.4 Global vs Greedy Procedures for Entropy Reduction

This work was done at USTL in collaboration with Donald Geman.

The construction of classification trees is nearly always top-down, locally optimal, and data-driven. Such recursive designs are often globally inefficient, for instance, in terms of the mean depth necessary to reach a given classification rate. We consider statistical models for which exact global optimization is feasible, using dynamic programming, and thereby demonstrate that recursive and global procedure may result in very different tree graphs and overall performance. Here is a toy example that was motivated by the work on face detection. There are two classes. One noted a is "object" and the other noted

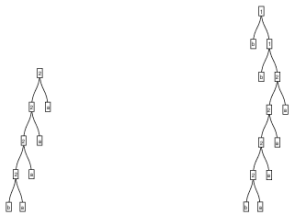


Figure 2.3: Left: Locally optimal tree. Right: Globally optimal tree. The error rates are the same but the *mean* depth of the global tree is smaller.

b is "background", with prior probabilities $p(a) = 10^{-4}$ and $p(b) = 1 - 10^{-4}$. There are two types of tests, X_1 and X_2. X_1 has a 0 false positive rate, i.e., keeps all the background together, but has false negative rate 0.5. X_2 has a 0 false negative rate, i.e., loses no objects, but 0.5 false positive rate. These tests are assumed to be repeatable, the sequence of outcomes being independent conditional on the class. Figure 2.3.4 shows the tree obtained by the greedy entropy reduction as well as a globally optimal tree with maximum depth 6. The error rate for these trees are approximatively the same but the mean depth is about 4 for the greedy one, and about 2.5 for the optimal one. At the same mean depth, the optimal strategy may have an error rate ten times smaller than the greedy strategy. See [8] and Chapter 7

2.4 Automatic Landmark Detection from Brain MRI

This work is being conducted at JHU in collaboration with Camille Vidal, born Izard.

An anatomical landmark in the brain is a well-defined point of the anatomy of the brain. Locating a landmark in a magnetic resonance brain image, or "landmarking," consists of selecting a particular voxel in the image, corresponding to the anatomical landmark in the imaged brain. This voxel, like an anchor, is a precious piece of information for measuring and registering brain structures. Landmarking can be a tedious manual procedure, expensive and time consuming. It might be error prone, difficult to assess, and dependent on the scanner and on the landmarker. We have developed a generic algorithm that permits one to partially automate the landmarking process. The algorithm has two components. One is an off-line procedure, the other is on-line. The former is a system that estimates the parameters of a probabilistic model from a training set of landmarked images using the Estimation Maximization (EM) algorithm. The later inputs an image as well as the parameters previously estimated and outputs a tentative location for the landmark as well as a covariance metric that assesses the remaining uncertainty. The selected location can then be validated or corrected manually. The probabilistic model has two components corresponding to photometry and the geometry. The former is a mixture of Gaussian distributions whereas the later is a probabilistic model over sets of deformations. We have considered various classes of affine deformations as well as small nonlinear deformations using kernels.

An instantiation of the method for detecting the apex of the Head of the Hippocampus

(HoH) is shown in Figure 2.4. See [39, 15, 12, 14, 13] and Chapter 10.

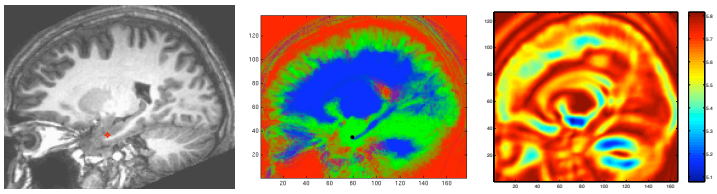


Figure 2.4: **Left:** A sagittal slice of the brain. The Apex of the head of the Hippocampus (HoH) is shown in red. **Center:** Probabilistic model predicting the probability for matter type given the location of the HoH. Red channel: cerebrospinal fluid. Green channel: grey matter. Blue channel: white matter. **Right :** Expected variance reduction in localizing the HoH according to the learned probabilistic model. Most informative voxels are in blue, least informative voxels are in red

2.5 Maximum Entropy Modeling and Small Sample Statistics

Maximum Entropy Modeling is a statistical modeling methodology aimed at selecting a probability distribution given a data set. It is a two-step procedure. In the first step, one chooses a subset of probability distribution that is consistent with the data. Typically, one constrains the mean of certain functions of the data, also called features, to agree with the empirical mean derived from the data at hand. In the second step, one chooses a *reference* probability distribution or positive measure. Then, the "closest" distribution to the reference, within the subset defined in the first step, is selected. For example, if the reference is the Uniform measure and the Kulback-Leibler distance is used to define "closest", this procedure amounts to selecting the distribution with maximum entropy under constraints.

The whole procedure might be viewed as an alternative to Bayesian modeling since one is not obliged to choose a whole prior over a set of distributions but rather a single element, the reference, together with a set of features.

This method was pioneered in statistical mechanics where the object of study is a very large set of interacting particles. The "microstate" is defined as the collection of the states of the particles. It is to be modeled. However, the set of microstates is so large that it cannot be directly modeled from observing a few instances. Alternatively, one has access to "macrostates". These are quantities that are averaged over the set of particles. Choosing the maximum entropy model among those which replicate the observed macrostate values leads to the important class of Gibbs models, or Markov Random Fields.

This approach was shown to be of great practical importance in low level imaging in [10]. The use of large sets of natural images has led, using MEM, to the construction of models for textures [45].

2.5.1 Models for the Texture of Skin

This work was done in collaboration with Mohamed Daoudi and Huicheng Zheng at USTL, while Mohamed and I were co-supervising Huicheng' PhD thesis.

Figure 2.5: Three models for the classification of skin pixels

In order to classify pixels as "skin" or "non skin", we have experimented with Maximum Entropy Modeling with tree approximations to Markov Random Fields. Indeed, when the underlying graph is a tree, the optimization procedure required to estimate the parameters of the model can be tackled by an efficient procedure, already used in natural language processing, known as iterative scaling. Moreover, classification of pixels as *skin* or *not skin* is achieved through an efficient combinatorial optimization procedure, closely related to dynamic programming and known as belief propagation. We build a sequence of three models by adding features one at a time. The observed statistics come from a collection of hand-segmented images. The first model imposes constraints on one-pixel color histograms given "skin" and given "non-skin" . The solution is a baseline model in which colors are conditionally independent. This model is well-known among practitioners. The baseline model is certainly too weak and does not take into account the fact that skin zones are made of large regions with regular shapes. Hence, in the second model, we add constraints on the distribution of neighboring labels (skin or not-skin) in order to smooth the solution. Finally, a color gradient is included in building the third model. Figure 2.5.1 depicts examples of the resulting segmentations. The color is proportional to the posterior probability for skin. State-of-the-art performance is reported. See[24, 44] and Chapter 11.

2.5.2 MEM in the Small Sample Setting and Language Modeling

This work was done at Johns Hopkins University, in collaboration with Sanjeev Khudanpur, Damianos Karakos and Ali Yazgan.

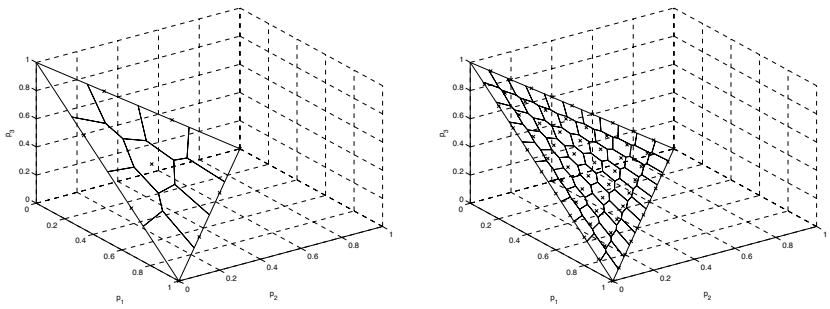


Figure 2.6: Maximum Likelihood Sets for $k = 3$ outcomes. Left: $n = 3$ observations. Right: $n = 10$ observations.

There are challenging applications in statistics where the number of samples is small compared to the dimensionality of the data. If one wants to adapt the MEM approach to these situations, one has to take into account the natural variability of the empirical mean of the features around their expectation in order to define a set of distributions consistent with the data. How can this be done in a systematic way? An example arises in natural language modeling where one needs to define the probability for the next word in a sequence. Even, more basically, one needs to estimate the probability of appearance of a word, independently of the past words. Assume there are $k = 50,000$ English words in the dictionary and that a corpus of $n = 1,000,000$ words from the Wall Street Journal is available. Typically, $13,000$ words are not present in the corpus and $13,000$ are seen only once. This is a small sample situation. When estimating conditional distributions, the small sample effect is even more severe. The simplest features in this context are the indicator functions for a presence of a word. There are k such features. However, the set of distributions over words that replicate the observed frequencies for each word is reduced to a single distribution – the empirical measure, or *type* which put zero mass on about $1/4$ of the vocabulary. We propose a parameter-free method to release the hard constraint on the word frequencies. We choose the set of distributions on words that make the observed frequencies more likely than any other with the same sample size. This defines a closed convex polyhedron in the space of distributions that we call the Maximum Likelihood Set; see Figure 2.5.2. We then choose the one in this set closest in Kulback-Leibler divergence to the Zipf distribution, the natural prior in this context. The obtained estimator is shown to be competitive with state-of-the art methods, see [25] and Chapter 8

2.6 Image registration for studying the progression of Tuberculosis in a preclinical study

This work was done at JHU in collaboration with Sanjay Jain, Laurent Younes, Camille Vidal and Saumya Gurbani.

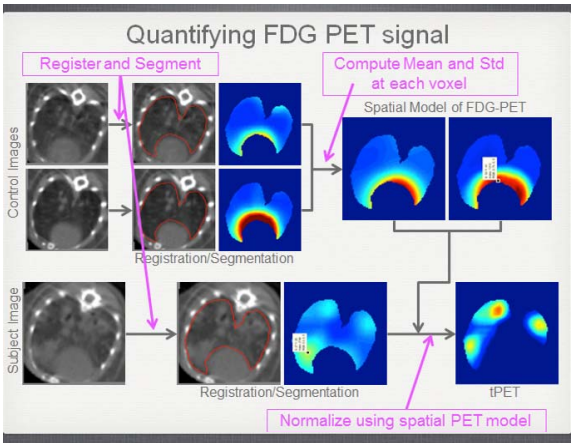


Figure 2.7: Registration pipeline

Many techniques have been proposed to segment organs from images. However the

segmentation of diseased organs remains challenging and frequently requires a sizeable amount of user interaction. The challenge consists of segmenting an organ while its appearance and its shape vary due to the presence of the disease in addition to individual variations. We propose a template registration technique that can be used to recover the complete segmentation of a diseased organ from a partial segmentation. The usual template registration method is modified in such a way that it is robust to missing parts. The proposed method is used to segment Mycobacterium tuberculosis (TB) infected lungs in CT images of experimentally infected mice, [39, 38]. This allows to measure precisely the inflammation generated by TB, [5, 31].

2.7 Modelling the time course of neurodegenerative diseases

This work was done at JHU in collaboration with Pierre Jedynak, Jerry Prince, Brian Caffo, Yulia Gel, Bo Liu, Andrew Lang, Runze Tang and Xhou Ye.

Alzheimer's disease (AD) is a dreadful neuro-degenerative disease. The Alzheimer's disease neuroimaging initiative (ADNI) is a publicly available clinical dataset including subjects diagnosed with AD dementia, mild cognitive impairment (MCI), and normal controls. In ADNI, hundreds of measurements, including clinical, cognitive, biochemical, genetic and imaging are available at baseline and longitudinally ($M =\sim 10$ visits) for more than $N = 900$ subjects. The AD trajectory of a subject is then a curve in the Euclidian space of dimension K (number of measurements) out of which M points are partially observed. The ADNI dataset provides N such curves. Since the subjects in the ADNI dataset are developing the same disease, we hypothesize that a few continuous latent variables explain the collection of measurements observed at successive visits, modulo certain individual characteristics. Specifically, we hypothesize that the curves defined above lie in a manifold of dimension $L < K$ embedded in $\mathbb{R}^K$ which we refer to as the *disease space*. We further hypothesize that individual characteristics, genetic or environmental, characterize the trajectory of a subject within this manifold as function of subject age. This point of view leads to the following compositional model:

$$y_{ijk} = f\left(g(t_{ij}; \theta_i); \rho_k\right) + \epsilon_{ijk} \tag{2.15}$$

Where y_{ijk} is the measurement with index k for subject i at visit j, t_{ij} is the age of subject i at visit j. $t \mapsto g(t, \theta_i)$ maps the age of a subject to a point in the disease manifold $\subset \mathbb{R}^L$, and $x \mapsto f(x, \rho_k)$ maps a point in $\mathbb{R}^L$ to an observable point in $\mathbb{R}^K$. θ_i is a collection of subject dependent parameters characterizing the trajectory of a subject within the disease manifold while ρ_k is a collection of measurement dependent parameters characterizing the disease space. ϵ_{ijk} is the residual noise. Scientific questions about AD can be reformulated in the language of the model in equation (2.15) and thus can be answered using parameter estimation and testing. "Do all subjects follow the same disease progression?" is equivalent to "Does $L = 1$?". "Does the Apoe4 genotype affects disease onset?" can be tested by measuring the correlation between θ and the Apoe4 genotype. "Does hippocampus volume change earlier in the disease progression than Tau mediated neuronal injury?" can be tested in using L=1 and comparing the fitted parameters for these 2 measures. The currently most accepted model of disease progression known as

"Cliff Jack curves", see [16], corresponds to a choice of K=5 measurements, L=1, and sigmoid functions for f. In [21], Chapter 12, we have fitted a closely related model with $K = 7$, $L = 1$, sigmoid functions for f and linear functions for g. This model allows explaining 62% of the variance of these 7 measurements over the whole cohort. This is the current state of the art. Also, one of the findings of this analysis is the discovery that the Rey auditory verbal test, 30 minutes recall, a measure which was not considered in [16], is an early indicator of AD progression [27]. Another outcome of this statistical analysis is an AD progression score for each subject in the ADNI cohort. This score, which is computed from a heterogeneous collection of measures, provides a continuous characterization of the disease stage of each subject. See figure 2.8.

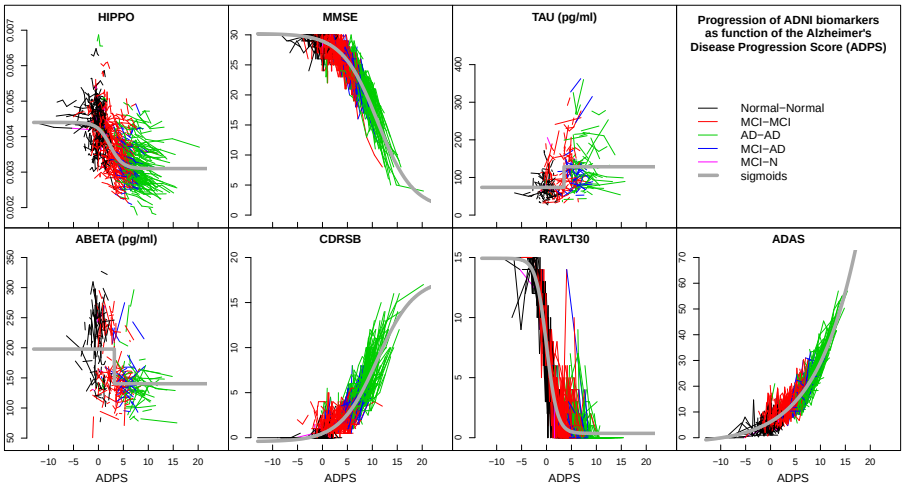


Figure 2.8: The values of seven biomarkers, measured at all visits of all ADNI subjects, are plotted on the normalized AD progression score (ADPS). Each connected polyline represents the consecutive visits of a single subject, and each line segment is colored according to the subject's clinical diagnoses between visits (see legend). The gray curves are the sigmoid functions representing the fitted behavior of each biomarker in the normalized space. (Reproduced from [21])

I plan to further develop this research and identify (Ulf Grenander's) patterns in neurodegenerative diseases from large and heterogeneous data.

Bibliography

[1] Yali Amit, Donald Geman, and Bruno Jedynak. Efficient focusing and face detection. In *Face Recognition*, pages 157–173. Springer Berlin Heidelberg, 1998.

[2] John A Bogovic, Bruno Jedynak, Rachel Rigg, Annie Du, Bennett A Landman, Jerry L Prince, and Sarah H Ying. Approaching expert results using a hierarchical cerebellum parcellation protocol for multiple inexpert human raters. *NeuroImage*, 64:616–629, 2013.

[3] HoJung Cho, Henrik Jönsson, Kyle Campbell, Pontus Melke, Joshua W Williams, Bruno Jedynak, Ann M Stevens, Alex Groisman, and Andre Levchenko. Self-organization in high-density bacterial colonies: efficient crowd control. *PLoS biology*, 5(11):e302, 2007.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006.

[5] Stephanie L Davis, Eric L Nuermberger, Peter K Um, Camille Vidal, Bruno Jedynak, Martin G Pomper, William R Bishai, and Sanjay K Jain. Noninvasive pulmonary [18f]-2-fluoro-deoxy-d-glucose positron emission tomography correlates with bactericidal activity of tuberculosis drug treatment. *Antimicrobial agents and chemotherapy*, 53(11):4879–4884, 2009.

[6] Peter I Frazier, Bruno Jedynak, and Li Chen. Sequential screening: a bayesian dynamic programming analysis of optimal group-splitting. In *Proceedings of the Winter Simulation Conference*, page 50. Winter Simulation Conference, 2012.

[7] Donald Geman and Bruno Jedynak. An active testing model for tracking roads in satellite images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(1):1–14, 1996.

[8] Donald Geman and Bruno Jedynak. Model-based classification trees. *IEEE Transactions on Information Theory*, 47(3):1075–1082, 2001.

[9] Donald Geman, Bruno Jedynak, et al. Shape recognition and twenty questions. 1993.

[10] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on PAMI*, 6:721–741, 1984.

[11] Hailiang Huang, Bruno M Jedynak, and Joel S Bader. Where have all the interactions gone? estimating the coverage of two-hybrid protein interaction maps. *PLoS computational biology*, 3(11):e214–e214, 2007.

[12] Camille Izard and Bruno Jedynak. Bayesian registration for anatomical landmark detection. In *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on*, pages 856–859. IEEE, 2006.

[13] Camille Izard and Bruno Jedynak. Statistical deformable model applied to anatomical landmark detection. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 444–447. IEEE, 2008.

[14] Camille Izard, Bruno Jedynak, and Craig EL Stark. Spline-based probabilistic model for anatomical landmark detection. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*, pages 849–856. Springer Berlin Heidelberg, 2006.

[15] Camille Izard, Bruno M Jedynak, and Craig EL Stark. Automatic landmarking of magnetic resonance brain images. In *Medical Imaging*, pages 1329–1340. International Society for Optics and Photonics, 2005.

[16] Clifford R Jack, Jr., David S Knopman, William J Jagust, Leslie M Shaw, Paul S Aisen, Michael W Weiner, Ronald C Petersen, and John Q Trojanowski. Hypothetical model of dynamic biomarkers of the Alzheimer's pathological cascade. *The Lancet Neurology*, 9(1):119–128, 2010.

[17] Bruno Jedynak. Blocking adult images based on statistical skin detection. *Electronic Letters on Computer Vision and Image Analysis*, 4(2):1–14, 2004.

[18] Bruno Jedynak, Peter I Frazier, and Raphael Sznitman. Twenty questions with noise: Bayes optimal policies for entropy loss. *Journal of Applied Probability*, 49(1):114–136, 2012.

[19] Bruno Jedynak and Damianos Karakos. Unigram language models using diffusion smoothing over graphs. *TextGraphs-2: Graph-Based Algorithms for Natural Language Processing*, page 33, 2007.

[20] Bruno Jedynak and Damianos Karakos. Finding a needle in a haystack: Conditions for reliable detection in the presence of clutter. *Statistics & Probability Letters*, 78(5):471–480, 2008.

[21] Bruno Jedynak, Andrew Lang, Bo Liu, Elyse Katz, Yanwei Zhang, Bradley Wyman, David Rauning, C. Pierre Jedynak, Brian Caffo, and Jerry Prince. Staging for neurodegenerative diseases: Validation with the alzheimer's disease neuroimaging initiative cohort. 2012. in preparation.

[22] Bruno Jedynak, Huicheng Zheng, and Mohamed Daoudi. Maximum entropy models for skin detection. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 180–193. Springer Berlin Heidelberg, 2003.

[23] Bruno Jedynak, Huicheng Zheng, and Mohamed Daoudi. Statistical models for skin detection. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 8, pages 92–92. IEEE, 2003.

[24] Bruno Jedynak, Huicheng Zheng, and Mohamed Daoudi. Skin detection using pairwise models. *Image and Vision Computing*, 23(13):1122–1130, 2005.

[25] Bruno M Jedynak and Sanjeev Khudanpur. Maximum likelihood set for estimating a probability mass function. *Neural computation*, 17(7):1508–1530, 2005.

[26] Bruno M Jedynak, Andrew Lang, Bo Liu, Elyse Katz, Yanwei Zhang, Bradley T Wyman, David Raunig, C Pierre Jedynak, Brian Caffo, and Jerry L Prince. A computational neurodegenerative disease progression score: Method and results with the alzheimer's disease neuroimaging initiative cohort. *NeuroImage*, 63(3):1478–1486, 2012.

[27] Bruno M. Jedynak, Bo Liu, Andrew Lang, Yulia Gel, and Jerry L. Prince for the Alzheimer's Disease Neuroimaging Initiative. A computational method for computing an alzheimer's disease progression score; experiments and validation with the adni dataset. *Neurobiology of Aging*, 2014. in press.

[28] D. Mumford. Pattern theory: the mathematics of perception. In *ICM*, 2002. http://www.dam.brown.edu/people/mumford/Papers/ICM02proceedings.pdf.

[29] N Penumetcha, Bruno Jedynak, M Hosakere, Elvan Ceyhan, Kelly N Botteron, and J Tilak Ratnanather. Segmentation of arteries in mprage images of the ventral medial prefrontal cortex. *Computerized Medical Imaging and Graphics*, 32(1):36–43, 2008.

[30] Neeraja Penumetcha, Suraj Kabadi, Bruno Jedynak, Charles Walcutt, Mokhtar H Gado, Lei Wang, and J Tilak Ratnanather. Feasibility of geometric-intensity-based semi-automated delineation of the tentorium cerebelli from mri scans. *Journal of Neuroimaging*, 21(2):e148–e155, 2011.

[31] Mrudula Pullambhatla, Jean Tessier, Graham Beck, Bruno Jedynak, Jens U Wurthner, and Martin G Pomper. [125i] fiau imaging in a preclinical model of lung infection: quantification of bacterial load. *American journal of nuclear medicine and molecular imaging*, 2(3):260, 2012.

[32] Raphael Sznitman, Anasuya Basu, Rogerio Richa, Jim Handa, Peter Gehlbach, Russell H Taylor, Bruno Jedynak, and Gregory D Hager. Unified detection and tracking in retinal microsurgery. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, pages 1–8. Springer Berlin Heidelberg, 2011.

[33] Raphael Sznitman and Bruno Jedynak. Active testing for face detection and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1914–1920, 2010.

[34] Raphael Sznitman, Aurelien Lucchi, Peter Frazier, Bruno Jedynak, and Pascal Fua. An optimal policy for target localization with application to electron microscopy. In

Proceedings of The 30th International Conference on Machine Learning, pages 1–9, 2013.

[35] Raphael Sznitman, Rogerio Richa, Russell H Taylor, Bruno Jedynak, and Gregory D Hager. Unified detection and tracking of instruments during retinal microsurgery. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(5):1263–1273, 2013.

[36] Theodoros Tsiligkaridis, Brian M. Sadler, and Alfred O. Hero III. Collaborative 20 questions for target localization. *CoRR*, abs/1306.1922, 2013.

[37] Camille Vidal, Dale Beggs, Laurent Younes, Sanjay K Jain, and Bruno Jedynak. Incorporating user input in template-based segmentation. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1434–1437. IEEE, 2011.

[38] Camille Vidal, Joshua Hewitt, Stephanie Davis, Laurent Younes, Sanjay Jain, and Bruno Jedynak. Template registration with missing parts: Application to the segmentation of m. tuberculosis infected lungs. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pages 718–721. IEEE, 2009.

[39] Camille Vidal and Bruno Jedynak. Learning to match: Deriving optimal template-matching algorithms from probabilistic image models. *International journal of computer vision*, 88(2):189–213, 2010.

[40] J Vogelstein, B Watson, A Packer, B Jedynak, R Yuste, and L Paninski. Model-based optimal inference of spike times and calcium dynamics given noisy and intermittent calcium-fluorescence imaging. *Biophysical Journal*, 97:636–655, 2009.

[41] Bruno M. Jedynak Weidong Han, Peter I. Frazier. Twenty questions for localizing multiple objects by counting: Bayes optimal policies for entropy loss. *submitted to IEEE Transactions on Information Theory*, 2014.

[42] A. L. Yuille and J. M.Coughlan. Fundamental limits of bayesian inference: Order parameters and phase transitions for road tracking. *IEEE Transactions on PAMI*, 22(2):160–173, February 2000.

[43] Huicheng Zheng, Mohamed Daoudi, and Bruno Jedynak. From maximum entropy to belief propagation: An application to skin detection. In *BMVC*, pages 1–10, 2004.

[44] Huicheng Zheng, Mohamed Daoudi, Bruno Jedynak, MIIRE LIFL-INT, and ENIC-Telecom Lille. Adult image detection using statistical model and neural network. *Electronic Letters on Computer Vision and Image Analysis*, 4(2):1–14, 2003.

[45] S.C. Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (frame): towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.

Chapter 3

Twenty Questions With Noise: Bayes Optimal Policies for Entropy Loss

J. Appl. Prob. **49,** 114–136 (2012)
Printed in England

TWENTY QUESTIONS WITH NOISE: BAYES OPTIMAL POLICIES FOR ENTROPY LOSS

BRUNO JEDYNAK,* *Johns Hopkins University*
PETER I. FRAZIER,** *Cornell University*
RAPHAEL SZNITMAN,*** *Johns Hopkins University*

Abstract

We consider the problem of twenty questions with noisy answers, in which we seek to find a target by repeatedly choosing a set, asking an oracle whether the target lies in this set, and obtaining an answer corrupted by noise. Starting with a prior distribution on the target's location, we seek to minimize the expected entropy of the posterior distribution. We formulate this problem as a dynamic program and show that any policy optimizing the one-step expected reduction in entropy is also optimal over the full horizon. Two such Bayes optimal policies are presented: one generalizes the probabilistic bisection policy due to Horstein and the other asks a deterministic set of questions. We study the structural properties of the latter, and illustrate its use in a computer vision application.

Keywords: Twenty questions; dynamic programing; bisection; search; object detection; entropy loss; sequential experimental design; Bayesian experimental design

2010 Mathematics Subject Classification: Primary 60J20
Secondary 62C10; 90B40; 90C39

1. Introduction

In this paper we consider the problem of finding a target $X^* \in \mathbb{R}^d$ by asking a knowledgeable oracle questions. Each question consists in choosing a set $A \subseteq \mathbb{R}^d$, querying the oracle whether X^* lies in this set, and observing the associated response. While this is closely related to the popular game of 'twenty questions', we consider here the case where answers from the oracle are corrupted with noise from a known model. This game appears naturally in a number of problems in stochastic search, stochastic optimization, and stochastic root finding. In this paper we present an illustrative application in computer vision.

We consider a Bayesian formulation of this problem using entropy loss. In dimension $d = 1$ we seek to minimize the expected entropy of the posterior after a fixed number of questions. After formulating the problem in Section 2, we show in Section 3 that any policy myopically maximizing the expected one-step reduction in entropy is also optimal in a fully sequential sense (Theorems 1 and 2), and to follow such a policy it is sufficient to query sets A whose posterior probability of containing X^* is a specific value given in Theorem 2. We then

Received 1 March 2011; revision received 16 June 2011.
* Postal address: Department of Applied Mathematics and Statistics, Johns Hopkins University, Whitehead 208-B, 3400 North Charles Street, Baltimore, MD 21218, USA. Email address: bruno.jedynak@jhu.edu
** Postal address: School of Operations Research and Industrial Engineering, Cornell University, 232 Rhodes Hall, Ithaca, NY 14850, USA.
Research supported in part by AFOSR YIP FA9550-11-1-0083.
*** Postal address: Johns Hopkins University, Hackerman Hall, 3400 North Charles Street, Baltimore, MD 21218, USA.
Research supported in part by NIH grant R01 EB 007969-01.

provide two specific Bayes optimal policies. The first, described in Section 4.1, poses questions about intervals, $A = (-\infty, x)$. The second, which we call the dyadic policy and describe in Section 4.2, poses questions about more general sets. We also provide further analysis of this second policy: a law of large numbers and a central limit theorem for the posterior entropy (Theorem 3), and an explicit characterization of the expected number of size-limited noise-free questions required to find the target after noisy questioning ceases (Theorem 4). In Section 5 we consider a modified version of the entropic loss in $d = 2$ dimensions, and show that a simple modification of the dyadic policy is asymptotically Bayes optimal for this loss function (Theorem 5). In Section 5 we also provide a central limit theorem for the posterior entropy under this policy (Theorem 6). In Section 6 we provide an illustrative application in computer vision. Concluding remarks are given in Section 7.

When the noise corrupting the oracle's responses is of a special form, that of a symmetric channel, the Bayes optimal policy for $d = 1$ with questions A restricted to be intervals (described in Section 4.1) takes a particularly natural form: choose $A = (-\infty, x)$, where x is the median of the posterior distribution. This policy, called the probabilistic bisection strategy, was first proposed in [12] (republished in [13]). This policy was recently shown to be optimal in the binary symmetric case by one of the authors in [31]. Burnašhev and Zigangirov [4] introduced a similar procedure that measures on either side of the median of the posterior over a discrete set of points, and showed that its error probability decays at an asymptotically optimal rate. For a review of these two procedures, see [5]. Both Karp and Kleinberg [14] and Ben-Or and Hassidim [1] also considered a noisy binary search problem with constant error probability over a discrete set of points, and gave optimality results for policies similar to measuring at the median of the posterior. In [14], this is part of a larger analysis in which the error probability may vary. Nowak [19], [20] analyzed noise-tolerant versions of generalized binary search for searching in a space of hypotheses. A parallel line of research has considered the case when the oracle is adversarial rather than stochastic, and is surveyed in [21].

When the questions are restricted to be intervals, the problem that we consider is similar to the stochastic root-finding problem considered in the seminal paper [24] and generalized to multiple dimensions in [3]. In the stochastic root-finding problem, one chooses a sequence of points $x_1, x_2, \ldots$ to query, and observes the corresponding values $f(x_1), f(x_2), \ldots$ of some decreasing function f at x, obscured by noise. The goal in this problem is to find the root of f. Procedures include the stochastic approximation methods of [3] and [24], as well as the Polyak–Ruppert averaging introduced independently in [22] and [25]. Asymptotic rates of convergence of these procedures are well understood; see [15, Chapter 10]. Our problem and the stochastic root-finding problem are similar because if X^* is the root of f then querying whether X^* is in $(-\infty, x)$ can be recast as querying whether $f(x) < 0$. The problems differ because the noise in observing whether $f(x) < 0$ depends upon x and is generally larger when $f(x)$ is closer to 0, while in our formulation we assume that the distribution of the oracle's response depends only on whether X^* is in the queried subset or not.

Both our problem and stochastic root-finding lie within the larger class of problems in sequential experimental design, in which we choose at each point in time which experiment to perform in order to optimize some overall value of the information obtained. The study of this area began with Robbins [23], who introduced the multi-armed bandit problem, later studied in [2], [11], [16], [32], and [33], among others. For a self-contained discussion of sequential experimental design in a Bayesian context, see [7].

2. Formulation of the problem

Nature chooses a continuous random variable X^* with density p_0 with respect to the Lebesgue measure over $\mathbb{R}^d$. The fact that X^* is continuous will turn out to be important and the arguments presented below do not generalize easily to the case where X^* is a discrete random variable.

To discover X^*, we can sequentially ask N questions. Asking the nth question, $0 \leq n \leq N-1$, involves choosing a Lebesgue measurable set $A_n \subset \mathbb{R}^d$ and evaluating: Does X^* belong to A_n?'. To avoid technical issues below, we require that A_n is the union of at most J_n half-open intervals, where $J_0, J_1, \ldots$ is a fixed sequence of natural numbers. The answer, denoted Z_n, is the indicator function of the event $\{X^* \in A_n\}$. However, Z_n is not openly communicated to us. Instead, Z_n is the input of a memoryless noisy transmission channel from which we observe the output Y_{n+1}. Here Y_{n+1} is a random variable which can be discrete or continuous, univariate or multivariate. The memoryless property of the channel expresses the fact that Y_{n+1} depends on Z_n, but not on previous questions or answers. As a consequence, repeatedly answering the same question may not provide the same answer each time. Moreover, we assume that the distribution of Y_{n+1} given Z_n does not depend on n. There is a measure μ on the space in which Y_{n+1} takes value, and the density with respect to μ of Y_{n+1} given Z_n is

$$\frac{\mathrm{P}(Y_{n+1} \in \mathrm{d}y \mid Z_n = z)}{\mathrm{d}\mu} = \begin{cases} f_1(y) & \text{if } z = 1, \\ f_0(y) & \text{if } z = 0. \end{cases} \tag{1}$$

If Y_{n+1} is discrete then we take μ to be a discrete measure, while if Y_{n+1} is continuous we take μ to be the Lebesgue measure. We require that the Shannon entropy of the conditional distribution $\mathrm{P}(Y_{n+1} \in \cdot \mid Z_n = z)$ be finite for both $z = 0$ and $z = 1$. At any time step n, we characterize what we know about X^* by computing the conditional density p_n of X^* given the history of previous measurements $D_n = (A_m, Y_{m+1})_{m=0}^{n-1}$. Following the terminology of Bayesian statistics, we call p_n the posterior density. The study of the stochastic sequences of densities p_n, under different policies, constitutes the main mathematical contribution of this paper. For an event A, we will use the notation

$$p_n(A) = \int_A p_n(x)\,\mathrm{d}x.$$

The posterior density p_{n+1} of X^* after observing D_{n+1} is elegantly described as a function of p_n, f_0, f_1, the nth question A_n, and the answer to this question Y_{n+1}.

Lemma 1. *On the event $A_n = A$ and $Y_{n+1} = y$, the posterior density on X^* is*

$$p_{n+1}(u) = \frac{1}{\mathcal{Z}}(f_1(y)\,\mathbf{1}_{\{u \in A\}} + f_0(y)\,\mathbf{1}_{\{u \notin A\}})p_n(u),$$

where

$$\mathcal{Z} = f_1(y)p_n(A) + f_0(y)(1 - p_n(A)). \tag{2}$$

Proof. On the event $A_n = A$ and $Y_{n+1} = y$, the posterior density $p_{n+1}(u) = \mathrm{P}(X^* \in \mathrm{d}u \mid D_{n+1})/\mathrm{d}\lambda = \mathrm{P}(X^* \in \mathrm{d}u \mid D_n, A_n = A,\ Y_{n+1} = y)/\mathrm{d}\lambda$, where λ is the Lebesgue measure, can be written using Bayes' formula as

$$\frac{1}{\mathcal{Z}} \frac{\mathrm{P}(Y_{n+1} \in \mathrm{d}y \mid D_n, A_n = A,\ X^* = u)}{\mathrm{d}\mu} \frac{\mathrm{P}(X^* \in \mathrm{d}u \mid D_n, A_n = A)}{\mathrm{d}\lambda}$$
$$= \frac{1}{\mathcal{Z}}(f_1(y)\,\mathbf{1}_{\{u \in A\}} + f_0(y)\,\mathbf{1}_{\{u \notin A\}})p_n(u),$$

where $\mathcal{Z}$ is the normalizing constant,

$$\mathcal{Z} = \int_u (f_1(y)\,\mathbf{1}_{\{u\in A\}} + f_0(y)\,\mathbf{1}_{\{u\notin A\}})\,p_n(u)\,\mathrm{d}u.$$

Later, we will take conditional expectations given the density p_n. Formally, these conditional expectations are taken with respect to the sigma-algebra generated by the stochastic process $\{p_n(u)\colon u \in I\}$. Because $p_n(u)$ for each u is a function of D_n by the recursive expression in Lemma 1, this sigma-algebra is a subset of the sigma-algebra generated by D_n.

We will measure the quality of the information gained about X^* from these N questions using the Shannon differential entropy. The Shannon differential entropy (see [6, Chapter 9]), or simply 'the entropy' of p_n, $H(p_n)$, is defined as

$$H(p_n) = -\int_{-\infty}^{+\infty} p_n(x)\log p_n(x)\,\mathrm{d}x,$$

where log is the logarithm to base 2. In particular, we consider the problem of finding a sequence of N questions such that the expected entropy of X^* after observing the Nth answer is minimized.

We will write this problem more formally as the infimum over policies of the expectation of the posterior entropy, but before doing so we must formally define a policy. Informally, a policy is a method for choosing the questions A_n as a function of the observations available at time n. The technical assumption that each question A_n is a union of only finitely many intervals ensures the Borel measurability of $H(p_N)$ under each policy.

First, A_n is the union of at most J_n half-open intervals, and so may be written as

$$A_n = \bigcup_{j=1}^{J_n} [a_{n,j}, b_{n,j}),$$

where $a_{n,j} \le b_{n,j}$ are elements of $\overline{\mathbb{R}} = \mathbb{R}\cup\{-\infty, +\infty\}$. If $a_{n,j} = -\infty$ then the corresponding interval is understood to be open on the left. Here $J_0, J_1, \ldots, J_{N-1}$ is any fixed sequence of natural numbers that is the same for all policies. If A_n comprises strictly less than J_n intervals then we may take $a_{n,j} = b_{n,j}$ for some j. When A_n is written in this way, the space in which A_n takes values may be identified with the space $\mathbb{A}_n = \{(a_j, b_j)\colon j = 1, \ldots, J_n,\ a_j \le b_j\}$, which is a closed subset of $\overline{\mathbb{R}}^{2J_n}$.

Then, with fixed p_0, p_n may be identified with the sequence $((a_{m,j}, b_{m,j})_{j=1}^{J_m}, Y_{m+1})_{m=0}^{n-1}$, which takes values in the space $\mathbb{S}_n = (\mathbb{A}_0 \times \cdots \times \mathbb{A}_{n-1}) \times \mathbb{R}^n$. Furthermore, the function $p_n \mapsto H(p_n)$ may be written as a measurable function from $\mathbb{S}_n$ to $\mathbb{R}$.

After having identified possible values for A_n with points in $\mathbb{A}_n$ and possible values for p_n with points in $\mathbb{S}_n$, we define a policy π to be a sequence of functions $\pi = (\pi_0, \pi_1, \ldots)$, where $\pi_n\colon \mathbb{S}_n \mapsto \mathbb{A}_n$ is a measurable function. We let Π be the space of all such policies. Any such policy π induces a probability measure on $((a_{n,j}, b_{n,j})_{j=1}^{J_n}, Y_{n+1})_{n=0}^{N-1}$. We let E^π indicate the expectation with respect to this probability measure. In a slight abuse of notation, we will sometimes talk of $p \in \mathbb{S}_n$ and $A \in \mathbb{A}_n$, by which we mean the density p associated with a vector in $\mathbb{S}_n$, or the set A associated with a vector in $\mathbb{A}_n$.

With this definition of a policy π, the associated measure E^π, and the space of all policies Π, the problem under consideration may be written as

$$\inf_{\pi\in\Pi} \mathrm{E}^\pi[H(p_N)]. \tag{3}$$

Any policy attaining the infimum is called optimal. We consider this problem for the general case in Section 3, and for the specific cases of $d = 1$ and $d = 2$ in Sections 4 and 5, respectively. In Section 5 we also consider a modification of this objective function that separately considers the entropy of the marginal posterior distribution, and ensures that both entropies are small. This prevents a policy from obtaining optimality by learning one coordinate of X^* without learning the other.

3. Entropy loss and channel capacity

In this section we consider the problem (3) of minimizing the expected entropy of the posterior over $\mathbb{R}^d$. We present general results characterizing optimal policies, which will be used to create specific policies in Sections 4 and 5.

We first present some notation that will be used within our results. Let φ be the function with domain [0, 1] defined by

$$\varphi(u) = H(uf_1 + (1-u)f_0) - uH(f_1) - (1-u)H(f_0);$$

$\varphi(u)$ is a mutual information for each u (see (7) and (10) below). The associated channel capacity C is

$$C = \sup_{u\in[0,1]} \varphi(u).$$

Below, in Theorem 1, we show that this maximum is attained in (0, 1). Let $u^* \in (0, 1)$ be a point attaining this maximum, so $\varphi(u^*) = C$.

We show that an optimal policy consists of choosing each A_n so that $p_n(A_n) = u^*$. When the A_n are chosen in this way, the expected entropy decreases arithmetically by the constant C at each step. Moreover, if the communication channel is symmetric in the sense that $\varphi(1-u) = \varphi(u)$ for all $0 \le u \le 1$, then $u^* = \frac{1}{2}$. In the noiseless case, or even the case where the supports of f_0 and f_1 do not overlap, the model is symmetric, $C = 1$, and the obvious bisection policy is optimal.

Optimal policies constructed by choosing $p_n(A_n) = u^*$ are greedy policies (or 'knowledge-gradient' policies, as defined in [9]), since they make decisions that would be optimal if only one measurement remained, i.e. if N were equal to $n + 1$. Such greedy policies are usually used only as heuristics, and so it is interesting that they are optimal in this problem.

Our analysis relies on dynamic programming. To support this analysis, we define the value function

$$V(p, n) = \inf_{\pi\in\Pi} \mathrm{E}^\pi[H(p_N) \mid p_n = p], \qquad p \in \mathbb{S}_n,\ n = 0, \ldots, N.$$

Standard results from controlled Markov processes show that this value function satisfies Bellman's recursion (see Section 3.7 of [8]),

$$V(p, n) = \inf_{A\in\mathbb{A}_n} \mathrm{E}[V(p_{n+1}, n+1) \mid A_n = A,\ p_n = p], \qquad p \in \mathbb{S}_n,\ n < N, \tag{4}$$

where the expectation is taken over Y_{n+1}, and any policy attaining the minimum of (4) is optimal (see Section 2.3 of [8]). In general, the results of [8] for general Borel models imply only that $V(\cdot, n)\colon \mathbb{S}_n \mapsto \mathbb{R}$ is universally measurable, and do not imply Borel measurability. However, we show below in Theorem 2 that, in our case, $V(\cdot, n)\colon \mathbb{S}_n \mapsto \mathbb{R}$ is a Borel-measurable function.

As a preliminary step toward solving Bellman's recursion, we present the following theorem, which shows that minimizing the expected entropy of the posterior one step into the future can be accomplished by choosing A_n as described above. Furthermore, it shows that the expected reduction in entropy is the channel capacity C.

Theorem 1. *We have*

$$\inf_{A\in\mathbb{A}_n} \mathrm{E}[H(p_{n+1}) \mid A_n = A,\, p_n] = H(p_n) - C, \tag{5}$$

where the expectation is taken over Y_{n+1}. Moreover, there exists a point $u^* \in (0, 1)$ such that $\varphi(u^*) = C$, and the minimum in (5) is attained by choosing A such that $p_n(A) = u^*$.

Proof. We first rewrite the expected entropy as

$$\mathrm{E}[H(p_{n+1}) \mid A_n = A,\, p_n] = H(p_n) - I(X^*, Y_{n+1} \mid A_n = A,\, p_n),$$

where $I(X^*, Y_{n+1} \mid A_n = A,\, p_n)$ is the mutual information between the conditional distributions of X^* and Y_{n+1} (see [6, Chapter 2]), and we recall that the entropy of X^* given $A_n = A$ and p_n is exactly $H(p_n)$. This leads to

$$\inf_{A\in\mathbb{A}_n} \mathrm{E}[H(p_{n+1}) \mid A_n = A,\, p_n] = H(p_n) - \sup_{A\in\mathbb{A}_n} I(X^*, Y_{n+1} \mid A_n = A,\, p_n). \tag{6}$$

Temporarily fixing A, we expand the mutual information as

$$I(X^*, Y_{n+1} \mid A_n = A,\, p_n) = H(Y_{n+1} \mid A_n = A,\, p_n) - H(Y_{n+1} \mid X^*, A_n = A,\, p_n), \tag{7}$$

where $H(\cdot \mid \cdot)$ is the conditional entropy, as defined in [6, Chapter 2]. Using (2),

$$H(Y_{n+1} \mid A_n = A,\, p_n) = H(p_n(A) f_1 + (1 - p_n(A)) f_0). \tag{8}$$

Also,

$$\begin{aligned} H(Y_{n+1} \mid X^*, A_n = A,\, p_n) &= \int_u p_n(u) H(Y_{n+1} \mid X^* = u,\ A_n = A,\, p_n)\,\mathrm{d}u \\ &= \int_{u\in A} p_n(u) H(f_1)\,\mathrm{d}u + \int_{u\notin A} p_n(u) H(f_0)\,\mathrm{d}u \\ &= H(f_1) p_n(A) + H(f_0)(1 - p_n(A)). \end{aligned} \tag{9}$$

The difference between (8) and (9) is $\varphi(p_n(A))$, and so

$$I(X^*, Y_{n+1} \mid A_n = A,\, p_n) = \varphi(p_n(A)). \tag{10}$$

This and (6) together show that

$$\sup_{A\in\mathbb{A}_n} I(X^*, Y_{n+1} \mid A_n = A,\, p_n) = \sup_{A\in\mathbb{A}_n} \varphi(p_n(A)) = \sup_{u\in[0,1]} \varphi(u) = C.$$

This shows (5), and that the infimum in (5) is attained by any set A with $\varphi(p_n(A)) = C$. It remains only to show the existence of a point $u^* \in (0, 1)$, with $\varphi(u^*) = C$.

First, φ is a continuous function, so its maximum over the compact interval $[0, 1]$ is attained. If the maximum is attained in $(0, 1)$ then we simply choose u^* to be this point. Now consider the case when the maximum is attained at $u \in \{0, 1\}$. Because φ is a mutual information for each u, it is nonnegative. Also, $\varphi(0) = \varphi(1) = 0$. Thus, if the maximum is attained at $u \in \{0, 1\}$ then $\varphi(u) = 0$ for all u, and we can choose u^* in the open interval $(0, 1)$.

We are now ready to present the main result of this section, which gives a simple characterization of optimal policies.

Theorem 2. *Any policy that chooses each A_n to satisfy $p_n(A_n) = u^* \in \arg\max_{u\in[0,1]} \varphi(u)$ is optimal. In addition, for each n, the value function $V(\cdot, n)\colon \mathbb{S}_n \mapsto \mathbb{R}$ is Borel measurable and is given by*

$$V(p_n, n) = H(p_n) - (N - n)C. \tag{11}$$

Proof. It is enough to show that, for each $n = 0, 1, \ldots, N$, the value function is given by (11), and that the described policy achieves the minimum in Bellman's recursion (4). Measurability of $V(\cdot, n)\colon \mathbb{S}_n \mapsto \mathbb{R}$ then follows from the fact that $p_n \mapsto H(p_n)$ is Borel measurable when written as a function from $\mathbb{S}_n$ to $\mathbb{R}$. We proceed by backward induction on n. The value function clearly has the claimed form at the final time $n = N$. Now, fix any $n < N$ and assume that the value function is of the form claimed for $n + 1$. Then, Bellman's recursion and the induction hypothesis show that

$$\begin{aligned} V(p_n, n) &= \inf_{A\in\mathbb{A}_n} \mathrm{E}[V(p_{n+1}, n+1) \mid A_n = A, p_n] \\ &= \inf_{A\in\mathbb{A}_n} \mathrm{E}[H(p_{n+1}) - (N - n - 1)C \mid A_n = A, p_n] \\ &= \inf_{A\in\mathbb{A}_n} \mathrm{E}[H(p_{n+1}) \mid A_n = A, p_n] - (N - n - 1)C \\ &= H(p_n) - C - (N - n - 1)C \\ &= H(p_n) - (N - n)C, \end{aligned} \tag{12}$$

where we have used (5) in Theorem 1 to obtain the fourth equality. Theorem 1 also shows that the infimum in (12) is attained when A satisfies $p_n(A) = u^*$, and so the described policy achieves the minimum in Bellman's recursion.

We offer the following interpretation of the optimal reduction in entropy shown in Theorem 2. First, the entropy of a random variable uniformly distributed over $[a, b]$ is $\log(b - a)$. The quantity $2^{H(X)}$ for a continuous random variable X can then be interpreted as the length of the support of a uniform random variable with the same entropy as X. We refer to this quantity more simply as the 'length of X'. If the prior distribution of X^* is uniform over [0, 1] then the length of X^* under p_0 is 1 and Theorem 2 shows that the expected length of X^* under p_N is no less than 2^{-CN}, where this bound on the expected length can be achieved using an optimal policy.

We conclude this section by discussing u^* and C in a few specific cases. In general, there are no simple expressions for u^* and C. However, in certain symmetric cases the following proposition shows that $u^* = \frac{1}{2}$.

Proposition 1. *If the channel has the symmetry*

$$\varphi(u) = \varphi(1 - u) \quad \textit{for all } 0 \le u \le 1 \tag{13}$$

then $\frac{1}{2} \in \arg\max_{u\in[0,1]} \varphi(u)$ and we may take $u^* = \frac{1}{2}$. Furthermore, if

$$H(uf_1 + (1 - u)f_0) = H(uf_0 + (1 - u)f_1) \quad \textit{for all } u \in [0, 1] \tag{14}$$

then this is sufficient to guarantee (13).

Proof. Let u' be a maximizer of $\varphi(u)$. It might be equal to u^*, or if there is more than one maximizer, it might differ. Note that $\frac{1}{2} = \frac{1}{2}u' + \frac{1}{2}(1 - u')$. The function φ is concave (see [6, Theorem 2.7.4, Chapter 2]), implying that $\varphi(\frac{1}{2}) \ge \frac{1}{2}\varphi(u') + \frac{1}{2}\varphi(1 - u')$. Now, using $\varphi(u') = \varphi(1 - u')$, we obtain $\varphi(\frac{1}{2}) \ge \varphi(u')$, which shows that $\frac{1}{2} \in \arg\max_{u\in[0,1]} \varphi(u)$. If (14) is met then $H(f_0) = H(f_1)$ by taking $u = 0$, and (13) follows directly from the definition of φ.

TABLE 1: Channel capacity and the value u^* at which the channel capacity is achieved. The binary symmetric case is treated in [31].

Channel	Model	Channel capacity	u^*
Binary symmetric	$\begin{array}{c\|cc} & 0 & 1 \\ \hline f_0 & 1-\varepsilon & \varepsilon \\ f_1 & \varepsilon & 1-\varepsilon \end{array}$	$1 - h(\varepsilon)$	$\frac{1}{2}$
Binary erasure	$\begin{array}{c\|ccc} & 0 & 1 & \mathrm{e} \\ \hline f_0 & 1-\varepsilon & 0 & \varepsilon \\ f_1 & 0 & 1-\varepsilon & \varepsilon \end{array}$	$1 - \varepsilon$	$\frac{1}{2}$
Z	$\begin{array}{c\|cc} & 0 & 1 \\ \hline f_0 & 1 & 0 \\ f_1 & \varepsilon & 1-\varepsilon \end{array}$	$h(u^*(1-\varepsilon)) - u^* h(\varepsilon)$	$\dfrac{1/(1-\varepsilon)}{1+\mathrm{e}^{h(\varepsilon)/(1-\varepsilon)}}$
Multivariate normal	$f_0 \sim N(m_0, \Sigma)$	Not analytical	$\frac{1}{2}$
Symmetric	$f_1 \sim N(m_1, \Sigma)$	Not analytical	$\frac{1}{2}$

A few simple channels with expressions for u^* and C are presented in Table 1. We use the notation $B(u)$ for a Bernoulli random variable with parameter u and $h(u)$ for $H(B(u))$, the entropy of this random variable. In the multivariate normal case, $u^* = \frac{1}{2}$ follows from Proposition 1 because $uf_1 + (1-u)f_0$ is the multivariate normal density with mean $um_1 + (1-u)m_0$ and variance Σ, and the entropy of a multivariate normal distribution does not depend on its mean, implying that (14) is satisfied.

4. One-dimensional optimal policies

We now present two specific policies in dimension $d = 1$ that satisfy the sufficient conditions for optimality given in Theorem 2: the probabilistic bisection policy and the dyadic policy. After defining these two policies in Sections 4.1 and 4.2, we study the sequence of entropies $(H(p_n) : n \geq 1)$ that they generate, focusing on the dyadic policy. In addition to Theorem 2, which shows that $\mathrm{E}^\pi[H(p_n)] = H(p_0) - nC$ for any optimal policy π, the analysis of the dyadic policy in Section 4.2 provides a strong law of large numbers and a central limit theorem for $H(p_n)$. In further analysis of the dyadic policy, in Section 4.3 we analyze the number of size-limited noise-free questions required to find X^* after noisy questioning with the dyadic policy ceases, which is a metric that is important in the application discussed in Section 6.

To support the analyses in Sections 4.1 and 4.2, we first give here a general expression for the one-step change in entropy, $H(p_{n+1}) - H(p_n)$, under any policy π satisfying $p_n(A_n) = u^*$.

Lemma 2. *We have*

$$\begin{aligned} &H(p_{n+1}) - H(p_n) \\ &\quad = -D\left(B\left(\frac{u^* f_1(y)}{Z}\right), B(u^*)\right) \\ &\qquad + \frac{u(1-u^*)}{Z}(f_1(y) - f_0(y))(H(p_n^+) - \log u^* - H(p_n^-) + \log(1-u^*)), \end{aligned} \tag{15}$$

where D is the Kullback–Leibler divergence.

Proof. First, we define two densities:

$$p_n^+(x) = \begin{cases} \dfrac{p_n(x)}{u^*} & \text{if } x \in A_n, \\ 0 & \text{if } x \in \bar{A}_n, \end{cases} \qquad p_n^-(x) = \begin{cases} \dfrac{p_n(x)}{1-u^*} & \text{if } x \in \bar{A}_n, \\ 0 & \text{if } x \in A_n, \end{cases}$$

where $\bar{A}_n$ is the complement of A_n. Their respective entropies are

$$H(p_n^+) = \log u^* - \frac{1}{u^*} \int_{A_n} p_n(x) \log p_n(x)\, \mathrm{d}x,$$

$$H(p_n^-) = \log(1-u^*) - \frac{1}{1-u^*} \int_{\bar{A}_n} p_n(x) \log p_n(x)\, \mathrm{d}x,$$

and $H(p_n) = u^* H(p_n^+) + (1-u^*) H(p_n^-) + h(u^*)$.

Using Lemma 1, for a given observation $Y_{n+1} = y$, we have

$$\begin{aligned} H(p_{n+1}) &= \log Z - p_{n+1}(A_n) \log f_1(y) - p_{n+1}(\bar{A}_n) \log f_0(y) \\ &\quad - \frac{1}{Z} f_1(y) \int_{A_n} p_n(x) \log p_n(x)\, \mathrm{d}x - \frac{1}{Z} f_0(y) \int_{\bar{A}_n} p_n(x) \log p_n(x)\, \mathrm{d}x \\ &= \log Z - \frac{1}{Z} u f_1(y) \log f_1(y) - \frac{1}{Z}(1-u) f_0(y) \log f_0(y) \\ &\quad - \frac{1}{Z} u^* f_1(y)(\log u^* - H(p_n^+)) - \frac{1}{Z}(1-u^*) f_0(y)(\log(1-u^*) - H(p_n^-)). \end{aligned}$$

Expanding and rearranging, we obtain (15).

Note also that, under an optimal policy, the density of Y_{n+1} is the mixture of densities $u^* f_1 + (1-u^*) f_0$ according to Lemma 1, and the random variables $Y_1, Y_2, \ldots$ are independent and identically distributed (i.i.d.).

4.1. Probabilistic bisection policy

We first consider the case when questions are limited to intervals $A = (-\infty, a)$, $a \in \mathbb{R}$. This limitation appears naturally in applications, such as stochastic root finding [24] and signal estimation [5]. In this case, an optimal policy consists of choosing a_n such that $\int_{-\infty}^{a_n} p_n(x)\, \mathrm{d}x = u^*$. Such an a_n always exists, but is not necessarily unique.

When the model is symmetric in the sense of Proposition 1, $u^* = \frac{1}{2}$, and a_n is the median of p_n. This policy of measuring at the median of the posterior is the probabilistic bisection policy introduced in [12]. Thus, the optimal policy with interval questions and general channels is a generalization of the probabilistic bisection policy, and we continue to refer to it as the probabilistic bisection policy even when $u^* \neq \frac{1}{2}$.

We briefly consider the behavior of $(H(p_n) : n \geq 1)$ under the probabilistic bisection policy. We assume a binary symmetric channel with noise parameter ε. Recall that $u^* = \frac{1}{2}$ in this case, and

$$D\left(B\left(\frac{f_1(Y_{n+1})}{2Z}\right), B\left(\frac{1}{2}\right)\right) = 1 - h(\varepsilon). \tag{16}$$

Moreover,

$$H(p_{n+1}) - H(p_n) = h(\varepsilon) - 1 + \left(\tfrac{1}{2} - \varepsilon\right) W_{n+1}(H(p_n^+) - H(p_n^-)),$$

where the W_n are i.i.d. Rademacher random variables. In this situation, even when p_0 is the density of the uniform distribution over the interval [0, 1], the behavior of the process $H(p_n)$ can be complicated. A simulation of $H(p_n)$ is presented in Figure 1. The high degree of

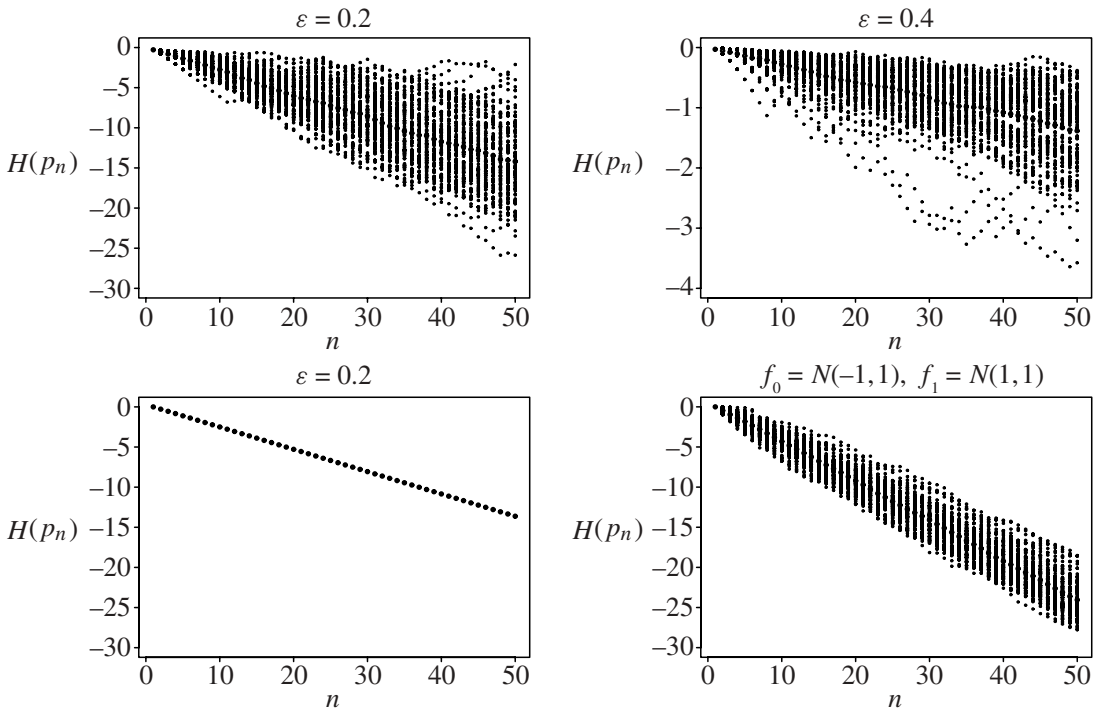


FIGURE 1: The process $H(p_n)$ for the binary symmetric channel. Here p_0 is Uniform([0, 1]). *Top:* the questions are chosen by the probabilistic bisection policy. *Top left:* $\varepsilon = 0.2$ and $C = 0.28$. *Top right:* $\varepsilon = 0.4$ and $C = 0.03$. *Bottom:* the questions are chosen according to the dyadic policy. *Bottom left:* binary symmetric channel $\varepsilon = 0.2$. *Bottom right:* normal channel, with $f_0 \sim N(-1, 1)$, $f_1 \sim N(1, 1)$, and $C = 0.47$.

variation of $H(p_n)$ around its mean value evident in Figure 1 may be disadvantageous in some applications. We do not pursue the probabilistic bisection policy further in this paper.

4.2. Dyadic policy

Consider now the situation where all sets in $\mathbb{A}_n$ are available as questions, and p_0 is piecewise constant with finite support. Let $I = \{I_k : k = 0, \ldots, K-1\}$ be a finite partition of the support of p_0 into intervals such that p_0 is constant and strictly positive in each of these intervals. We assume that each interval I_k is closed on the left and open on the right, so $I_k = [a_k, b_k)$ with $a_k \in \mathbb{R}$ and $b_k \in \mathbb{R}$. This assumption is without loss of generality, because if it is not met, we can alter the prior density p_0 on a set of Lebesgue measure 0 (which does not change the corresponding prior probability measure) to meet it. We also assume that the constants J_n used to construct $\mathbb{A}_n$ satisfy $J_n \geq 2^{n+1}K$. If this restriction is not met then we are free to increase J_n in most applications.

For each $k = 0, \ldots, K-1$, we partition I_k into two intervals, $A_{0,2k}$ and $A_{0,2k+1}$:

$$A_{0,2k} = [a_{0,2k}, b_{0,2k}) = [a_k, a_k + u^*(b_k - a_k)),$$
$$A_{0,2k+1} = [a_{0,2k+1}, b_{0,2k+1}) = [a_k + u^*(b_k - a_k), b_k).$$

With this partition, the mass $p_0(A_{0,2k}) = u^*\, p_0(I_k)$. The question asked at time 0 is

$$A_0 = \bigcup_{k=0}^{K-1} A_{0,2k},$$

and $p_0(A_0) = u^*$.

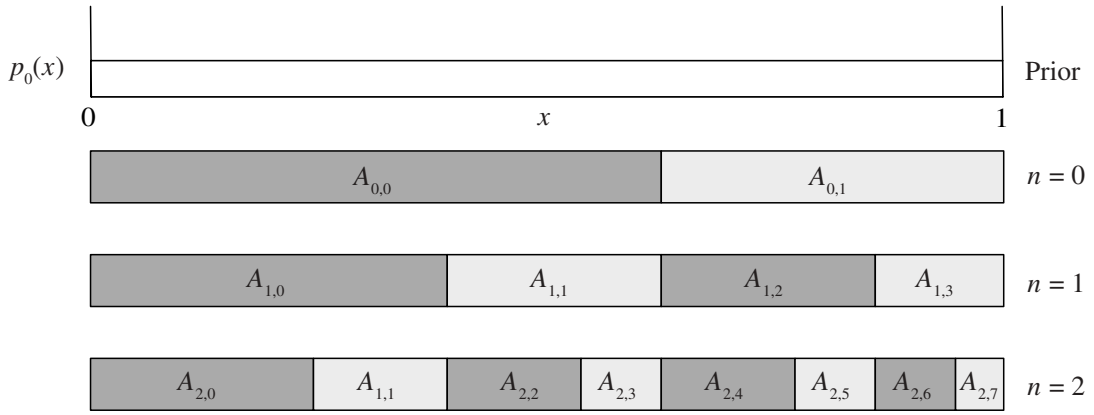


FIGURE 2: Illustration of the dyadic policy when p_0 is uniform on $[0, 1]$ and $u^* = \frac{5}{8}$. The prior is displayed above illustrations of the sets $A_{n,k}$ for $n = 0, 1, 2$. Each question A_n is the union of the dark gray subsets $A_{n,k}$ for that value of n.

We use a similar procedure recursively for each $n = 0, 1, \ldots$ to partition each $A_{n,k}$ into two intervals, $A_{n+1,2k}$ and $A_{n+1,2k+1}$, and then construct the question A_{n+1} from these partitions. Let $K_n = 2^{n+1}K$, and, for $k = 0, \ldots, K_n - 1$, define

$$A_{n+1,2k} = [a_{n+1,2k}, b_{n+1,2k}) = [a_{n,k}, a_{n,k} + u^*(b_{n,k} - a_{n,k})),$$
$$A_{n+1,2k+1} = [a_{n+1,2k+1}, b_{n+1,2k+1}) = [a_{n,k} + u^*(b_{n,k} - a_{n,k}), b_{n,k}).$$

Then, from these, the question to be asked at time $n + 1$ is

$$A_{n+1} = \bigcup_{k=0}^{K_n - 1} A_{n+1,2k}.$$

This construction is illustrated in Figure 2.

Observe that $p_{n+1}(A_{n+1,2k}) = u^*\, p_{n+1}(A_{n,k})$ implies that

$$p_{n+1}(A_{n+1}) = \sum_{k=0}^{K_n - 1} u^* p_{n+1}(A_{n,k}) = u^*$$

because $\{A_{n,k} : k = 0, \ldots, K_n - 1\}$ is a partition of the support of p_0. Thus, this construction satisfies $p_n(A_{n+1}) = u^*$, and is optimal. In addition, the sets $A_0, \ldots, A_{n-1}$ are constructed without knowledge of the responses, and, thus, this policy is nonadaptive. This is useful in applications, allowing multiple questions to be asked simultaneously. We call this policy the dyadic policy because each question is constructed by dividing the previous question's intervals into two pieces.

We now provide an analysis that leads to a law of large numbers and a central limit theorem for $H(p_n)$ under this policy when n is large. Under the dyadic policy, we have

$$H(p_n^+) = H(p_n) + \log u^* \quad \text{and} \quad H(p_n^-) = H(p_n) + \log(1 - u^*),$$

which implies, using (15), that

$$H(p_{n+1}) - H(p_n) = -D\left(B\left(\frac{u^* f_1(Y_{n+1})}{u^* f_1(Y_{n+1}) + (1 - u^*) f_0(Y_{n+1})}\right), B(u^*)\right), \tag{17}$$

where Y_n is, as already stated, a sequence of i.i.d. random variables with density $u^* f_1 + (1 - u^*) f_0$. We read from (17) that $H(p_n)$ is, in this case, a sum of i.i.d. random variables. Moreover, each one is bounded above and below. Indeed,

$$\begin{aligned} 0 &\le D\left(B\left(\frac{u^* f_1(Y_{n+1})}{u^* f_1(Y_{n+1}) + (1-u^*) f_0(Y_{n+1})} \right), B(u^*) \right) \\ &\le \max(D(B(0), B(u^*)), D(B(1), B(u^*))), \end{aligned}$$

implying the bound

$$\min(\log(u^*), \log(1-u^*)) \le H(p_{n+1}) - H(p_n) \le 0. \tag{18}$$

For the binary symmetric channel, (17) reduces to a constant, as noted in (16). This proves the following theorem.

Theorem 3. *For any piecewise constant p_0, using the dyadic policy,*

$$\lim_{n\to\infty} \frac{H(p_n)}{n} = -C \quad \textit{almost surely (a.s.)} \tag{19}$$

and

$$\lim_{n\to\infty} \frac{H(p_n) + nC}{\sqrt{n}} \stackrel{\mathrm{D}}{=} N(0, \sigma^2), \tag{20}$$

where σ^2 is the variance of the increment $H(p_{n+1}) - H(p_n)$, which can be computed from the distribution given in (17). A degenerate situation occurs for the binary symmetric channel with noise ε. In this case, the sequence $H(p_n) = H(p_0) - nC$ is constant.

The dyadic policy is illustrated in the two bottom diagrams of Figure 1, where $H(p_n)$ is plotted as a function of n. The binary symmetric channel model with $\varepsilon = 0.2$ is shown in the bottom-left diagram. The sequence $H(p_n)$ is constant, in sharp contrast with the behavior of $H(p_n)$ for the same model under the probabilistic bisection policy, shown in the top-left diagram. Finally, a normal channel is presented in the bottom-right diagram.

4.3. Expected number of noise-free questions

In this section we consider an alternative to entropy for measuring performance, which arises in the example considered in Section 6. We suppose that, in addition to the noisy questions previously discussed, we also have the ability to ask a noise-free oracle whether X^* lies in a given set, where the sets about which we can ask noise-free questions come from some restricted class, e.g. their size is below a threshold. In Section 6, the sets about which we can ask noise-free questions correspond to pixels in an image. We suppose that after a fixed number N of noisy questions, we query sets using the noise-free questions until we find X^*. The loss function that arises naturally in this situation is the expected number of noise-free questions until X^* is found.

Given a posterior p_N resulting from the first stage of noisy questions, the optimal way in which to ask the noise-free questions is to first sort the available sets about which noise-free questions can be asked, in decreasing order of their probability of containing X^* under p_N. Then, query these sets in this order until X^* is found. Observing that X^* is not in a particular set alters the probability of the other sets, but does not change the order of these probabilities. Thus, it is sufficient to ask the noise-free questions in an order that depends only upon p_N, and no subsequent information.

We give below in Theorem 4 an explicit expression for the expected number of noise-free questions required after the dyadic policy completes. Before giving this expression in Theorem 4, we have the following preliminary result. In both this result and Theorem 4, we assume the dyadic policy, a uniform p_0, and a binary symmetric channel with noise parameter ε.

Proposition 2. *For each $k \in \{0, \ldots, 2^N - 1\}$, let M_k be the number of noisy questions A_n whose answer has indicated $X^* \in A_{N-1,k}$, either via $A_{N-1,k} \subseteq A_n$ and $Y_n = 1$, or $A_{N-1,k} \subsetneq I \setminus A_n$ and $Y_n = 0$. Then, the density of $A_{N-1,k}$ under p_N is*

$$|I|^{-1} 2^N (1-\varepsilon)^{M_k} \varepsilon^{N-M_k}.$$

Furthermore, for each $m \in \{0, \ldots, N\}$, the number of k with $M_k = m$ is deterministic, and is equal to $\binom{N}{m}$.

Proof. In the proof, we refer to $A_{N-1,k}$ as B_k. During noisy questioning, each time an answer indicates $X^* \in B_k$, we multiply the posterior density on B_k by $2(1-\varepsilon)$, and each time an answer indicates $X^* \notin B_k$, we multiply by 2ε. Since the prior density was $|I|^{-1}$, the posterior density on B_k after all N measurements is $|I|^{-1} 2^N (1-\varepsilon)^{M_k} \varepsilon^{N-M_k}$.

For each $k \in \{0, \ldots, 2^N - 1\}$, let $b_{kn} = \mathbf{1}_{\{B_k \subseteq A_n\}}$, and define the binary sequence $b_k = (b_{k1}, \ldots, b_{kN})$. By construction of the sets B_k under the dyadic policy, each b_k is unique. Since there are 2^N possible binary sequences of N bits, and 2^N sets B_k, the mapping between B_k and b_k is a bijection.

Consider a sequence of answers to noisy questions, $Y_1, \ldots, Y_N$. For each b_k, define a subset $D_k = \{n \in \{1, \ldots, N\} \colon b_{kn} = Y_n\}$. Each b_k defines a unique subset D_k. Since there are 2^N subsets and 2^N sequences b_k, each subset $D \subseteq \{1, \ldots, N\}$ is equal to some D_k. Thus, the mapping between b_k and D_k is a bijection.

Because $M_k = |D_k|$, the number of k with $M_k = m$ is equal to the number of subsets $D \subseteq \{1, \ldots, N\}$ of size m. This number is exactly $\binom{N}{m}$.

Proposition 2 shows that the number of sets $A_{N-1,k}$ with any given posterior density $|I|^{-1} 2^N (1-\varepsilon)^m \varepsilon^{N-m}$ is deterministic. Figure 3(a) shows this posterior probability distribution, after sorting the sets according to their density, for $N = 5$ and $\varepsilon = 0.3$. The expectation under p_N of the number of noise-free questions required to find X^* depends only upon this sorted posterior probability density, and is thus also deterministic. We now give an expression for this expectation in Theorem 4.

Theorem 4. *In each interval $A_{N-1,k}$ for $k = 0, \ldots, 2^N - 1$, assume that there are ℓ disjoint, equally sized sets about which we can ask noise-free questions. Then the expectation under p_N of the number of noise-free questions required to find X^* is*

$$\sum_{m=0}^{N} \binom{N}{m} (1-\varepsilon)^m \varepsilon^{N-m} \left[\frac{\binom{N}{m} + 1/\ell}{2} + \sum_{m'=m+1}^{N} \binom{N}{m'} \right] \ell.$$

Proof. First, if we have a collection of disjoint subsets $C_1, \ldots, C_K$, each with probability $1/K$ of containing X^*, and we query each subset in order of increasing index until we find X^*, then we ask k questions when $X^* \in C_k$ and the expected number of questions asked is $\sum_{k=1}^{K} k \, \mathrm{P}(X^* \in C_k) = \sum_{k=1}^{K} k/K = (K+1)/2$. Under p_N, Proposition 2 shows that X^* has probability

$$\binom{N}{m} (1-\varepsilon)^m \varepsilon^{N-m} \tag{21}$$

of being in a subset $A_{N-1,k}$ with $M_k = m$, because there are $\binom{N}{m}$ such intervals, each of

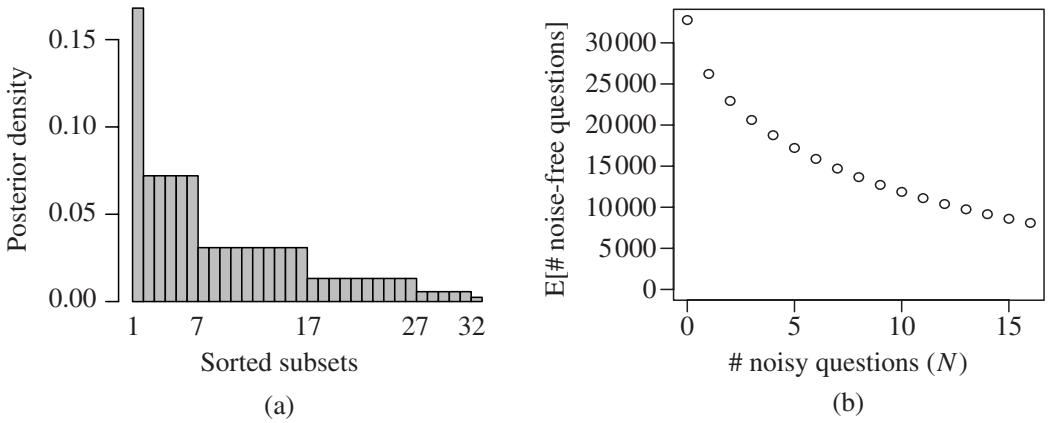


FIGURE 3: (a) The posterior density p_N for the binary symmetric channel with the dyadic policy, with subsets $A_{N-1,k}$ sorted in order of decreasing posterior density $p_N(x)$, and $N = 5$. (b) The expected number of noise-free questions as a function of N, for a fixed collection of 2^{16} subsets about which noise-free questions may be asked. In both (a) and (b), $\varepsilon = 0.3$.

size $2^{-N}|I|$, and each of density $|I|^{-1}2^N(1-\varepsilon)^m\varepsilon^{N-m}$. Then, because the number of noise-free questions available in each $A_{N-1,k}$ is ℓ, the expected number of noise-free questions, conditioned on X^* being in a subset $A_{N-1,k}$ with $M_k = m$, is

$$\frac{\binom{N}{m}\ell + 1}{2} + \sum_{m'=m+1}^{N} \binom{N}{m'}\ell. \tag{22}$$

Here, the first term is the number of questions asked in subsets with $M_k = m$, and the second term is the number of questions asked in subsets with $M_k > m$, which had a strictly higher density $p_N(x)$ and were queried earlier. The result follows by combining (21) and (22) and summing over k.

Using Theorem 4, we consider the effect of varying N. Suppose that the sets about which noise-free questions may be asked are pixels in an image, as in the example in Section 6. Take $I = [0, 1]$, and suppose that each pixel is of size 2^{-L} and occupies a region $[k2^{-L}, (k+1)2^{-L}]$ for some $k = 0, \ldots, 2^L$. If sets $A_{N-1,k}$ must contain an integer numbers of pixels then we may naturally consider any N between 0 and L. For any such N, the number of pixels ℓ in a subset $A_{N-1,k}$ is $\ell = 2^{L-N}$. In this setting, the expected number of noise-free questions asked as a function of N is shown in Figure 3(b) for $L = 16$ and $\varepsilon = 0.3$. It can be seen from the figure that there is a dramatic decrease in the expected number of noise-free questions as the number of noisy questions N increases.

5. Optimal policies in two dimensions with entropy loss

We now consider the case $d = 2$, in which X^* is a two-dimensional random variable, $X^* = (X_1^*, X_2^*)$, with joint density p_0. To minimize the expected entropy $\mathrm{E}[H(p_N)]$ of the two-dimensional posterior distribution on X^* at time N, Theorem 2 from Section 3 shows that it is optimal to use any policy satisfying $p_n(A_n) = u^*$.

While the objective function $\mathrm{E}[H(p_N)]$ is natural in dimension $d = 1$, it has a drawback in $d = 2$ and higher dimensions. This is well illustrated using an example. Assume that X_1^* and

X_2^* are independent and uniformly distributed over intervals of lengths s_1 and s_2, respectively. Then $H(p) = \log(s_1 s_2)$. In this case, $H(p)$ can be arbitrarily small even if the entropy of one of the marginal densities remains large, e.g. $s_2 = 1$.

This leads us to consider objective functions without this drawback. For example, we might wish to solve $\inf_\pi \mathrm{E}^\pi[\max(H_1(p_N), H_2(p_N))]$, where $H_1(p_N) = H(\int p_N(\cdot, u_2)\,\mathrm{d}u_2)$ and $H_2(p_N) = H(\int p_N(u_1, \cdot)\,\mathrm{d}u_1)$ are the entropies of the marginals. However, solving this problem directly seems out of reach. Instead, we focus on reducing $\mathrm{E}^\pi[\max(H_1(p_N), H_2(p_N))]$ at an asymptotically optimal rate by solving

$$V(p) = \inf_\pi \liminf_{N\to\infty} \frac{1}{N} \mathrm{E}^\pi[\max(H_1(p_N), H_2(p_N)) \mid p_0 = p]. \tag{23}$$

We use the lim inf to include policies for which the limit might not exist. Throughout this section, we assume that both $H_1(p_0)$ and $H_2(p_0)$ are finite.

For further simplification, we assume that questions concern only one coordinate. That is, the sets queried are either of type 1, $A_n = B \times \mathbb{R}$, where B is a finite union of intervals of $\mathbb{R}$, or, alternatively, of type 2, $A_n = \mathbb{R} \times B$. In each case, we assume that the response passes through a memoryless noisy channel with densities $f_0^{(1)}$ and $f_1^{(1)}$ for questions of type 1, and $f_0^{(2)}$ and $f_1^{(2)}$ for questions of type 2. Let C_1 and C_2 be the channel capacities for questions of type 1 and 2, respectively. We also assume that p_0 is a product of its marginals. This guarantees that p_n for all $n > 0$ remains a product of its marginals and that only one marginal distribution is modified at each point in time. This is shown by the following lemma.

Lemma 3. *Assume that $p_n(u_1, u_2) = p_n^{(1)}(u_1) p_n^{(2)}(u_2)$ and that we choose a question of type 1 with $A_n = B \times \mathbb{R}$. Then, given $Y_{n+1} = y$,*

$$p_{n+1}(u_1, u_2) = \frac{1}{\mathcal{Z}_1}(f_1^{(1)}(y)\,\mathbf{1}_{\{u_1 \in B\}} + f_0^{(1)}(y)\,\mathbf{1}_{\{u_1 \notin B\}}) p_n^{(1)}(u_1) p_n^{(2)}(u_2),$$

where $\mathcal{Z}_1 = f_1^{(1)}(y) p_n^{(1)}(B) + f_0^{(1)}(y)(1 - p_n^{(1)}(B))$.

Similarly, if we choose a question of type 2 with $A_n = \mathbb{R} \times B$ then

$$p_{n+1}(u_1, u_2) = \frac{1}{\mathcal{Z}_2}(f_1^{(2)}(y)\,\mathbf{1}_{\{u_2 \in B\}} + f_0^{(2)}(y)\,\mathbf{1}_{\{u_2 \notin B\}}) p_n^{(2)}(u_2) p_n^{(1)}(u_1),$$

where $\mathcal{Z}_2 = f_1^{(2)}(y) p_n^{(2)}(B) + f_0^{(2)}(y)(1 - p_n^{(2)}(B))$.

Proof. The proof is straightforward using Bayes formula, and is similar to the proof of Lemma 1 in the one-dimensional case.

In the two-dimensional setting, any policy can be understood as making two decisions at each time n. The first decision is which coordinate to query, that is, whether to ask a question of type 1 or type 2. Given this choice, the second decision is which question of this type to ask, which corresponds to a finite union of intervals of $\mathbb{R}$. As before, these decisions may depend only upon the information gathered by time n, for which the corresponding sigma-algebra is $\mathcal{F}_n$. For $N > 0$, let S_N be the number of questions of type 1 answered by time N. That is, S_N is the number of $n \in \{0, \ldots, N-1\}$ such that A_n is of the form $A_n = B \times \mathbb{R}$. We take $S_0 = 0$.

We first present a lower bound on the expected decrease in the entropy of each marginal posterior distribution.

Lemma 4. *Under any valid policy π,*

$$\mathrm{E}^\pi[H_1(p_n)] \geq H_1(p_0) - C_1 \mathrm{E}^\pi[S_n], \qquad \mathrm{E}^\pi[H_2(p_n)] \geq H_2(p_0) - C_2(n - \mathrm{E}^\pi[S_n]).$$

Proof. Define $M_n^{(1)} = H_1(p_n) + C_1 S_n$ and $M_n^{(2)} = H_2(p_n) + C_2(n - S_n)$. We will show that $M^{(1)}$ and $M^{(2)}$ are submartingales. Focusing first on $M^{(1)}$, we calculate

$$\mathrm{E}^\pi[M_{n+1}^{(1)} \mid \mathcal{F}_n] = \mathrm{E}^\pi[H_1(p_{n+1}) \mid \mathcal{F}_n] + C_1 S_{n+1}$$

since S_{n+1} is $\mathcal{F}_n$-measurable. We consider two cases. First, if $S_{n+1} = S_n$ (which occurs if A_n is of type 2) then $H_1(p_{n+1}) = H_1(p_n)$ and the $\mathcal{F}_n$-measurability of $H_1(p_n)$ implies that $\mathrm{E}^\pi[M_{n+1}^{(1)} \mid \mathcal{F}_n] = M_n^{(1)}$. Second, if $S_{n+1} = S_n + 1$ (which occurs if A_n is of type 1) then Theorem 2 implies that $\mathrm{E}^\pi[H_1(p_{n+1}) \mid \mathcal{F}_n] \geq H_1(p_n) - C_1$. Hence,

$$\mathrm{E}^\pi[M_{n+1}^{(1)} \mid \mathcal{F}_n] \geq C_1(S_n + 1) + H_1(p_n) - C_1 = M_n^{(1)},$$

which shows that $M_n^{(1)}$ is a submartingale. The proof is similar for $M_n^{(2)}$.

Now, because $M_n^{(1)}$ is a submartingale, $\mathrm{E}^\pi[M_n^{(1)}] \geq M_0^{(1)}$, which implies that $\mathrm{E}^\pi[H_1(p_n)] \geq H_1(p_0) - C_1 \mathrm{E}^\pi[S_n]$. Proceeding similarly for $M_n^{(2)}$ completes the proof.

Consider the following policy, notated π^*. At step n, choose the type of question at random, choosing type 1 with probability $C_2/(C_1 + C_2)$ and type 2 with probability $C_1/(C_1 + C_2)$. Then, in the dimension chosen, choose the subset to be queried according to the one-dimensional dyadic policy.

We show below in Theorem 5 that π^* is optimal for the objective function (23). Before presenting this result, which is the main result of this section, we present an intermediate result concerning the limiting behavior of π^*. This intermediate result is essentially a strong law of large numbers for the objective function (23).

Lemma 5. *Let* $T_N = (1/N)\max(H_1(p_N), H_2(p_N))$. *Under* π^*, *as* $N \to \infty$,

$$T_N \to -\frac{C_1 C_2}{C_1 + C_2} \quad \textit{a.s.} \tag{24}$$

Moreover, there is a constant K *such that* $|T_N| < K$ *for all* N.

Proof. Recall that S_N is the number of questions of type 1 answered by time N, so $S_N/N \to C_2/(C_1+C_2)$ a.s. The law of large numbers established in (19) for the one-dimensional posterior shows that $H_1(p_N)/S_N \to -C_1$ a.s. Combining these two facts shows that $H_1(p_N)/N \to -C_1 C_2/(C_1 + C_2)$ a.s. By a similar argument, $H_2(p_N)/N \to -C_1 C_2/(C_1 + C_2)$ a.s., which shows (24).

We now show the bound on $|T_N|$. Using π^*, according to (18),

$$H_1(p_N) = H_1(p_0) + \sum_{n=1}^{N} Z_n,$$

where the Z_n are independent bounded random variables and

$$|Z_n| \leq |\min(\log(u), \log(1-u))| = \beta.$$

As a consequence, for any $N \geq 1$, $|H_1(p_N)/N| \leq |H_1(p_0)| + \beta$. The same is true for $H_2(p_N)$, which proves that there is a constant K such that $|T_N| < K$.

We now present the main result of this section.

Theorem 5. *The policy π^* is optimal with respect to (23). Moreover, the optimal value is, for any p_0 with $H(p_0) < \infty$,*

$$V(p_0) = -\frac{C_1 C_2}{C_1 + C_2}. \tag{25}$$

Proof. First we show that the value in (25) constitutes a lower bound for $V(p_0)$. Second, we show that (25) is an upper bound on $V(p_0)$ using the properties of the policy π^* presented in Lemma 5.

For the lower bound,

$$\begin{aligned} V(p_0) &\geq \inf_{\pi} \liminf_{N\to\infty} \frac{1}{N} \max(\mathrm{E}^{\pi}[H_1(p_N)], \mathrm{E}^{\pi}[H_1(p_N)]) \\ &\geq \inf_{\pi} \liminf_{N\to\infty} \frac{1}{N} \max(H_1(p_0) - \mathrm{E}[S_N]C_1, H_2(p_0) - (N - \mathrm{E}[S_N])C_2) \\ &= \inf_{0\leq a\leq 1} \max(-aC_1, -(1-a)C_2) \\ &= -\frac{C_1 C_2}{C_1 + C_2}. \end{aligned}$$

We obtained the first line using Jensen's inequality, the second line using Lemma 4, the third line by choosing $a = \liminf_{n\to\infty} \mathrm{E}[S_N]/N$, and the fourth line by recalling that $C_1 > 0$ and $C_2 > 0$.

Now, for the upper bound,

$$\begin{aligned} V(p_0) &\leq \liminf_{N\to\infty} \mathrm{E}^{\pi^*}\left[\max\left(\frac{H_1(p_N)}{N}, \frac{H_2(p_N)}{N}\right)\right] \\ &= \mathrm{E}^{\pi^*}\left[\max\left(\liminf_{N\to\infty} \frac{H_1(p_N)}{N}, \liminf_{N\to\infty} \frac{H_2(p_N)}{N}\right)\right] \\ &= -\frac{C_1 C_2}{C_1 + C_2}. \end{aligned}$$

The uniform bound on T_N from Lemma 5 is sufficient to justify the exchange between the limit and the expected value in going from the first to the second line.

We remark as an aside that in the case where $C_1 = C_2$, this policy is also optimal for the value function (3) since it verifies (11).

We conclude this section by providing a central limit theorem for the objective under this policy π^*.

Theorem 6. *Under π^*,*

$$\lim_{n\to\infty} \frac{1}{\sqrt{n}}\left[\max(H_1(p_n), H_2(p_n)) + \frac{C_1 C_2}{C_1 + C_2} n\right] \stackrel{\mathrm{D}}{=} \frac{\max(\sigma_1\sqrt{C_2}Z_1, \sigma_2\sqrt{C_1}Z_2)}{\sqrt{C_1 + C_2}}. \tag{26}$$

Here, Z_1 and Z_2 are independent standard normal random variables, and σ_i^2 is the variance of the increment of $H_i(p_{n+1}) - H_i(p_n)$ when measuring type i, whose distribution is given by (17).

Proof. For $i = 1, 2$, let $S_{n,i}$ be the number of questions of type i answered by time n, so $S_{n,1} = S_n$ and $S_{n,2} = n - S_n$. Let $t_{s,i} = \inf\{n\colon S_{n,i} = s\}$ for $s = 0, 1, \ldots$. Then $t_{0,i} = 0$ and $\{t_{s,i}\colon s = 1, 2, \ldots\}$ are the times when questions of type i are answered. Thus, each stochastic process $\{H_i(p_{t_{s,i}})\colon s = 0, 1, \ldots\}$ for $i = 1, 2$ has a distribution identical to that of the entropy of

the one-dimensional posterior under the dyadic policy. In addition, the two stochastic processes are independent.

The central limit theorem established in (20) shows that

$$\lim_{s\to\infty} \frac{H_i(p_{t_{s,i}}) + sC_i}{\sqrt{s}} \stackrel{\mathrm{D}}{=} \sigma_i Z_i,$$

where each Z_i is a standard normal random variable and Z_1 is independent of Z_2.

From the definition of $t_{s,i}$,

$$\lim_{s\to\infty} \frac{H_i(p_{t_{s,i}}) + sC_i}{\sqrt{s}} \stackrel{\mathrm{D}}{=} \lim_{n\to\infty} \frac{H_i(p_n) + S_{n,i}C_i}{\sqrt{S_{n,i}}}.$$

Let $j = 1$ when $i = 2$, and $j = 2$ when $i = 1$. Then $\lim_{n\to\infty} S_{n,i}/n = C_j/(C_1 + C_2)$ a.s. and

$$\lim_{n\to\infty} \frac{H_i(p_n) + S_{n,i}C_i}{\sqrt{S_{n,i}}} \stackrel{\mathrm{D}}{=} \lim_{n\to\infty} \frac{H_i(p_n) + nC_1C_2/(C_1 + C_2)}{\sqrt{n}} \sqrt{\frac{C_1 + C_2}{C_j}}.$$

These three facts imply that

$$\lim_{n\to\infty} \frac{H_i(p_n) + nC_1C_2/(C_1 + C_2)}{\sqrt{n}} \stackrel{\mathrm{D}}{=} \sqrt{\frac{C_j}{C_1 + C_2}}\sigma_i Z_i.$$

This proves (26) for the limit.

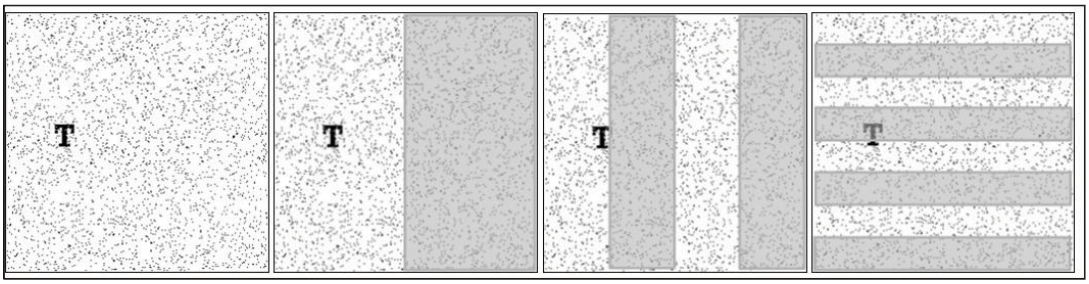
6. LaTeX character localization

In this section we present an application of the dyadic policy to a well-established problem in computer vision: object localization. While the probabilistic bisection policy has already been applied in computer vision, see [10] and [27], the dyadic policy has not, and we feel that it offers considerable promise in this application area.

In the object localization problem, we are given an image and a known object, and must output parameters that describe the *pose* of the object in the image. In the simplest case, the pose is defined by a single pixel, but more complex cases can include, e.g. a rotation angle, a scale factor, or a bounding box. Machine learning techniques have led to the development of classifiers that, given a specific pose, provide accurate answers to the binary question 'Is the object in this pose?'. In our model, we assume these classifiers act as oracles, i.e. are perfect, even though they may occasionally classify incorrectly in practice. Classifiers such as support vector machines [28] and boosting [26] are combined with discriminant features (see, e.g. [18]) to provide the most accurate algorithms (see [29] and [30]). To find the object's pose within an image, classifiers are evaluated at nearly every possible pose, which is computationally costly. We demonstrate that the use of the dyadic policy rather than this brute force approach considerably reduces the computational cost. Although a detailed comparison would be beyond the scope of the illustrative example we present here, the branch and bound algorithm used in [17] is an alternative methodology for reducing computational cost in object localization.

6.1. LaTeX character images, noisy queries, and model estimation

The task we consider is localizing a specific LaTeX character in a binary image. In this setting, an image is a binary matrix $I \in \{0, 1\}^{m\times m}$, where the image has m rows and m columns. A LaTeX character is another smaller binary image $J \in \{0, 1\}^{j\times j}$, where $j < m$. We present experiments where the character of interest, or pattern, is the letter 'T'. We assume that the pattern is always



(a) (b) (c) (d)

FIGURE 4: (a) Example of an image containing the character 'T'. (b)–(d) Examples of subset-based questions. The gray sectors in images (b)–(d) represent the queried regions A_2^1, A_2^2, and A_1^3, respectively.

present in the image, and fully visible (i.e. not occluded by other objects or only partially visible in the image). The goal is to find the location $X^* = (X_1^*, X_2^*)$ of the pixel at the upper-left corner of the pattern within the image.

We generated 1000 images, each of size 256×256 pixels. Each image has a black background (i.e. pixel values of zero), and contains a single fully visible 'T' at a random location in the image. This 'T' is a binary image of size 32×32 pixels (see Figure 4(a)). Noise is added to the image by flipping each pixel value independently with probability 0.1. We then randomly assign each image into one of two sets of approximately equal size: one for training and the other for testing. The training set is used to learn the noise model as described below, and the testing set is used to evaluate the performance of the algorithm.

In this task, querying a set A corresponds to asking whether the upper-left corner of the 'T' resides in this set. We use a simple image-processing technique to provide a noisy answer to this question. The technique we use is chosen for its simplicity, and other more complex image-processing techniques might produce more informative responses, improving the overall performance of the algorithm.

In describing this technique, we first observe that all the images are of size 256×256 pixels and so any pixel coordinate can be represented in base 2 using two 8-bit strings, or octets. For example, the pixel with column–row location (32,14) is represented by (00100000, 00001110). We define 16 sets of pixels. Let A_1^i, $i = 1, \ldots, 8$, be the set of pixels whose column pixel coordinate has a 1 for its ith bit. Similarly, let A_2^i, $i = 1, \ldots, 8$, be the set of pixels whose row pixel coordinate has a 1 for its ith bit. Figure 4(b)–(d) respectively show the sets A_1^1, A_1^2, and A_2^3. For any given image I and set A_j^i, we define the response

$$y(A_j^i) = \sum_{x \in A_j^i} I(x) - \sum_{x \notin A_j^i} I(x), \tag{27}$$

where $I(x) \in \{0, 1\}$ is the binary image's value at pixel x. The motivation for using the response defined by (27) is that $y(A_j^i)$ is more likely to be large when A_j^i contains the 'T'.

Although the response $y(A_j^i)$ is entirely determined by the image I and the location of the 'T' within it, our algorithm models the response using a noise model of the form (1). For simplicity, we assume that both the density f_1 of $y(A)$ when A contains the 'T', and the density f_0 of $y(A)$ when A does not contain the 'T', are normal with respective distributions $N(\mu, \sigma^2)$ and $N(-\mu, \sigma^2)$. The training set is used to estimate these parameters, leading to $\mu = 64.76$ and $\sigma = 105.7$. Because the model is symmetric, $u^* = 0.5$. The channel capacity is estimated with Monte Carlo integration to be $C = 0.23$.

6.2. Prior, posterior, and algorithm

We let $X^* = (X_1^*, X_2^*)$, $X_1^* \in [0, 255]$ and $X_2^* \in [0, 255]$, with p_0 uniform over the domain of X^*. Since the sets A_j^i constrain only one coordinate, the posterior over X^* is a product distribution, as was discussed in Section 5. The posterior for each coordinate $j = 1, 2$ was computed in Lemma 3. We now specialize to the model at hand using the notation '$\propto$' to define equality up to a term that does not depend on x_j:

$$p_8^{(j)}(x_j) \propto \prod_{i=1}^{8} (f_1(y_j^i)\,\mathbf{1}_{\{x_j \in A_j^i\}} + f_0(y_j^i)\,\mathbf{1}_{\{x_j \notin A_j^i\}}),$$

$$\log p_8^{(j)}(x_j) \propto \sum_{\{i\,:\,x_j \in A_j^i\}} \log \frac{f_1(y_j^i)}{f_0(y_j^i)} \propto \sum_{\{i\,:\,x_j \in A_j^i\}} y_j^i.$$

The algorithm has two phases: (i) the noisy query phase and (ii) the noise-free query phase. The noisy query phase comes first, and uses the dyadic policy to obtain a posterior distribution on X^*. The implementation of this noisy query phase is facilitated by the nonadaptive nature of the dyadic policy's questions, which allows us to compute the answers to the questions all at once. The noise-free query phase then uses the posterior resulting from the first phase, together with a sequence of size-limited noise-free questions, to determine the exact location of X^*.

Noisy query phase. Given an image I, we begin by computing $y(A_j^i) = y_j^i$ for each $j = 1, 2$ and $i = 1, \ldots, 8$. We then compute $\ell(x)$ for each pixel x, which is proportional to the logarithm of the posterior density at x,

$$\ell(x) = \sum_{\{i\,:\,x \in A_1^i\}} y_1^i + \sum_{\{i\,:\,x \in A_2^i\}} y_2^i.$$

The top row of Figure 5 shows example images from our test set, while the bottom row of Figure 5 shows the corresponding ℓ-images, in which the value of $\ell(x)$ is plotted for each pixel.

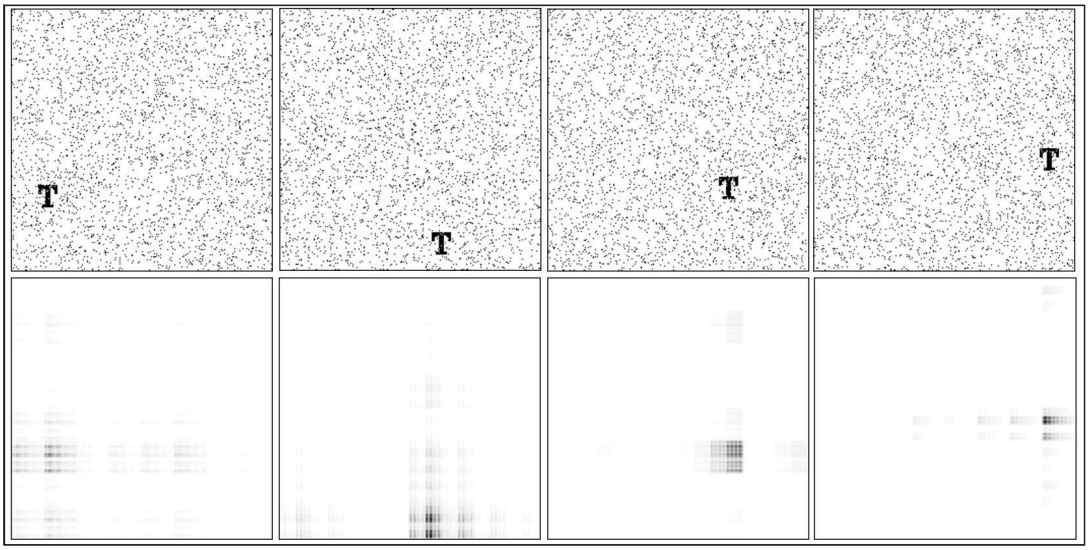


FIGURE 5: Pixel reordering: example images from the test set (*top*) and corresponding ℓ-images (*bottom*). Dark regions indicate pixels more likely to contain the character, while light regions are less likely.

Dark regions of the ℓ-image indicate pixels with large $\ell(x)$, which are more likely to contain the 'T'.

Noise-free query phase. We sort the pixels in decreasing order of $\ell(x)$. We then sequentially perform noise-free evaluations at each pixel x in this order until the true pixel location X^* is found. To perform a noise-free evaluation at a given pixel, we compare the 'T' pattern with the 32×32 pixel square from the image with upper-left corner at x to see if they match. When X^* is found, we stop and record the number of noise-free evaluations performed.

6.3. Results

We validated the algorithm above by evaluating it on the test set described in Section 6.1. To do this, we ran the algorithm on each image and recorded the number of noise-free evaluations required to locate the target character. The results described below (i) demonstrate that the dyadic policy significantly reduces the number of noise-free evaluations required to locate the 'T' character, and (ii) allows us to visualize the results summarized in (11), (19), and (20) within the context of this application.

Recall that each image has $256 \times 256 = 65\,536$ pixels. Over 500 test images, the mean, median, and standard deviation of the number of noise-free evaluations are 2021.5, 647, and 4066.9, respectively. This corresponds to a speed-up factor of 15 over an exhaustive (and typical) search policy. Figure 6(a) shows the sample distribution of the number of noise-free evaluations. We also computed the entropy of the posterior distribution after the 16 noisy questions are answered. According to (11), $\mathrm{E}[H(p_{16})] = H(p_0) - 16C = 16 - 16(0.23) = 12.32$, which is in agreement with the empirically observed value $\mathrm{E}[H(p_{16})] = 12.3$ (with standard deviation 0.716). We also visualized the convergence of the entropy for *each* image, as predicted by the law of large numbers in (19). In Figure 6(b), we plot $H(p_n)/n$, $n = 0, \ldots, 16$, for each image in our test set. The empirical variance at $n = 16$ is very small. Finally, according to (20), the distribution of $(H(p_n) - (H(p_0) - nC))/\sqrt{n}$ should be approximately normal. In Figure 7(a) we present the histogram and in Figure 7(b) we present a normal Q-Q plot, demonstrating close agreement with the normal distribution.

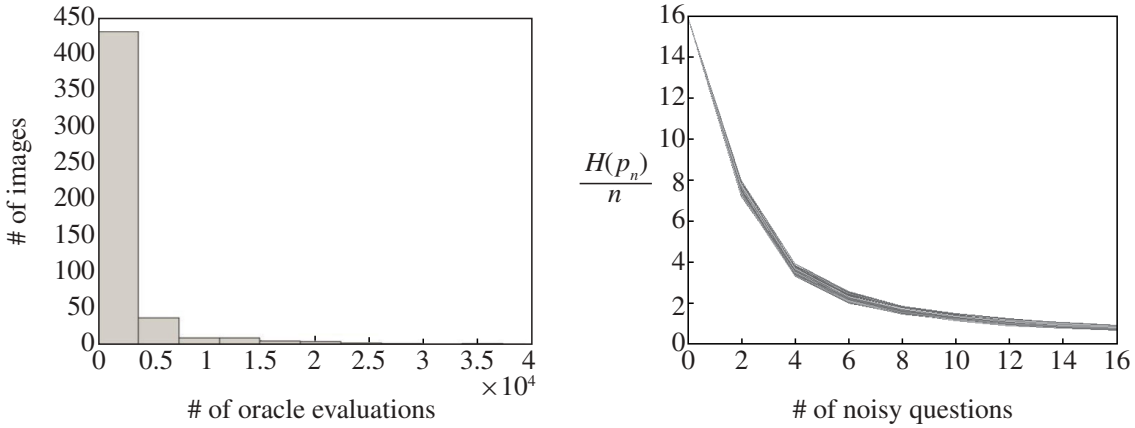


FIGURE 6: Noise-free evaluations and convergence in entropy. (a) The distribution of the number of noise-free evaluations needed to locate the target character. (b) Plot of $H(p_n)/n$ as a function of n. Each line corresponds to one image, with $H(p_n)/n$ plotted over $n = 1, \ldots, 16$. $H(p_n)/n$ converges to $1 - C$.

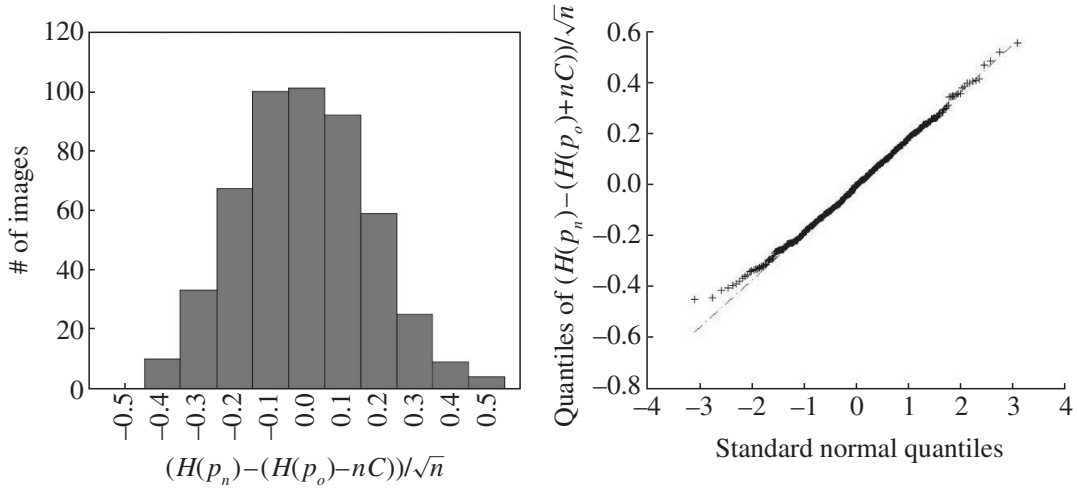


FIGURE 7: Central limit theorem. (a) Distribution of $(H(p_n) - (H(p_0) - nC))/\sqrt{n}$, with mean -0.01. The distribution is close to Gaussian as the Q-Q plot (b) shows.

7. Conclusion

We have considered the problem of twenty questions with noisy responses, which arises in stochastic search, stochastic optimization, computer vision, and other application areas. By considering the entropy as our objective function, we obtained sufficient conditions for Bayes optimality, which we then used to show optimality of two specific policies: probabilistic bisection and the dyadic policy. This probabilistic bisection policy generalizes a previously studied policy, while we believe that the dyadic policy has not been previously considered.

The dyadic policy asks a deterministic set of question, despite being optimal among fully sequential policies. This lends it to applications that allow multiple questions to be asked simultaneously. The structure of this policy also lends itself to further analysis. We provided a law of large numbers, a central limit theorem, and an analysis of the number of noise-free questions required after noisy questioning ceases. We also showed that a generalized version of the dyadic policy is asymptotically optimal in two dimensions for a more robust version of the entropy loss function. We then demonstrated the use of this policy on an example problem from computer vision.

A number of interesting and practically important questions present themselves for future work. First, our optimality results assume the entropy as the objective, but in many applications other objectives are more natural, e.g. the expected number of noise-free questions as in Section 4.3, or the mean-squared error. Second, our results assume that noise is added by a memoryless transmission channel. In many applications, however, the structure of the noise depends upon the questions asked, which calls for generalizing the results herein to this more complex style of noise dependence. We feel that these and other questions will be fruitful areas for further study.

References

[1] BEN-OR, M. AND HASSIDIM, A. (2008). The Bayesian learner is optimal for noisy binary search (and pretty good for quantum as well). In *2008 49th Ann. IEEE Symp. Foundations of Computer Science*. IEEE Computer Society Press, Washington, DC, pp. 221–230.

[2] BERRY, D. A. AND FRISTEDT, B. (1985). *Bandit Problems*. Chapman & Hall, London.

[3] BLUM, J. R. (1954). Multidimensional stochastic approximation methods. *Ann. Math. Statist.* **25,** 737–744.

[4] Burnašhev, M. V. and Zigangirov, K. Š. (1974). A certain problem of interval estimation in observation control. *Problemy Peredachi Informatsii* **10,** 51–61.
[5] Castro, R. and Nowak, R. (2008). Active learning and sampling. In *Foundations and Applications of Sensor Management*, Springer, pp. 177–200.
[6] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley, New York.
[7] DeGroot, M. H. (1970). *Optimal Statistical Decisions*. McGraw Hill, New York.
[8] Dynkin, E. B. and Yushkevich, A. A. (1979). *Controlled Markov Processes*. Springer, New York.
[9] Frazier, P. I., Powell, W. B. and Dayanik, S. (2008). A knowledge-gradient policy for sequential information collection. *SIAM J. Control Optimization* **47,** 2410–2439.
[10] Geman, D. and Jedynak, B. (1996). An active testing model for tracking roads in satellite images. *IEEE Trans. Pattern Anal. Machine Intelligence* **18,** 1–14.
[11] Gittins, J. C. (1989). *Multi-Armed Bandit Allocation Indices*. John Wiley, Chichester.
[12] Horstein, M. (1963). Sequential decoding using noiseless feedback. *IEEE Trans. Inf. Theory* **9,** 136–143.
[13] Horstein, M. (2002). Sequential transmission using noiseless feedback. *IEEE Trans. Inf. Theory* **9,** 136–143.
[14] Karp, R. M. and Kleinberg, R. (2007). Noisy binary search and its applications. In *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms*, ACM, New York, pp. 881–890.
[15] Kushner, H. J. and Yin, G. G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd edn. Springer, New York.
[16] Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.* **6,** 4–22.
[17] Lampert, C. H., Blaschko, M. B. and Hofmann, T. (2009). Efficient subwindow search: a branch and bound framework for object localization. *IEEE Trans. Pattern Anal. Machine Intelligence* **31,** 2129–2142.
[18] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Internat. J. Comput. Vision* **60,** 91–110.
[19] Nowak, R. (2008). Generalized binary search. In *2008 46th Ann. Allerton Conf. Commun., Control, and Computing*, pp. 568–574.
[20] Nowak, R. (2009). Noisy generalized binary search. *Adv. Neural Inf. Processing Systems* **22,** 1366–1374.
[21] Pelc, A. (2002). Searching games with errors—fifty years of coping with liars. *Theoret. Comput. Sci.* **270,** 71–109.
[22] Polyak, B. T. (1990). A new method of stochastic approximation type. *Automat. Remote Control* **51,** 937–946.
[23] Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* **58,** 527–535.
[24] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Ann. Math. Satist.* **22,** 400–407.
[25] Ruppert, D. (1988). Efficient estimators from a slowly convergent Robbins-Monro procedure. Tech. Rep. 781, School of Operations Research and Industrial Engineering, Cornell University.
[26] Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning* **5,** 197–227.
[27] Sznitman, R. and Jedynak, B. (2010). Active testing for face detection and localization. *IEEE Trans. Pattern Anal. Machine Intelligence* **32,** 1914–1920.
[28] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.
[29] Vedaldi, A., Gulshan, V., Varma, M. and Zisserman, A. (2009). Multiple kernels for object detection. In *Proc. Internat. Conf. Computer Vision*, pp. 606–613.
[30] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *Internat. J. Comput. Vision* **57,** 137–154.
[31] Waeber, R., Frazier, P. I. and Henderson, S. G. (2011). A Bayesian approach to stochastic root finding. In Proc. 2011 Winter Simulation Conference, eds S. Jain *et al.,* IEEE.
[32] Whittle, P. (1981). Arm-acquiring bandits. *Ann. Prob.* **9,** 284–292.
[33] Whittle, P. (1988). Restless bandits: activity allocation in a changing world. In *A Celebration of Applied Probability* (J. Appl. Prob. Spec. Vol. **25A**), ed. J. Gani, Applied Probability Trust, Sheffield, pp. 287–298.

Chapter 4

Twenty Questions for Localizing Multiple Objects by Counting: Bayes Optimal Policies for Entropy Loss

Twenty Questions for Localizing Multiple Objects by Counting: Bayes Optimal Policies for Entropy Loss

Weidong Han[1], Peter I. Frazier[2] and Bruno M. Jedynak[1]

[1]Department of Applied Mathematics and Statistics, Johns Hopkins University
[2]School of Operations Research and Information Engineering, Cornell University

May 20, 2014

Abstract

We consider the problem of twenty questions with noiseless answers, in which we aim to locate multiple objects by querying the number of objects in each of a sequence of chosen sets. We assume a joint Bayesian prior density on the locations of the objects and seek to choose the sets queried to minimize the expected entropy of the Bayesian posterior distribution after a fixed number of questions. An optimal policy for accomplishing this task is characterized by the dynamic programming equations, but the curse of dimensionality prevents its tractable computation. We first derive a lower bound on the performance achievable by an optimal policy. We then provide explicit performance bounds relative to optimal for two computationally tractable policies: greedy, which maximizes the one-step expected reduction in entropy; and dyadic, which splits the search domain in successively finer partitions. We also show that greedy performs at least as well as the dyadic policy. This can help when choosing the policy most appropriate for a given application: the dyadic policy is easier to compute and nonadaptive, allowing its use in parallel settings or when questions are inexpensive relative to computation; while the greedy policy is more computationally intensive but also uses questions more efficiently, making it the better choice when robust sequential computation is possible. Numerical experiments demonstrate that both procedures outperform a divide-and-conquer benchmark policy from the literature, called sequential bifurcation. Finally, we further characterize performance under the dyadic policy by showing that the entropy of the posterior distribution is asymptotically normal.
.

1 Introduction

We consider the following set-guessing problem. Let $\Omega = \mathbb{R}$ be the real line and $\theta = (\theta_1, \ldots, \theta_k) \in \Omega^k$ be a vector containing the unknown locations of k objects, where $k \geq 1$ is known. One can sequentially choose subsets $A_1, A_2, \ldots$ of Ω, query the number of objects in each set, and obtain a series of noiseless answers $X_1, X_2, \ldots$. In studying this problem, our goal is to devise a method

We plan a parallel submission of a short (eight-page) summary of this work to the 2014 Neural Information Processing Systems (NIPS) conference, which will also include numerical simulations and further discussions of the potential applications of the material. If it is accepted at NIPS, we will reference it in the introduction. This is in accordance with IEEE Transactions on Information Theory's policy on prior publication. See `http://www.comm.utoronto.ca/trans-it/author-info.shtml`.

for choosing the questions that allows us to find θ as accurately as possible, given a finite budget of questions. We work in a Bayesian setting, and use the entropy of the posterior distribution on θ to measure accuracy.

While the adaptive method with minimal expected posterior entropy is described by the dynamic programming principle, and could in principle be computed using dynamic programming, current computational techniques do not allow doing this in a tractable way. In this paper, we provide a lower bound on this minimal expected entropy; and analyze two specific methods, providing in one case an explicit expression for the expected entropy, and in the other case a tractable upper bound.

The previous literature on similar problems can be classified into two groups: those that consider a single object ($k = 1$); and those that consider multiple objects ($k \geq 1$).

Among single-object versions of this problem, the earliest is the Rényi-Ulam game [1, 2]. In this game, one person (the responder) thinks of a number between one and one million and another person (the questioner) chooses a sequence of subsets to query in order to find this number. The responder can answer either YES or NO and is allowed to lie a given number of times.

Variations of the Renyi-Ulam game have been considered in [3]. Among these variations, the following continuous probabilistic version, first studied in [4], is similar to the problem we consider: The responser thinks of a number $\theta \in [0, 1]$ and the questioner aims to find a set $A \subset [0, 1]$ with measure less than ϵ such that $\theta \in A$ with probability at least q. In addition, the responser lies with probability no more than p. Whether the questioner can win this game based on the error probability p is analyzed and searching algorithms using $O(\log \frac{1}{\epsilon})$ queries are provided.

Among previous work on the single-object problem, perhaps the closest to the current work is [5], which considered a Bayesian setting and used the entropy of the posterior distribution to measure of accuracy, as we do here. It considered two policies, a greedy policy called probabilistic bisection, which was originally proposed in [6] and further studied in [7, 8], and the dyadic policy. [9] generalized the probabilistic bisection policy to multiple questioners. Here, we generalize both policies to multiple objects.

Our work contrasts with this previous work on the single-object problem by considering multiple objects.

The previous literature includes work on three multiple-object problems: the Group Testing problem [10, 11, 12, 13, 14]; the subset-guessing game associated with the Random Chemistry algorithm [15, 16]; and the Guessing Secret game [17]. In the Group Testing problem, questions are of the form: *Is* $A \cap S \neq \emptyset$*?* In the subset-guessing game associated with the Random Chemistry algorithm, questions are of the form *Is* $S \subset A$*?*. In the Guessing Secret game, when queried with a set A, the responder chooses an element from S according to any rule that he likes, and tells the questioner whether this chosen element is in A. The chosen element itself is not revealed, and may change after each question. Thus, the answer is 1 when $S \subset A$, 0 when $A \cap S = \emptyset$, and can be either 0 or 1 otherwise.

Our work contrasts with this previous work by considering a problem where the answer provided by the responser is not binary but instead counts the number of objects in the queried set.

These multiple-object-localization games find application in constructions of block codes [18, 19], searching for auto-catalytic sets of molecules [15], searching for collections of multiple contingencies leading to cascading power failures in models of electrical networks [20], computer vision [21, 22, 23], and screening for stochastic simulation [24, 25].

Now, in Section 2, we state the problem more formally, and summarize our main results.

2 Problem Formulation and Summary of Main Results

Let $\theta = (\theta_1, \ldots, \theta_k)$ be a random vector taking values in $\mathbb{R}^k$. θ_i represents the location of the ith object of interest, $i = 1, \ldots, k$. We assume that $\theta_1, \ldots, \theta_k$ are i.i.d. with density f_0, and joint density p_0. We refer to p_0 as the Bayesian prior probability distribution on θ. We will ask a series of $N > 0$ questions to locate $\theta_1, \ldots, \theta_k$, where each question takes the form of a subset of $\mathbb{R}$, and the answer to this question is the number of objects in this subset. More precisely, for each $n \in \{1, 2, \ldots, N\}$, the n^{th} question is $A_n \subset \mathbb{R}$ and its answer is

$$X_n = \mathbb{1}_{A_n}(\theta_1) + \cdots + \mathbb{1}_{A_n}(\theta_k), \tag{1}$$

where $\mathbb{1}_A$ is the indicator function of the set A. Our choice of the set A_n may depend upon the answers to all previous questions, and upon some initial randomization through a uniform random variable Z on $[0, 1]$ chosen independently of θ. Thus, the set A_n is random, through its dependence on Z, and the answers to previous questions.

We call a rule for choosing the questions A_n a *policy*. Formally, we define a policy π to be a sequence $\pi = (\pi_1, \ldots, \pi_N)$, where π_n is a Borel-measurable subset of $[0, 1] \times \{0, 1, \ldots, k\}^{n-1} \times \mathbb{R}$. With a policy π specified, the choice of A_n is then $A_n = \{t \in \mathbb{R} : (Z, X_{1:n-1}, t) \in \pi_n\}$, so that specifying π_n implicitly specifies a rule for choosing A_n based on the random seed Z and the history $X_{1:n-1}$. Here, we have used the notation $X_{a:b}$ for any natural numbers a and b to indicate the sequence $(X_a, \ldots, X_b)$ if $a \geq b$, and the empty sequence if $a < b$. We define $\theta_{a:b}$ and $A_{a:b}$ similarly. The distribution of A_n thus implicitly depends on π. When we wish to highlight this dependence, we will use the notation P^π and E^π to indicate probability and expectation respectively. However, when the policy being studied is clear, we will simply use P and E .

We refer to the posterior probability distribution on θ after n questions as p_n, so p_n is the conditional distribution of θ given $X_{1:n}$ and $A_{1:n}$. Equivalently, under any fixed policy π, p_n is the conditional distribution of θ given Z and $X_{1:n}$. This posterior p_n can be computed using Bayes rule: $p_n(u)$ is proportional to $p_0(u)$ over the set $\left\{u \in \mathbb{R}^k : X_m = \sum_{i=1}^k \mathbb{1}_{A_m}(u_i),\ 1 \leq m \leq n\right\}$, and 0 outside. The dependence on Z arises because A_n may depend on Z, in addition to $X_{1:n-1}$.

After we exhaust our budget of N questions, we will measure the quality of what we have learned from them via the differential entropy $H(p_N)$ of the posterior distribution p_N on θ at this final time,

$$H(p_N) = -E[\log p_N] = -\int_{\mathbb{R}^k} p_N(u_{1:k}) \log(p_N(u_{1:k}))\, du_{1:k}. \tag{2}$$

Throughout this paper, we use "log" to denote the logarithm to base 2. We let $H_0 = H(p_0)$, and we assume $-\infty < H(p_0) < +\infty$. The posterior distribution p_N, as well as its entropy $H(p_N)$, are random for $N > 0$, as they depend on $X_{1:N}$ and Z. Thus, we measure the quality of a policy $\pi \in \Pi$ when given N questions using

$$R(\pi, N) = E^\pi[H(p_N)]. \tag{3}$$

Our goal in this paper is to characterize the solution to the optimization problem

$$\inf_{\pi \in \Pi} R(\pi, N). \tag{4}$$

Any policy that attains this infimum is called *optimal*.

While (4) can be formulated as a partially observable Markov decision process [26], and can be solved, in principle, via dynamic programming, the state space of this dynamic program is the space of posterior distributions over θ, and the extreme size of this space prevents solving this dynamic program through brute-force computation.

Thus, in this paper, rather than attempting to compute the optimal policy, we provide an easily computed lower bound on (4), and then study two classes of policies relative to this lower bound: greedy policies, and dyadic policies.

By a *greedy policy*, we mean any policy that chooses each of its questions to minimize the expected entropy of the posterior distribution one step forward in time,

$$A_n \in \arg\min_A E[H(p_n)|p_{n-1}, A_n = A], \text{for all } n = 1, 2, \ldots, N, \tag{5}$$

where the argmin is taken over all Borel-measurable subsets of $\mathbb{R}$. We show in Section 6 that this argmin exists.

To define the *dyadic policy*, let us recall that the quantile function of θ_1 is

$$Q(p) = \inf\{u \in \mathbb{R} : p \leq F_0(u)\}, \tag{6}$$

where F_0 is the cumulative distribution function of θ_1, corresponding to its density f_0. The dyadic policy consists in choosing at step $n \geq 1$ the set

$$A_n = \left(\bigcup_{j=0}^{2^{n-1}-1} \left(Q\left(\frac{2j+1}{2^n}\right), Q\left(\frac{2j+2}{2^n}\right)\right] \right) \bigcap \text{supp}(f_0), \tag{7}$$

where $\text{supp}(f_0)$ is the support of f_0, i.e., the set of values $u \in \mathbb{R}$ for which $f_0(u) > 0$. For example, when f_0 is uniform over $(0, 1]$, the dyadic policy is the one in which the first question is $A_1 = \left(\frac{1}{2}, 1\right]$, the second question is $A_2 = \left(\frac{1}{4}, \frac{1}{2}\right] \cup \left(\frac{3}{4}, 1\right]$, ..., and each subsequent question is obtained by subdividing $(0, 1]$ into 2^n equally sized subsets, and including every second subset. A further illustration of the dyadic question sets A_n is provided in Figure 3 in Section 5. This definition of the dyadic policy generalizes a definition provided in [5] for single objects.

We are now ready to present our main results:

$$H_0 - \log(k+1)N \leq \inf_{\pi \in \Pi} R(\pi, N) \leq R(\pi_G, N) \leq R(\pi_D, N) = H_0 - H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right) N, \tag{8}$$

where π_G is any greedy policy, π_D is the dyadic policy, and Bin indicates the binomial distribution.

The first inequality in (8) is an information theoretic inequality (proved in Section 3). The second inequality is trivial since an optimal policy is at least as good as any other policy. The third inequality comes from a detailed computation of the posterior distribution p_N of θ after observing N answers for any possible sequence of N questions (see Section 6.2). Additionally, we show that this inequality cannot be reversed, by presenting a special case in which there is a greedy policy whose performance is strictly better than that of the dyadic policy (see Section 6.3). The last equality comes from the characterization of the posterior distribution p_N in the special case of the dyadic policy (see Section 5.2).

The power of these results is illustrated by Figure 1, which shows, as a function of the number of objects k, the number of questions required to reduce the expected entropy of the posterior on

their locations by 20 bits per object. The figure shows the number of questions needed under the dyadic policy (solid line, and right-most expression in (8)); under two benchmark policies described below, Benchmark 1 and Benchmark 2 (dotted, and dash-dotted lines); and a lower bound on the number needed under the optimal policy (dashed line, and left-most expression in (8)). By (8), we know that the number of extra questions required by using either the dyadic or the greedy, instead of the optimal policy, is bounded above by the distance between the solid and dashed lines.

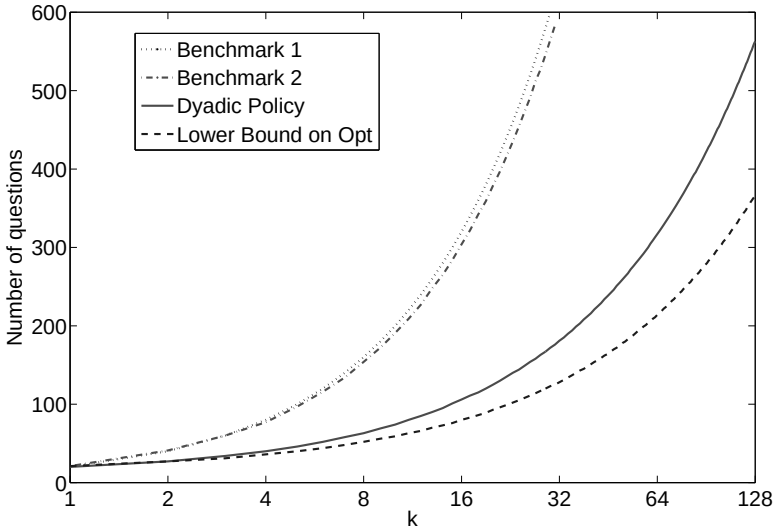


Figure 1: Number of questions needed to reduce the entropy by 20 bits per object under two benchmark policies and the dyadic policy, and a lower bound on the number under the optimal policy. The dyadic policy significantly outperforms both benchmarks and its performance is relatively close to the lower bound on the optimal possible from (8). The performance of the greedy policy is between that of the dyadic and optimal policies.

Benchmark 1 identifies each object individually, using an optimal single-object strategy. It first asks questions to localize the first object θ_1, reducing the entropy of our posterior distribution on that object's location by 20 bits. This requires 20 questions, and can be achieved, for example, by a bisection policy, [6]. It then uses the same strategy to localize each subsequent second object, requiring 20 questions per object[2]. The total number of questions required under this policy to achieve 20 bits of entropy reduction per object is $20k$.

Benchmark 2 is adapted from the sequential bifurcation policy of [25]. While [25] considered an application setting somewhat different from the problem that we consider here (screening for discrete event simulation), we were able to modify their policy to allow it to be used in our setting. A detailed description of the modified policy is provided in Appendix A. It makes full use of the ability to ask questions about multiple objects simultaneously, and improves slightly over Benchmark 1. We view this policy as the best previously proposed policy from the literature for solving the problem that we consider.

[2]Implementing Benchmark 1 would require the ability to ask questions about whether or not a single specified object (e.g., object θ_1) resides in a queried set, rather than the number of objects in that set. While this ability is not included in our formal model, Benchmark 1 is nevertheless a useful comparator.

The figure shows that a substantial saving over both benchmarks is possible through the dyadic or greedy policy. For example, for $k = 2^4 = 16$ objects, Benchmark 1 and Benchmark 2 require 320 and 304 questions respectively. In contrast, the dyadic policy requires 106 questions, which is nearly 3 times smaller than required by the benchmarks. Furthermore, (8) shows that the greedy policy performs at least as well as the dyadic policy. Thus, localizing objects' locations jointly can be much more efficient than localizing them one-at-a-time, and the dyadic and greedy policies are implementable policies that can achieve much of the potential efficiency gains.

The figure also shows, again at $k = 2^4 = 16$ objects, that the optimal policy requires at least 80 questions, while the dyadic and greedy require no more than 106 questions, and so are within a factor of 1.325 of optimal. This is remarkable, when we compare how little is lost when going from the hard-to-compute optimal policy to the easily computed dyadic policy, with how much is gained by going to the dyadic from one of the two benchmark policies considered.

The dyadic policy can be computed extremely quickly, and can even be pre-computed, as the questions asked do not depend on the answers to previous questions. This makes it convenient in settings where multiple questions can be asked simultaneously, e.g., in a parallel or distributed computing environment. The greedy policy requires more computational effort than the dyadic policy, but is still substantially easier to compute than the optimal policy, and provides performance at least as good as that of the dyadic policy, as shown by (8), and sometimes strictly better, as will be shown in Section 6.3.

We see in the figure that the dyadic policy's value and the value of the optimal policy come together at $k = 1$. This can also be seen directly from our theoretical results. When $k = 1$, the left-hand and right-hand sides of (8) are equal, since $\text{Bin}\left(k, \frac{1}{2}\right)$ becomes a $\text{Bernoulli}(\frac{1}{2})$ random variable, whose entropy is $\log(2) = 1$. This shows, when $k = 1$, that the expected entropy reduction under the dyadic is the same as the lower bound on this reduction under the optimal policy, which in turn shows that both dyadic and greedy policies are optimal, and the lower bound is tight. This result can also seen through results obtained in [5]. When $k = 1$, the well-known bisection policy is a greedy policy, and the dyadic is also greedy, i.e., satisfies (5).

We begin our analysis in Section 3, by justifying the left-most inequality in (8). We then provide an explicit expression for the posterior distribution in Section 4, which is used in later analysis. We analyze the dyadic policy in Section 5, and the greedy policy in Section 6. Finally, we offer concluding remarks in Section 7.

3 A Lower Bound on the Expected Entropy after a Fixed Number of Questions and Answers

In this section, below in Theorem 1, we prove the first inequality in (8), which is a lower bound on the expected entropy after a fixed number of questions and answers.

We first introduce some notation, used here, and throughout the paper. For any pair of random variables W, V, we define $H(W\|V)$ to be the random variable taking the value

$$-\int_{-\infty}^{\infty} f(w|V=v)\log f(w|V=v)\,dw \tag{9}$$

for each $V = v$, assuming the conditional density function $f(w|V = v)$ exists. The "usual" conditional entropy is related to it by

$$H(W|V) = E[H(W\|V)]. \tag{10}$$

We now provide here, in Lemma 1, an expression for the expected entropy after additional questions. This lemma is based on the idea that each additional question reduces the entropy of $\theta_{1:k}$ by an amount that can be expressed in terms of the conditional entropy of the answer to that question. The total entropy reduction can then be computed as a sum of the contributions from each question, which we use later to study the expected total entropy reduction under specific policies.

Lemma 1. *Under any policy π,*

$$E[H(p_{n+1})|B_n] = H(p_n) - H(X_{n+1}\|B_n), \text{ for all } n = 0, 1, \ldots, N-1, \tag{11}$$

where $B_n = (Z, X_{1:n})$ denotes the random vector in the history observable before asking the question A_{n+1}, which is deterministic once $B_n = b_n$ is fixed. Moreover,

$$E[H(p_N)] = H_0 - \sum_{n=0}^{N-1} H(X_{n+1}|B_n). \tag{12}$$

Proof. First of all, we prove the recursive relation (11). $H(p_n)$ is the entropy of the posterior distribution of θ, which is random through its dependence on the past history B_n, hence we can rewrite it as $H(p_n) = H(\theta\|B_n)$. Similarly, $H(p_{n+1}) = H(\theta\|B_{n+1}) = H(\theta\|B_n, X_{n+1})$. Since all three terms in (11) are $\sigma(B_n)$-measurable random variables, it suffices to prove (11) holds for any fixed history $B_n = b_n$, i.e.

$$E[H(\theta\|B_n, X_{n+1})|B_n = b_n] = H(\theta|B_n = b_n) - H(X_{n+1}|B_n = b_n). \tag{13}$$

Using information theoretic arguments, we have

$$\begin{aligned}
E[H(\theta\|B_n, X_{n+1})|B_n = b_n] &= \sum_{x_{n+1}=0}^{k} H(\theta|B_n = b_n, X_{n+1} = x_{n+1}) P(X_{n+1} = x_{n+1}|B_n = b_n) && \text{(14a)}\\
&= H(\theta|X_{n+1}, B_n = b_n) && \text{(14b)}\\
&= H(\theta, X_{n+1}|B_n = b_n) - H(X_{n+1}|B_n = b_n) && \text{(14c)}\\
&= H(X_{n+1}|\theta, B_n = b_n) + H(\theta|B_n = b_n) - H(X_{n+1}|B_n = b_n) && \text{(14d)}\\
&= H(\theta|B_n = b_n) - H(X_{n+1}|B_n = b_n) && \text{(14e)}
\end{aligned}$$

where (14b) comes from the definition of conditional entropy and (14c), (14d) come from the chain rule for conditional entropy. (14e) holds as the first term in (14d) vanishes because the information of θ completely determines the answer X_{n+1}. This proves (13).

Now, in order to prove (12), let us first obtain a recursive relation in unconditional expected entropy of posterior distributions. Taking the expectation over B_n on both sides of (11),

$$E\left[E[H(p_{n+1})|B_n]\right] = E[H(p_n)] - E\left[H(X_{n+1}\|B_n)\right]. \tag{15}$$

Note that $E\left[E[H(p_{n+1})|B_n]\right] = E[H(p_{n+1})]$ by the iterated conditioning property of conditional expectation. Moreover, $E\left[H(X_{n+1}\|B_n)\right] = H(X_{n+1}|B_n)$ according to the definition of conditional entropy in (10). Hence, (15) is equivalent to

$$E[H(p_{n+1})] = E[H(p_n)] - H(X_{n+1}|B_n). \tag{16}$$

Applying (16) iteratively for $n = N-1, \ldots, 0$, we obtain (12), which concludes the proof. □

Now, applying (12) in Lemma 1 and using an information theoretic argument, we are able to show the first inequality in our main result (8).

Theorem 1.

$$\inf_{\pi\in\Pi} R(\pi, N) \geq H_0 - \log(k+1)N. \tag{17}$$

Moreover, when $k > 1$, this inequality is strict.

Proof. Since conditioning always reduces entropy, we have

$$H(X_{n+1}|B_n) \leq H(X_{n+1}), \text{ for all } n = 0, 1, \ldots, N-1. \tag{18}$$

Combining (12) with (18), the expected entropy must satisfy

$$E[H(p_N)] \geq H_0 - \sum_{n=1}^{N} H(X_n). \tag{19}$$

Recall that for all $n = 1, 2, \ldots, N$, X_n is a discrete random variable with $k+1$ possible outcomes, namely $0, 1, \ldots, k$. The maximum possible value for the entropy $H(X_n)$ is $\log(k+1)$, obtained when each outcome of X_n has the same probability $\frac{1}{k+1}$, i.e. $H(X_n) \leq \log(k+1)$. Thus, by (19),

$$E[H(p_N)] \geq H_0 - \log(k+1)N. \tag{20}$$

Since (20) is true for any policy π, and indicating the dependence of $E[H(p_N)]$ on the policy π in our notation, we have

$$\inf_{\pi\in\Pi} R(\pi, N) = \inf_{\pi\in\Pi} E^{\pi}[H(p_N)] \geq H_0 - \log(k+1)N. \tag{21}$$

This proves our claim (17).

We now prove that the inequality (17) is strict when $k > 1$, i.e. when there is more than one object. Consider any fixed $B_0 = Z = z$, which specifies the questions set A_1. Recall from (1) that $X_1 = \mathbb{1}_{A_1}(\theta_1) + \cdots + \mathbb{1}_{A_1}(\theta_k)$ and that $\theta_1, \ldots, \theta_k$ are independent. As a consequence, $X_1 \mid Z = z \sim \text{Bin}(k, p)$, where $p = \int_{A_1} f_0(u)\, du$. Therefore, $H(X_1|Z = z) = H(\text{Bin}(k, p)) < \log(k+1)$ when $k > 1$, implying $H(X_1|B_0) < \log(k+1)$, so that there is no policy that can achieve the lower bound. □

4 Explicit Characterization of the Posterior Distribution

In this section, we first derive in Section 4.1 an explicit formula for the posterior distribution on the locations of the objects, and introduce some additional notation. We then provide in Section 4.2 an example illustrating this notation and the posterior distribution. This example also will be used later, in Section 6.3, to show that greedy is sometimes strictly better than dyadic. Finally, in Section 4.3, we compute the conditional distribution of the next answer X_n given previous answers $X_{1:n-1}$, which we will use later to analyze the value of a policy.

4.1 The Posterior Distribution of the Objects

Consider a fixed n, where $1 \leq n \leq N$. For each binary sequence of length n, $s = \{s_1, \ldots, s_n\}$, let

$$C_s = \left(\bigcap_{1 \leq j \leq n; s_j = 1} A_j \right) \bigcap \left(\bigcap_{1 \leq j \leq n; s_j = 0} A_j^c \right) \bigcap \text{supp}(f_0). \tag{22}$$

The collection $\{C_s : C_s \neq \emptyset, s \in \{0,1\}^n\}$ is a partition of the support of f_0. A history of n questions provides information on which sets C_s contain which objects among $\theta_{1:k}$.

We will think of a sequence of binary sequences $s^{(1)}, \ldots, s^{(k)}$ as a sequence of codewords indicating the sets in which each of the objects $\theta_{1:k}$ reside, i.e, indicating that θ_1 is in $C_{s^{(1)}}$, θ_2 is in $C_{s^{(2)}}$, etc. We may consider each binary sequence $s^{(1)}, \ldots, s^{(k)}$ to be a column vector, and place them into an $n \times k$ binary matrix, $\mathcal{S}$. This binary matrix then codes the location of all k objects, and is a codeword for their joint location.

Moreover, to characterize the location of the random vector $\theta = (\theta_{1:k})$ in terms of its codeword $\mathcal{S}$, define $C_{\mathcal{S}} \subset \mathbb{R}^k$ to be the Cartesian product

$$C_{\mathcal{S}} = C_{s^{(1)}} \times \cdots \times C_{s^{(k)}}. \tag{23}$$

To be consistent with an answer X_j, we must have exactly X_j objects located in the question set A_j for each $1 \leq j \leq n$. This can be described in terms of a constraint on the matrix $\mathcal{S}$ as $s_j^{(1)} + \cdots + s_j^{(k)} = X_j$, i.e., that the sum of the j^{th} row in the matrix $\mathcal{S}$ is X_j. Thus, after observing the answers to the questions $X_{1:n} = x_{1:n}$, the set of all possible joint codewords describing $\theta_{1:k}$ is

$$E_n = \{\mathcal{S} | s^{(1)}, \ldots, s^{(k)} \in \{0,1\}^n, C_{s^{(1)}}, \ldots, C_{s^{(k)}} \neq \emptyset, s_j^{(1)} + \cdots + s_j^{(k)} = x_j, \text{for all } 1 \leq j \leq n\}. \tag{24}$$

An example will be provided in Section 4.2 to illustrate this construction.

Given this notation, we observe the following lemma:

Lemma 2. *Let the random seed $Z = z$ be fixed. Then, for each $x_{1:n}$, the event $\{X_{1:n} = x_{1:n}\}$ can be rewritten*

$$\{X_{1:n} = x_{1:n}\} = \left\{ \theta \in \bigcup_{\mathcal{S} \in E_n} C_{\mathcal{S}} \right\}, \tag{25}$$

where we recall that E_n depends on $x_{1:n}$ and z. Moreover for any $\mathcal{S}, \mathcal{T} \in E_n$ with $\mathcal{S} \neq \mathcal{T}$, the two sets $C_{\mathcal{S}}$ and $C_{\mathcal{T}}$ are disjoint.

Proof. Clearly, according to the definition of E_n in (24), when $\theta \in \bigcup_{\mathcal{S} \in E_n} C_{\mathcal{S}}$, the answers that we observe must satisfy $X_{1:n} = x_{1:n}$. On the other hand, suppose $\theta_{1:k} \notin \bigcup_{\mathcal{S} \in E_n} C_{\mathcal{S}}$. Then $\theta_{1:k}$ belongs to some nonempty set $C_{\mathcal{S}}$ where $\mathcal{S} \notin E_n$. Hence, there exists j, $1 \leq j \leq n$, such that $s_j^{(1)} + \cdots + s_j^{(k)} \neq x_j$, which implies that the answer to the question A_j is $X_j = s_j^{(1)} + \cdots + s_j^{(k)} \neq x_j$. This proves (25).

Now, for any $\mathcal{S} \neq \mathcal{T}$, there exists i with $1 \leq i \leq k$ such that $s^{(i)} \neq t^{(i)}$. This implies that $C_{s^{(i)}}$ and $C_{t^{(i)}}$ are disjoint and the last assertion follows. □

At this point, the explicit characterization of the posterior distribution is immediate and we have the following lemma.

Lemma 3.

$$p_n(u_{1:k}) = \frac{p_0(u_{1:k})}{p_0\left(\bigcup_{\mathcal{S}\in E_n} C_{\mathcal{S}}\right)} \text{ ,for } u_{1:k} \in \bigcup_{\mathcal{S}\in E_n} C_{\mathcal{S}}, \tag{26}$$

and $p_n(u_{1:k}) = 0$ *for* $u_{1:k} \notin \bigcup_{\mathcal{S}\in E_n} C_{\mathcal{S}}$. *Here, for any measurable set* A, $p_0(A)$ *denotes the integral* $\int_A p_0(u_{1:k})\, du_{1:k}$. *Moreover,*

$$p_0\left(\bigcup_{\mathcal{S}\in E_n} C_{\mathcal{S}}\right) = \sum_{\mathcal{S}\in E_n} p_0(C_{\mathcal{S}}) = \sum_{\mathcal{S}\in E_n} f_0(C_{s^{(1)}}) \dots f_0(C_{s^{(k)}}), \tag{27}$$

where $f_0(C_{s^{(i)}})$ *denotes the integral* $\int_{C_{s^{(i)}}} f_0(u)\, du$.

4.2 Examples Illustrating the Posterior Distribution

To illustrate the previous construction, and also to provide the foundation for a later analysis in Section 6.3 showing the greedy policy is strictly better than the dyadic policy in some settings, we provide two examples of the posterior distribution, arising from two different responses to the same sequence of questions.

Suppose θ_1, θ_2 are two objects located in (0,1] with a uniform prior distribution f_0. Let A_1 and A_2 be the first two questions of the dyadic policy, so $A_1 = \left(\frac{1}{2}, 1\right]$ and $A_2 = \left(\frac{1}{4}, \frac{1}{2}\right] \cup \left(\frac{3}{4}, 1\right]$. Then consider two possibilities for the answers to these questions:

Example 1: Suppose $X_1 = 0$ and $X_2 = 2$. According to (24), there is only one matrix $\mathcal{S}$ in the collection E_2, which has $s^{(1)} = s^{(2)} = (0, 1)^T$. Thus $E_2 = \{\mathcal{S}_1\}$ where

$$\mathcal{S}_1 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}. \tag{28}$$

By (26) in Lemma 3, we have that $p_2(u_{1:2}) = 16$ when $u_{1:2}$ is in $\left(\frac{1}{4}, \frac{1}{2}\right] \times \left(\frac{1}{4}, \frac{1}{2}\right]$, and 0 otherwise.

Example 2: Suppose $X_1 = 1$ and $X_2 = 1$. According to (24), there are four matrices in the collection $E_2 = \{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$,

$$\mathcal{S}_1 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \mathcal{S}_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathcal{S}_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathcal{S}_4 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}. \tag{29}$$

By (26) in Lemma 3, the posterior distribution has density $p_2(u_{1:2}) = 16$ when $u_{1:2}$ is in $\left(0, \frac{1}{4}\right] \times \left(\frac{3}{4}, 1\right]$ or $\left(\frac{1}{4}, \frac{1}{2}\right] \times \left(\frac{1}{2}, \frac{3}{4}\right]$ or $\left(\frac{1}{2}, \frac{3}{4}\right] \times \left(\frac{1}{4}, \frac{1}{2}\right]$ or $\left(\frac{3}{4}, 1\right] \times \left(0, \frac{1}{4}\right]$, and is 0 otherwise.

All possible joint locations of θ_1, θ_2 in the two examples above are shown in Figure 2.

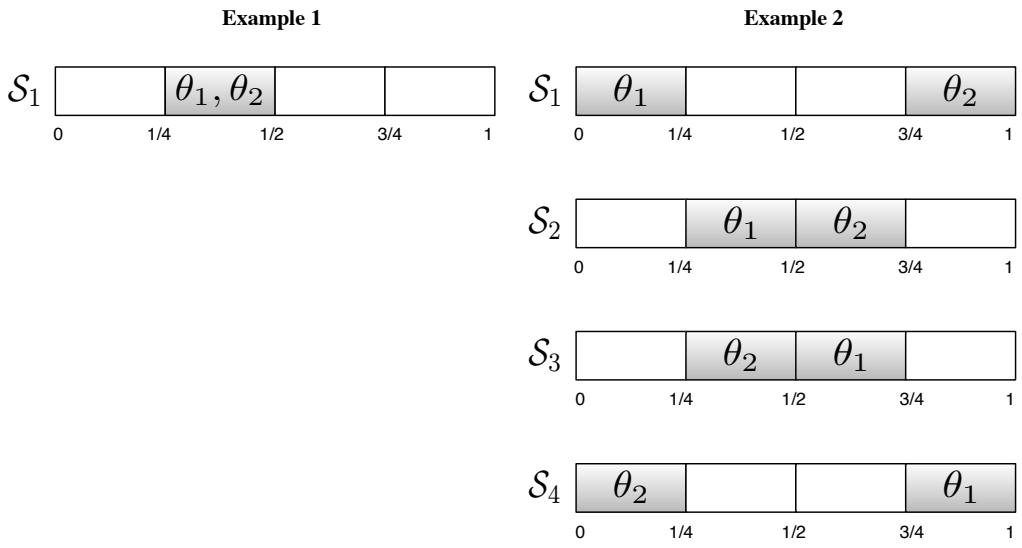


Figure 2: Illustration of the locations of the two objects θ_1, θ_2 specified by each matrix given in (28) and (29). The dark subsets mark the location of the objects θ_1, θ_2.

4.3 The Posterior Predictive Distribution of X_{n+1}

We now provide an explicit form for the posterior predictive distribution of X_{n+1}, i.e., its conditional distribution given the history $X_{1:n}$ and the external source of randomness in the policy Z. This is useful because Lemma 1 shows that the expected entropy $E[H(p_N)]$ can be computed using the conditional entropy of X_{n+1} given $B_n = (Z, X_{1:n})$. We use this in Sections 5.2 and 6.2 to compute the expected entropy for the dyadic and greedy policies respectively.

For $n = 0$, we have demonstrated in the proof of Theorem 1 that X_1 follows the binomial distribution $\text{Bin}(k, f_0(A_1))$ given Z.

Now, consider any $n \in \{1, 2, \ldots, N-1\}$, and any fixed history $b_n = (z, x_{1:n})$. Using the equality (25) presented in Lemma 2 we have,

$$\begin{aligned} &P(X_{n+1} = x | B_n = b_n) \\ &= \sum_{\mathcal{S} \in E_n} P(X_{n+1} = x, \theta \in C_{\mathcal{S}} | B_n = b_n) \\ &= \sum_{\mathcal{S} \in E_n} P(X_{n+1} = x | \theta \in C_{\mathcal{S}}, B_n = b_n) P(\theta \in C_{\mathcal{S}} | B_n = b_n). \end{aligned} \tag{30}$$

Now, since for any $\mathcal{S} \in E_n$, $\{\theta \in C_{\mathcal{S}}, Z = z\} \subset \{B_n = b_n\}$ according to Lemma 2, we can simplify:

$$P(X_{n+1} = x | \theta \in C_{\mathcal{S}}, B_n = b_n) = P(X_{n+1} = x | \theta \in C_{\mathcal{S}}, Z = z). \tag{31}$$

Also, using Lemma 3, we obtain

$$P(\theta \in C_{\mathcal{S}} | B_n = b_n) = \frac{f_0(C_{s^{(1)}}) \ldots f_0(C_{s^{(k)}})}{\sum_{\mathcal{S} \in E_n} f_0(C_{s^{(1)}}) \ldots f_0(C_{s^{(k)}})}. \tag{32}$$

Finally, according to (1), X_{n+1} is the sum of k Bernoulli random variables $\mathbb{1}_{A_{n+1}}(\theta_1), \ldots, \mathbb{1}_{A_{n+1}}(\theta_k)$. Given the event $\{\theta \in C_{\mathcal{S}}, Z = z\}$, these k Bernouili r.v's are conditionally independent with respective parameters $q_1 = \frac{f_0(A_{n+1} \cap C_{s(1)})}{f_0(C_{s(1)})}, \ldots, q_k = \frac{f_0(A_{n+1} \cap C_{s(k)})}{f_0(C_{s(k)})}$. This conditional independence can be verified as follows. Consider any fixed binary vector $w \in \{0,1\}^k$. For each $i = 1, \ldots, k$, let D_i be equal to A_{n+1} if $w_i = 1$ and its complement A_{n+1}^c if $w_i = 0$. Then,

$$\begin{aligned} &P(\mathbb{1}_{A_{n+1}}(\theta_i) = w_i,\ i = 1, \ldots, k | \theta \in C_{\mathcal{S}}, Z = z) = P(\theta_i \in D_i,\ i = 1, \ldots, k | \theta \in C_{\mathcal{S}}, Z = z) \\ &= \frac{p_0(D_1 \cap C_{s^{(1)}} \times \cdots \times D_k \cap C_{s^{(k)}})}{p_0(C_{\mathcal{S}})} = \frac{\prod_{i=1}^k p_0(D_i \cap C_{s^{(i)}})}{\prod_{i=1}^k p_0(C_{s^{(i)}}))} = \prod_{i=1}^k \frac{p_0(D_i \cap C_{s^{(i)}})}{p_0(C_{s^{(i)}}))} \\ &= \prod_{i=1}^k P(\theta_i \in D_i | \theta \in C_{\mathcal{S}}, Z = z) = \prod_{i=1}^k P(\mathbb{1}_{A_{n+1}}(\theta_i) = w_i | \theta \in C_{\mathcal{S}}, Z = z). \end{aligned} \tag{33}$$

Using the fact that X_{n+1} is the sum of k conditionally independent Bernoulli random variables given $\theta \in C_{\mathcal{S}}$ and $Z = z$, we may provide an explicit probability mass function. When $q_1 = \cdots = q_k$, X_{n+1} is conditionally $\text{Bin}(k, q_1)$ given $\theta \in C_{\mathcal{S}}$ and $Z = z$. In general, let $W_1, \ldots, W_n$ be n independent discrete random variables with $W_i \sim \text{Bernoulli}(q_i)$, where $q_1, \ldots, q_n$ are any real numbers in [0,1]. The distribution of $Y = W_1 + \ldots + W_n$ is called *Poisson Binomial* distribution, which was first studied by S. D. Poisson in [27]. We denote the distribution of Y by $\text{PB}(q_1, \ldots, q_n)$ and its probability mass function $P(Y = y) = f_{\text{PB}}(y; q_1, \ldots, q_n)$ is given by

$$f_{\text{PB}}(y; q_1, \ldots, q_n) = \sum_{w=(w_1,\ldots,w_n) \in \{0,1\}^n; w_1 + \ldots + w_n = y} \prod_{j=1}^n q_j^{w_j} (1 - q_j)^{1 - w_j}, \tag{34}$$

and has mean and variance given by

$$\begin{aligned} E[Y] &= q_1 + \ldots + q_n, \\ Var[Y] &= q_1(1 - q_1) + \ldots + q_n(1 - q_n). \end{aligned} \tag{35}$$

Using this definition of the Poisson Binomial distribution, the conditional distribution of X_{n+1} given $\theta \in C_{\mathcal{S}}$ and $Z = z$ is $\text{PB}(q_1, \ldots, q_n)$.

Finally, putting together equations (30), (32), and the fact that X_{n+1} is conditionally $\text{PB}(q_1, \ldots, q_n)$ given $\theta \in C_{\mathcal{S}}$ and $Z = z$ provides the following characterization of the conditional probability mass function of X_{n+1} given $B_n = (Z, X_{1:n}) = b_n$.

Theorem 2. *For $n = 0$, given $\{B_0 = b_0\} = \{Z = z\}$, $X_1 \sim \text{Bin}(k, f_0(A_1))$. For $n = 1, 2, \ldots, N-1$, given $B_n = (Z, X_{1:n}) = b_n$, X_{n+1} is a mixture of Poisson Binomial distributions with probability mass function:*

$$\begin{aligned} &P(X_{n+1} = x | B_n = b_n) \\ = &\sum_{\mathcal{S} \in E_n} \frac{f_0(C_{s(1)}) \ldots f_0(C_{s(k)})}{\sum_{\mathcal{T} \in E_n} f_0(C_{t(1)}) \ldots f_0(C_{t(k)})} f_{\text{PB}}\left(x, q_1 = \frac{f_0(A_{n+1} \cap C_{s(1)})}{f_0(C_{s(1)})}, \ldots, q_k = \frac{f_0(A_{n+1} \cap C_{s(k)})}{f_0(C_{s(k)})}\right). \end{aligned} \tag{36}$$

5 The Dyadic Policy for Localizing Multiple Objects

We now present the first policy of interest: the *dyadic policy*. This policy is easy to implement, and is non-adaptive, allowing its use in parallel computing environments. The description of the dyadic

policy will be given in Section 5.1. In Section 5.2, we will prove the theorem concerning the value of this policy and derive the last equality in our main results (8). Finally, asymptotic normality of $H(p_N)$ under the dyadic policy will be provided in Section 5.3.

5.1 Description of the dyadic policy

The definition of the dyadic policy is given in (7). In this section, we provide an iterative construction of this policy, introducing notation which will be useful later on.

First, we partition the support of f_0 into two subsets, $A_{1,0}$ and $A_{1,1}$:

$$A_{1,0} = \left(Q\left(0\right), Q\left(\frac{1}{2}\right)\right] \cap \operatorname{supp}(f_0), \tag{37a}$$

$$A_{1,1} = \left(Q\left(\frac{1}{2}\right), Q\left(1\right)\right] \cap \operatorname{supp}(f_0), \tag{37b}$$

where Q, as defined in (6), denotes the quantile function. With this partition, the question asked at time 1 is

$$A_1 = A_{1,1}. \tag{38}$$

Then we adopt a similar procedure recursively for each $n = 1, \ldots, N-1$ to partition $A_{n,j}$ into two subsets, $A_{n+1,2j}$ and $A_{n+1,2j+1}$ and then construct the question from these partitions. For $j = 0, \ldots, 2^n - 1$, define

$$A_{n+1,2j} = \left(Q\left(\frac{2j}{2^{n+1}}\right), Q\left(\frac{2j+1}{2^{n+1}}\right)\right] \cap \operatorname{supp}(f_0), \tag{39a}$$

$$A_{n+1,2j+1} = \left(Q\left(\frac{2j+1}{2^{n+1}}\right), Q\left(\frac{2j+2}{2^{n+1}}\right)\right] \cap \operatorname{supp}(f_0), \tag{39b}$$

Then the question asked at time $n+1$ is

$$A_{n+1} = \bigcup_{j=0}^{2^n-1} A_{n+1,2j+1}. \tag{40}$$

An illustration of these sets A_n is provided below in Figure 3.

Note that the dyadic policy is non-adaptive, as only the prior distribution is used to construct the next set and not the answer to previous questions.

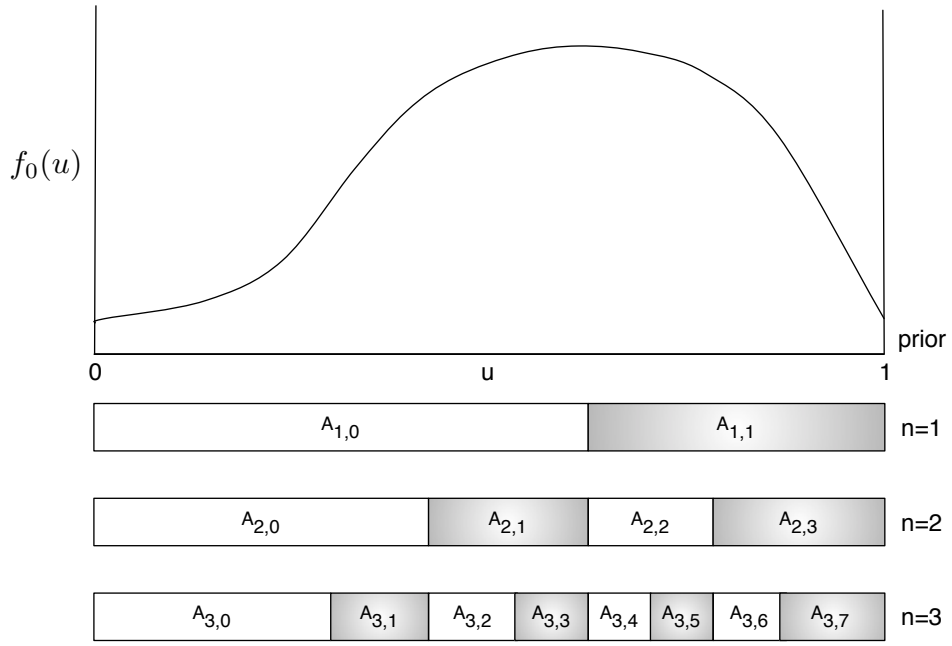


Figure 3: Illustration of the dyadic policy. The prior density with support $[0, 1]$ is displayed above the illustrations of the sets $A_{n,k}$ for $n = 1, 2, 3$. The question set A_n is the union of the dark subsets $A_{n,k}$ for that value of n.

5.2 The value of the dyadic policy

The value of the dyadic policy is stated as follows:

Theorem 3. *Under the dyadic policy π_D,*

$$R(\pi_D, N) = H_0 - H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right) N. \tag{41}$$

Proof. In this proof, we will first simplify the equation (36) in Theorem 2 to obtain the posterior distribution of X_{n+1} under the dyadic policy. Then we will calculate the entropy $H^{\pi_D}(X_{n+1}|B_n)$ and employ Lemma 1 to compute the value of the dyadic policy.

At time n, where $1 \leq n \leq N$, the support of f_0 is partitioned into pairwise disjoint subsets $\{A_{n,0}, \ldots, A_{n,2^n-1}\}$. Recall the definition of C_s in (22). The sets C_s provide a bijection which maps a binary sequence $s \in \{0, 1\}^n$ to a subset $A_{n,j(s)}$ for some $j(s) \in \{0, 1, \ldots, 2^n - 1\}$. Hence, $C_{s^{(i)}}$ in (36) can be rewritten as

$$C_{s^{(i)}} = A_{n,j(s^{(i)})}, \text{ for some index } j(s^{(i)}) \in \{0, 1, \ldots, 2^n - 1\}. \tag{42}$$

According to the construction of dyadic questions in Section 5.1, $A_{n+1} = \bigcup_{j=0}^{2^n-1} A_{n+1,2j+1}$. Moreover, $A_{n+1,2j(s^{(i)})+1} \subset A_{n,j(s^{(i)})}$ and $A_{n+1,2j+1} \cap A_{n,j(s^{(i)})} = \emptyset$, for all $j \neq j(s^{(i)})$. Thus, by (42) we have

$$A_{n+1} \cap C_{s^{(i)}} = A_{n+1,2j(s^{(i)})+1}. \tag{43}$$

Combining the above result with the fact that $f_0(A_{n+1,2j(s^{(i)})+1}) = \frac{1}{2} f_0(A_{n,j(s^{(i)})})$ yields

$$\frac{f_0(A_{n+1} \cap C_{s^{(i)}})}{f_0(C_{s^{(i)}})} = \frac{1}{2}, \tag{44}$$

and this is true for all $i = 1, 2, \ldots, k$.

Thus, for $n \geq 1$, we can simplify (36) in Theorem 2 as

$$\begin{aligned}
& p_n(X_{n+1} = x | B_n = b_n) \\
= & \sum_{\mathcal{S} \in E_n} \frac{f_0(C_{s(1)}) \ldots f_0(C_{s(k)})}{\sum_{\mathcal{T} \in E_n} f_0(C_{t(1)}) \ldots f_0(C_{t(k)})} f_{\mathrm{PB}}\left(x, q_1 = \frac{f_0(A_{n+1} \cap C_{s(1)})}{f_0(C_{s(1)})}, \ldots, q_k = \frac{f_0(A_{n+1} \cap C_{s(k)})}{f_0(C_{s(k)})}\right) \\
= & \sum_{\mathcal{S} \in E_n} \frac{f_0(C_{s(1)}) \ldots f_0(C_{s(k)})}{\sum_{\mathcal{T} \in E_{n+1}} f_0(C_{t(1)}) \ldots f_0(C_{t(k)})} f_{\mathrm{PB}}\left(x, q_1 = \tfrac{1}{2}, \ldots, q_k = \tfrac{1}{2}\right) \\
= & f_{\mathrm{PB}}\left(x, q_1 = \tfrac{1}{2}, \ldots, q_k = \tfrac{1}{2}\right) \frac{\sum_{\mathcal{S} \in E_n} f_0(C_{s(1)}) \ldots f_0(C_{s(k)})}{\sum_{\mathcal{T} \in E_n} f_0(C_{t(1)}) \ldots f_0(C_{t(k)})} \\
= & f_{\mathrm{PB}}\left(x, q_1 = \tfrac{1}{2}, \ldots, q_k = \tfrac{1}{2}\right).
\end{aligned} \tag{45}$$

The density above is just the density of the binomial distribution $\mathrm{Bin}\left(k, \frac{1}{2}\right)$. We proved that given $\{B_n = b_n\}$, X_{n+1} is distributed as $\mathrm{Bin}\left(k, \frac{1}{2}\right)$ and $H^{\pi_D}(X_{n+1} | B_n = b_n) = H\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right)$ for all $n = 1, \ldots, N-1$. Thus, taking the expectation over all possible realizations of B_n, we obtain

$$H^{\pi_D}(X_{n+1} | B_n) = H\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right). \tag{46}$$

Since $f_0(A_1) = \frac{1}{2}$ under the dyadic policy, according to Theorem 2, $X_1 | Z = z$ is distributed as $\mathrm{Bin}\left(k, \frac{1}{2}\right)$ for any fixed z and $H^{\pi_D}(X_1 | B_0) = H\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right)$ as well.

Therefore, according to (12) in Lemma 1,

$$R(\pi_D, N) = E^{\pi_D}[H(p_N)] = H_0 - \sum_{n=0}^{N-1} H^{\pi_D}(X_{n+1} | B_n) = H_0 - H\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right) N. \tag{47}$$

□

Note that this is the last equality in our main result (8).

5.3 Convergence in entropy under the dyadic policy

In real applications, however, we are concerned not only about the expected entropy $E^{\pi_D}[H(p_N)|p_0]$ but also about the actual entropy $H(p_N)$ that we obtain in a specific trial. It would be beneficial if the actual entropy did not deviate too much from its expected value. It turns out to be the case for the dyadic policy under the assumptions that the prior density f_0 is bounded from above. Lemma 4 provides a decomposition formula for the actual entropy $H(p_n)$ into a sum of two terms. The first term is a sum of i.i.d. random variables. The second term is a converging martingale as will be shown in Lemma 5. Finally, Theorem 4 provides almost sure convergence and asymptotic normality for $H(p_n)$ as a direct consequence of Lemma 4 and 5. Note that the dyadic policy is deterministic, i.e., it does not make use of the random seed Z. As a consequence, in this section, we use $X_{1:n}$ to denote the history up to time n without including Z.

Lemma 4. *Under the dyadic policy, for all $n = 1, 2, \ldots, N$,*

$$H(p_n) = -\sum_{j=1}^{n} Z_j + I_2(n), \tag{48}$$

where $I_2(n)$ is a random variable and $Z_j = k - \log \binom{k}{X_j}$ with X_j following i.i.d binomial distribution $\mathrm{Bin}(k, \frac{1}{2})$.

Proof. Let $X_{1:n} = x_{1:n}$ be fixed. According to Lemma 3,

$$p_n(u_{1:k}) = \frac{p_0(u_{1:k})}{p_0\left(\bigcup\limits_{\mathcal{S}\in E_n} C_{\mathcal{S}}\right)} = \frac{f_0(u_1)\ldots f_0(u_k)}{\sum\limits_{\mathcal{S}\in E_n} f_0(C_{s^{(1)}})\ldots f_0(C_{s^{(k)}})}, \tag{49}$$

where $(u_{1:k}) \in C := \bigcup\limits_{\mathcal{S}\in E_n} C_{\mathcal{S}}$.

Under the dyadic policy, the support of f_0 is partitioned into 2^n subsets with identical probability masses after the final step and each $C_{s^{(i)}}$ is one such subset, for $i = 1, 2, \ldots, k$. Thus, we have

$$f_0(C_{s^{(i)}}) = 2^{-n}, \text{ for } i = 1, 2, \ldots, k \text{ and } \mathcal{S} \in E_n. \tag{50}$$

Let $|E_n|$ be the cardinality of E_n. Note that under the dyadic policy, every binary sequence s of length N corresponds to a nonempty set C_s. Furthermore, in step j, there are $\binom{k}{x_j}$ ways to choose the j^{th} row in the matrix satisfying the definition in (24), for $j = 1, 2, \ldots, n$. Thus, by the product rule,

$$|E_n| = \prod_{j=1}^{n} \binom{k}{x_j}. \tag{51}$$

By (50) and (51),

$$p_0(C) = \sum_{\mathcal{S}\in E_n} f_0(C_{s^{(1)}})\ldots f_0(C_{s^{(k)}}) = 2^{-nk} \prod_{j=1}^{n} \binom{k}{x_j}. \tag{52}$$

Combining the result above and the definition of the differential entropy, we have

$$\begin{aligned} H(p_n) =& -\int_C p_n(u_{1:k}) \log(p_n(u_{1:k}))\, du_{1:k} \\ =& -\int_C \frac{p_0(u_{1:k})}{p_0(C)} \log\left(\frac{p_0(u_{1:k})}{p_0(C)}\right) du_{1:k} \\ =& \left[\frac{\log\left(p_0(C)\right)}{p_0(C)} \int_C p_0(u_{1:k})\, du_{1:k}\right] + \left[-\frac{1}{p_0(C)} \int_C p_0(u_{1:k}) \log\left(p_0(u_{1:k})\right)\, du_{1:k}\right] \\ =& I_1(n) + I_2(n), \end{aligned} \tag{53}$$

where $I_1(n)$ and $I_2(n)$ denote the first term and the second term in the last equation above. $I_1(n)$ can be easily computed as

$$I_1(n) = \frac{\log\left(p_0(C)\right)}{p_0(C)} \int_C p_0(u_{1:k})\, du_{1:k} = \log\left(p_0(C)\right) = -\left(nk - \sum_{j=1}^{n} \log \binom{k}{x_j}\right). \tag{54}$$

Now consider $X_{1:n}$ as random variables. By Theorem 2, we see that under the dyadic policy, $X_{1:n}$ is a sequence of i.i.d. random variables $\text{Bin}\left(k, \frac{1}{2}\right)$. Moreover, $I_2(n)$ is random through its dependence on the random support C. Therefore, combining (53) and (54), we prove the claim in Lemma 4 by setting $Z_j = k - \log \binom{k}{X_j}$. □

Define $I_2(0) = H(p_0) = H_0$ so that (48) is also satisfied for $n = 0$. Applying the result above, we can furthermore analyze the term $I_2(n)$ and derive the following lemma.

Lemma 5. *Assume there exists $M > 0$ such that $f_0(u) \leq M$ for all $u \in \mathbb{R}$. Then the random variable $I_2(n)$ in (48) converges to a random variable $I_2(\infty)$ almost surely as $n \to \infty$, where $I_2(\infty)$ is a random variable and $E[|I_2(\infty)|] < \infty$.*

Proof. We prove almost sure convergence using the martingale convergence theorem (see Theorem 35.5 in [28]). First, let us calculate the expected value of Z_j as follows.

$$E(Z_j) = \sum_{j=0}^{k} \left(k - \log \binom{k}{j}\right) \binom{k}{j} 2^{-k}. \tag{55}$$

Therefore, $E(Z_j) = H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right)$ since

$$H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right) = -\sum_{j=0}^{k} \binom{k}{j} 2^{-k} \log\left(\binom{k}{j} 2^{-k}\right) = \sum_{j=0}^{k} \left(k - \log \binom{k}{j}\right) \binom{k}{j} 2^{-k}. \tag{56}$$

Now, let us verify that $I_2(n)$ is a martingale. According to (48),

$$E[I_2(n+1)|X_{1:n}] = E\left[H(p_{n+1}) + \sum_{j=1}^{n+1} Z_j \middle| X_{1:n}\right] \tag{57a}$$

$$= H(p_n) - H(X_{n+1}\|X_{1:n}) + \sum_{j=1}^{n} Z_j + E[Z_{n+1}|X_{1:n}] \tag{57b}$$

$$= I_2(n) - H(X_{n+1}\|X_{1:n}) + E[Z_{n+1}|X_{1:n}] \tag{57c}$$

$$= I_2(n) - H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right) + E[Z_{n+1}] \tag{57d}$$

$$= I_2(n), \tag{57e}$$

where (57b) is true by (11) in Lemma 1 and the fact that $Z_{1:n}$ is $\sigma(X_{1:n})$-measurable. (57d) holds because we have proved under the dyadic policy, $X_{n+1}|X_{1:n} \sim \text{Bin}\left(k, \frac{1}{2}\right)$, which is independent of $X_{1:n}$, and Z_{n+1} is also independent of $X_{1:n}$. (57e) holds because we have proved $E[Z_{n+1}] = H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right)$.

Next, we want to show that $E[|I_2(n)|] < \infty$. Let us fix $X_{1:n} = x_{1:n}$ and expand $I_2(n)$ in as

$$\begin{aligned} I_2(n) &= -\frac{1}{p_0(C)} \sum_{\mathcal{S} \in E_n} \int_{C_{\mathcal{S}}} f_0(u_1) \dots f_0(u_k) \log\left(f_0(u_1) \dots f_0(u_k)\right) du_{1:k} \\ &= -\frac{1}{p_0(C)} \sum_{\mathcal{S} \in E_n} \sum_{i=1}^{k} \left(\int_{C_{s(i)}} f_0(u_i) \log(f_0(u_i))\, du_i \prod_{j \neq i}^{k} \int_{C_{s(j)}} f_0(u_k)\, du_j \right) \\ &= -\frac{1}{p_0(C)} \sum_{\mathcal{S} \in E_n} \sum_{i=1}^{k} 2^{-n(k-1)} \int_{C_{s(i)}} f_0(u_i) \log(f_0(u_i))\, du_i. \end{aligned} \tag{58}$$

Now consider the integral $\int_{C_{s(i)}} f_0(u_i) \log(f_0(u_i))\, du_i$. Since $f_0(u_i) \leq M$, we can obtain an upper bound for $\int_{C_{s(i)}} f_0(u_i) \log(f_0(u_i))\, du_i$ as

$$\int_{C_{s(i)}} f_0(u_i) \log(f_0(u_i))\, du_i \leq \log M \int_{C_{s(i)}} f_0(u_i)\, du_i = 2^{-n} \log M. \tag{59}$$

Substituting (52) and (59) into (58), we have

$$I_2(n) \geq -k \log M. \tag{60}$$

Furthermore, define $I_2^+(n) = \max(I_2(n), 0), I_2^-(n) = \max(-I_2(n), 0)$ and we have

$$E[|I_2(n)|] = E[I_2^+(n)] + E[I_2^-(n)] = E[I_2(n)] + 2E[I_2^-(n)] \leq H_0 + 2k \log M, \tag{61}$$

where the last equation follows from the fact that $E[I_2(n)] = I_2(0) = H_0$ since I_2 is a martingale and $I_2^-(n) \leq k \log M$ by (60). Therefore, using the martingale convergence theorem, $I_2(n)$ converges to a random variable $I_2(\infty)$ almost surely with $E[|I_2(\infty)|] \leq H_0 + 2k \log M$.

□

From the proof above we can see that if f_0 is uniform over $(0, 1]$, $f_0(u_i) = 1$ for all $u_i \in (0, 1]$ and thus the term I_2 is 0. Therefore, in this case, $H(p_n) = -\left(nk - \sum_{j=1}^{n} \log \binom{k}{X_j}\right)$.

Now the following theorem is a direct consequence of the preceding lemmas.

Theorem 4. *Assume there exists $M > 0$ such that $f_0(u) \leq M$ for all $u \in \mathbb{R}$. Then under the dyadic policy,*

$$\lim_{N \to \infty} \frac{H(p_N)}{N} = -H\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right) \text{ almost surely,} \tag{62}$$

and

$$\lim_{N \to \infty} \frac{H(p_N) + NH\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right)}{\sqrt{N}} \stackrel{d}{=} N(0, \sigma^2), \tag{63}$$

where σ^2 is the variance of the random variable $\log \binom{k}{X}$ with $X \sim \mathrm{Bin}\left(k, \frac{1}{2}\right)$.

Proof. According to Lemma 5, $\lim_{N\to\infty} \frac{I_2(N)}{N} = \lim_{N\to\infty} \frac{I_2(\infty)}{N} = 0$ almost surely. Hence, by (48) in Lemma 4,

$$\lim_{N\to\infty} \frac{H(p_N)}{N} = \lim_{N\to\infty} \frac{I_2(N)}{N} - \frac{1}{N}\sum_{j=1}^{N} Z_j = 0 - E[Z_1] = -H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right) \tag{64}$$

almost surely.

To prove (63), note that

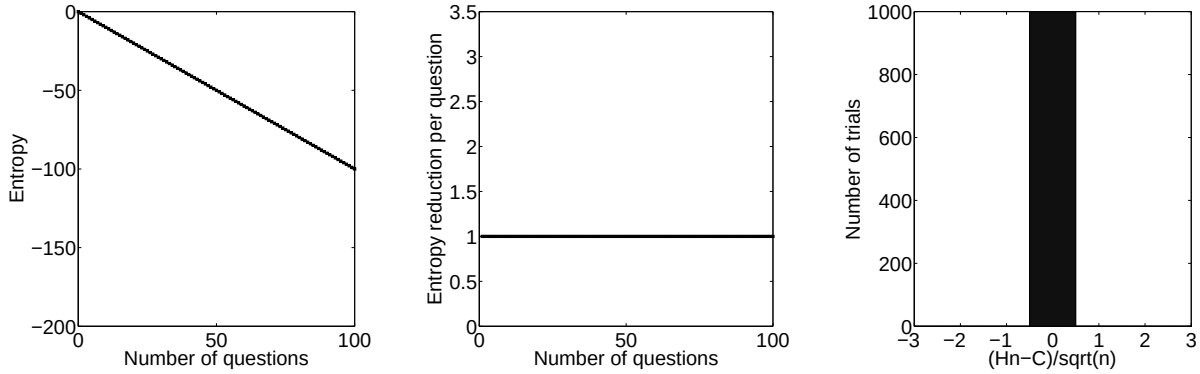
$$\frac{H(p_N) + NH\left(\text{Bin}\left(k, \frac{1}{2}\right)\right)}{\sqrt{N}} = \frac{I_2(N) - \sum_{i=1}^{N} Z_j + NH\left(\text{Bin}\left(k, \frac{1}{2}\right)\right)}{\sqrt{N}}. \tag{65}$$

Furthermore, since by Lemma 5 $I_2(N)$ converges to $I_2(\infty)$ almost surely and $E[|I_2(\infty)|] < \infty$, $\frac{I_2(N)}{\sqrt{N}} \to 0$ almost surely, which implies $\frac{I_2(N)}{\sqrt{N}} \xrightarrow{\mathcal{L}} 0$. On the other hand, $E(Z_j) = H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right)$ and $Var(Z_j) = Var\left(\log\binom{k}{X}\right) = \sigma^2$, where $X \sim \text{Bin}\left(k, \frac{1}{2}\right)$. Hence, by the central limit theorem, we have $\frac{-\sum_{i=1}^{N} Z_j + NH\left(\text{Bin}\left(k,\frac{1}{2}\right)\right)}{\sqrt{N}} \xrightarrow{\mathcal{L}} \mathcal{N}(0, \sigma^2)$. Therefore, by Slutsky's Theorem (Theorem 25.4 in [28]),

$$\frac{H(p_N) + NH\left(\text{Bin}\left(k, \frac{1}{2}\right)\right)}{\sqrt{N}} = \frac{I_2(N)}{\sqrt{N}} + \frac{-\sum_{i=1}^{N} Z_j + NH\left(\text{Bin}\left(k, \frac{1}{2}\right)\right)}{\sqrt{N}} \xrightarrow{\mathcal{L}} \mathcal{N}(0, \sigma^2). \tag{66}$$

□

Figure 4 below shows the simulation results for localizing one object, two objects, and three objects under the dyadic policy, respectively. We assume the prior density f_0 is uniform over $(0, 1]$ and ask 100 questions to locate the objects. The top line corresponds to locating a single object. In this case, the dyadic policy is actually optimal and identical to the greedy policy as was proved in [5]. Moreover, the entropy process $H(p_n)$ is in this case deterministic. The middle and bottom lines show the results for respectively $k = 2$ and $k = 3$ objects. In this case, the entropy process $H(p_n)$ is not deterministic anymore. The entropy reduction per question which is visualized in the second column is asymptotically equal to $H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right)$ according to the law of large numbers. The third column illustrates the asymptotic normality of the entropy process for the dyadic policy.



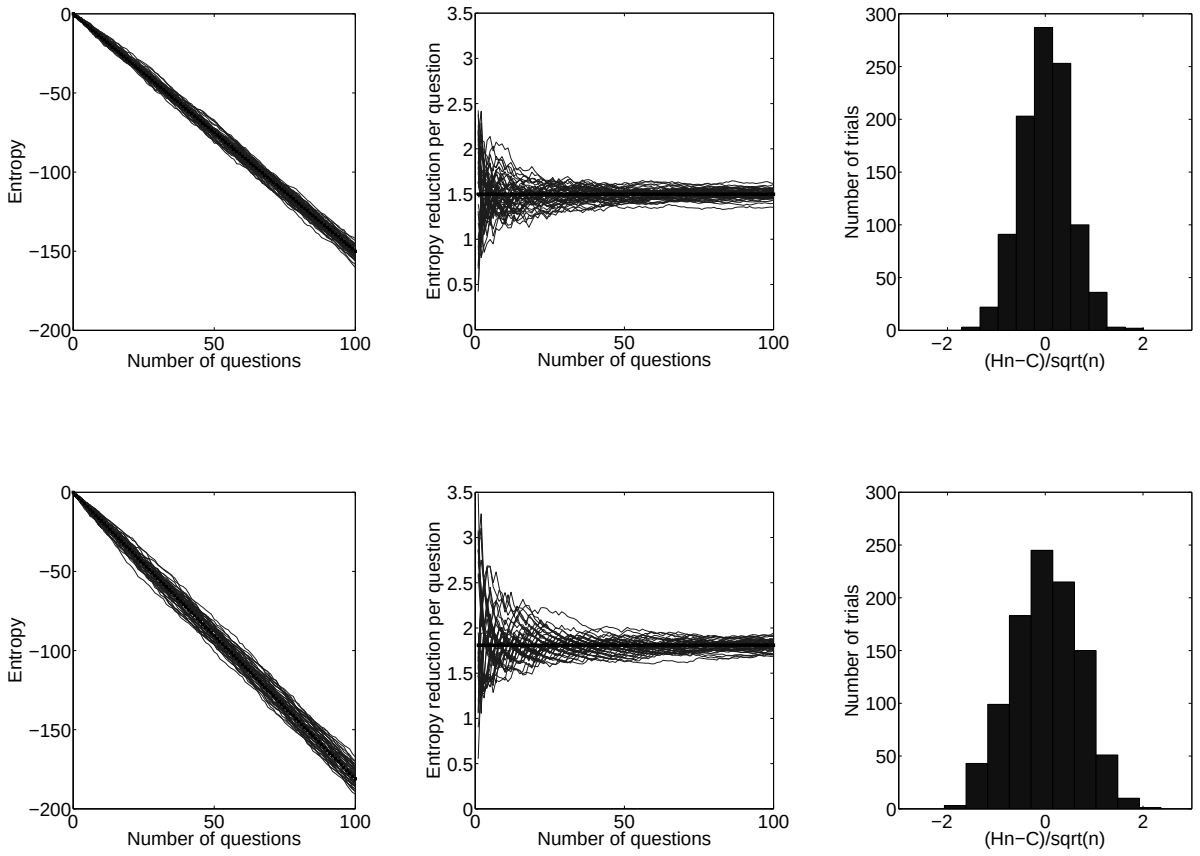


Figure 4: Simulation results for localizing one, two, three objects under the dyadic policy. $N = 100$ and f_0 is uniform over $(0, 1]$. The horizontal graphs above show the actual trajectories of entropy $H(p_n)$, average reduction in entropy $-\frac{H(p_n)}{n}$, and normality of $\frac{H(p_N)+NH\left(\mathrm{Bin}\left(k,\frac{1}{2}\right)\right)}{\sqrt{N}}$, respectively.

6 The Greedy Policy for Localizing Multiple Objects

In this section, we will present the second policy of interest–the *greedy policy*. The greedy policy is a family of policies (not unique) which pursue a maximal one-step expected reduction in entropy. Despite having a better performance than the dyadic policy, the greedy policy is difficult for us to parametrize and implement. A description of the greedy policy will be given in Section 6.1 and an upper bound of its value is shown in Section 6.2, which verifies our claim of the third inequality in the main results (8). Furthermore, we will provide an example in which the greedy policy outperforms the dyadic policy in Section 6.3 and thus this inequality cannot be reversed.

6.1 Description of the greedy policy

Unlike the dyadic policy, the greedy policy is adaptive, that is, the actual policy depends on the previous answers that we already observed, and at each step the question set $A_n \subset \mathbb{R}$ is defined in (5) to maximize the one-step expected reduction in entropy.

We prove that this argmin exists below in Theorem 5. The computation of the greedy policy might be complicated in some cases, however, the greedy policy is strictly better than the dyadic policy and we will demonstrate this point in Section 6.3.

6.2 The value of the greedy policy

Although deriving the value of the greedy policy seems impossible, we are able to employ Lemma 1 to derive an upper bound of it as the following.

Theorem 5. *The argmin* (5) *defining the class of greedy policies exists. Under any greedy policy* π_G*,*

$$R(\pi_G, N) \leq H_0 - H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right) N. \tag{67}$$

Proof. Fix some history $B_n = (Z, X_{1:n}) = b_n$. We first show existence of the argmin from (5), restated here as

$$\arg\min_A E[H(p_{n+1})|p_n, A_{n+1} = A], \tag{68}$$

where we recall that the minimum is taken over all Borel-measurable subsets of $\mathbb{R}$.

Since conditioning on the posterior distribution p_n under any fixed policy is equivalent to conditioning on $\{B_n = (Z, X_{1:n}) = b_n\}$, using (11) in Lemma 1, we have

$$\begin{aligned} E[H(p_{n+1})|p_n, A_{n+1} = A] &= E[H(p_{n+1})|B_n = b_n, A_{n+1} = A] \\ &= H(p_n|B_n = b_n, A_{n+1} = A) - H(X_{n+1}|B_n = b_n, A_{n+1} = A). \end{aligned} \tag{69}$$

Since the first term $H(p_n|B_n = b_n, A_{n+1} = A)$ does not depend on $\{A_{n+1} = A\}$, (68) can be rewritten as

$$\begin{aligned} &\arg\min_A H(p_n|B_n = b_n, A_{n+1} = A) - H(X_{n+1}|B_n = b_n, A_{n+1} = A) \\ &= \arg\max_A H(X_{n+1}|B_n = b_n, A_{n+1} = A). \end{aligned} \tag{70}$$

When $n = 0$, according to Theorem 2, we can rewrite the above argmax as

$$\arg\max_A H\left(\text{Bin}\left(k, f_0(A)\right)\right). \tag{71}$$

The maximum is achieved by any questions set A such that $f_0(A) = \frac{1}{2}$. For example, the first dyadic question $\left(Q\left(\frac{1}{2}\right), Q(1)\right] \cap \text{supp} f_0$ is one of such sets. This also proves $H^{\pi_G}(X_1|B_0) = H\left(\text{Bin}\left(k, \frac{1}{2}\right)\right)$.

When $n \geq 1$, using (36) in Theorem 2, we can rewrite the argmax in (70) as

$$\arg\max_A H\left(\sum_{\mathcal{S} \in E_n} \alpha(\mathcal{S}) \text{PB}\left(\frac{f_0(A \cap C_{s^{(1)}})}{f_0(C_{s^{(1)}})}, \ldots, \frac{f_0(A \cap C_{s^{(k)}})}{f_0(C_{s^{(k)}})}\right)\right), \tag{72}$$

where $\alpha(\mathcal{S}) = \frac{f_0(C_{s^{(1)}}) \cdots f_0(C_{s^{(k)}})}{\sum_{\mathcal{T} \in E_n} f_0(C_{t^{(1)}}) \cdots f_0(C_{t^{(k)}})}$ and $\sum_{\mathcal{S} \in E_n} \alpha(\mathcal{S}) = 1$.

Let $\mathbb{S} = \{s \in \{0,1\}^n : C_s \neq \emptyset\}$, and fix some arbitrary order of these elements so that $\mathbb{S}$ becomes a sequence rather than a set. For each $s \in \mathbb{S}$, let $r_s(A) = f_0(A \cap C_s)/f_0(C_s)$ so that (72) can be rewritten as

$$\arg\max_A H\left(\sum_{\mathcal{S} \in E_n} \alpha(\mathcal{S}) \text{PB}\left(r_{s^{(1)}}(A), \ldots, r_{s^{(k)}}(A)\right)\right). \tag{73}$$

For each Borel-measurable subset A of $\mathbb{R}$, $r(A) = (r_s(A) : s \in \mathbb{S})$ is an element of $[0,1]^{|\mathbb{S}|}$. Moreover, for each $r \in [0,1]^{|\mathbb{S}|}$, there is a Borel-measurable $A \subset \mathbb{R}$ such that $r(A) = r$. This is because the continuity of the prior cumulative density function allows us to construct the desired subset A as a union of sets, one for each element of $\mathbb{S}$. In this construction, the subset of A corresponding to $s \in \mathbb{S}$ is a subset of C_s containing a fraction r_s of the prior mass of C_s. This shows that the argmax (72) exists iff the following argmax exists:

$$\arg\max_{r \in [0,1]^{|\mathbb{S}|}} H\left(\sum_{\mathcal{S} \in E_n} \alpha(\mathcal{S}) \mathrm{PB}\left(r_{s^{(1)}}, \ldots, r_{s^{(k)}}\right)\right). \tag{74}$$

The function $r \mapsto H\left(\sum_{\mathcal{S} \in E_n} \alpha(\mathcal{S}) \mathrm{PB}\left(r_{s^{(1)}}, \ldots, r_{s^{(k)}}\right)\right)$ is continuous, and the set $[0,1]^{|\mathbb{S}|}$ is compact, so this argmax is attained. This shows that the argmax (5) defining the class of greedy policies is well-defined.

We now show an upper bound on the value of any greedy policy π_G by showing a lower bound on this quantity. The argument above also shows that under any greedy policy π_G, for $n \geq 1$,

$$\begin{aligned}
H^{\pi_G}(X_{n+1}|B_n = b_n) &= \max_A H(X_{n+1}|B_n = b_n, A_{n+1} = A) && \text{(75a)}\\
&= \max_{r \in [0,1]^{|\mathbb{S}|}} H\left(\sum_{\mathcal{S} \in E_n} \alpha(\mathcal{S}) \mathrm{PB}\left(r_{s^{(1)}}, \ldots, r_{s^{(k)}}\right)\right) && \text{(75b)}\\
&\geq \max_{r \in [0,1]^{|\mathbb{S}|}} \sum_{\mathcal{S} \in E_n} \alpha(\mathcal{S})\, H\left(\mathrm{PB}\left(r_{s^{(1)}}, \ldots, r_{s^{(k)}}\right)\right) && \text{(75c)}\\
&\geq \sum_{\mathcal{S} \in E_n} \alpha(\mathcal{S})\, H\left(\mathrm{PB}\left(\frac{1}{2}, \ldots, \frac{1}{2}\right)\right) && \text{(75d)}\\
&= H\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right). && \text{(75e)}
\end{aligned}$$

Above, we use the concavity of the entropy function to obtain the inequality (75c), and that $\mathrm{PB}(\frac{1}{2}, \ldots, \frac{1}{2})$ a special case of a Poisson Binomial Distribution to obtain (75d). The last line, (75e), follows from $\sum_{\mathcal{S} \in E_n} \alpha(\mathcal{S}) = 1$ and the fact that $\mathrm{PB}(\frac{1}{2}, \ldots, \frac{1}{2})$ is the $\mathrm{Bin}\left(k, \frac{1}{2}\right)$ distribution.

Furthermore, taking the expectation over all possible realizations of B_n, we obtain for $n \geq 1$,

$$H^{\pi_G}(X_{n+1}|B_n) \geq H\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right). \tag{76}$$

Recall that we already have $H^{\pi_G}(X_1|B_0) = H\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right)$ from previous arguments.

Finally, (12) in Lemma 1 shows

$$R(\pi_G, N) = E^{\pi_G}[H(p_N)] = H_0 - \sum_{j=0}^{N-1} H^{\pi_G}(X_{n+1}|B_n) \leq H_0 - H\left(\mathrm{Bin}\left(k, \frac{1}{2}\right)\right) N. \tag{77}$$

□

6.3 A setting in which the greedy policy is strictly better than the dyadic policy

In this section, we show that the greedy policy is *strictly* better than the dyadic policy under some circumstances.

Example 3: Suppose θ_1, θ_2 are two objects located in (0,1] with the prior f_0 being uniform over $(0, 1]$, and A_1 and A_2 the first two questions of the dyadic policy, $A_1 = \left(\frac{1}{2}, 1\right]$ and $A_2 = \left(\frac{1}{4}, \frac{1}{2}\right] \cup \left(\frac{3}{4}, 1\right]$. Now consider the following family of questions A_3 indexed by $0 \leq \alpha, \beta \leq 1$:

$$A_3 = \left(\frac{1-\alpha}{4}, \frac{1}{4}\right] \cup \left(\frac{2-\beta}{4}, \frac{1}{2}\right] \cup \left(\frac{3-\beta}{4}, \frac{3}{4}\right] \cup \left(\frac{4-\alpha}{4}, 1\right]. \tag{78}$$

According to (36), given $X_1 = 2$ and $X_2 = 0$, the point mass function of X_3 is

$$P(X_3 = x) = f_{\mathrm{PB}}(x; q_1 = \beta, q_2 = \beta), \tag{79}$$

which is a Binomial distribution with parameter β. The maximum entropy is then achieved when $\beta = 0.5$. Note that the dyadic question, corresponding to $\alpha = \beta = 0.5$, verifies this condition and as a consequence is also a valid question for the greedy policy.

Now, more interestingly, assume that $X_1 = X_2 = 1$, then, according to (36), the point mass function of X_3 is

$$\begin{aligned} p_2(X_3 = x) &= \frac{1}{4} f_{\mathrm{PB}}(x; q_1 = \alpha, q_2 = \alpha) + \frac{1}{4} f_{\mathrm{PB}}(x; q_1 = \beta, q_2 = \beta) \\ &+ \frac{1}{4} f_{\mathrm{PB}}(x; q_1 = \beta, q_2 = \beta) + \frac{1}{4} f_{\mathrm{PB}}(x; q_1 = \alpha, q_2 = \alpha), \end{aligned} \tag{80}$$

which simplifies to

x	$p_2(X_3 = x)$
0	$\frac{1}{2}(1-\alpha)^2 + \frac{1}{2}(1-\beta)^2$
1	$\alpha(1-\alpha) + \beta(1-\beta)$
2	$\frac{1}{2}\alpha^2 + \frac{1}{2}\beta^2$

Now, one can choose values for α and β such that $p_2(X_3 = x) = \frac{1}{3}$, $x = 0, 1, 2$. Specifically,

$$\alpha = \frac{1 + \frac{\sqrt{3}}{3}}{2} \text{ and } \beta = \frac{1 - \frac{\sqrt{3}}{3}}{2}. \tag{81}$$

In this case $H(p_2(X_3 = \cdot)) = \log(3) > 1.5$ which shows that the greedy policy is in this case strictly better than the dyadic policy.

7 Conclusion

We have considered the problem of twenty questions with noiseless answers, in which we aimed at locating multiple objects simultaneously. There are a variety of applications associated with this problem, such as group testing, computer vision, stochastic simulation and bioinformatics. By adopting the approach of minimizing the expected entropy of the posterior distribution, we derived a lower bound on the expected entropy and studied two classes of policies, the *dyadic policy* and the *greedy policy*. Although the greedy policy, as we have shown, outperforms the dyadic policy in reducing the expected entropy, the latter employs a series of pre-determined question sets and thus is easy to implement. In addition, the dyadic policy beats traditional policies such as the *sequential bifurcation policy* and is relatively stable in the sense that the average reduction in entropy converges under certain assumptions (Section 5.3).

Also, there are several questions calling for future works. First, in real applications, noisy answers provide a more natural and accurate approximation but we only considered noiseless answers in this paper. Second, we assumed the number of the objects is known, but in a more general setting, this assumption should be released. Third, another objective function such as the mean-squared error can replace the expected entropy, which measures the performance of a specific policy differently. We feel that researches in these and other questions will be prosperous and fruitful.

Acknowledgments

We would like to thank Li Chen for the fruitful discussions and the preliminary work which eventually led to this manuscript. Bruno Jedynak was partially supported by NSF IIS-0964416 and by the Science of Learning Institute at Johns Hopkins University through the research grant untitled "Spatial Localization Through Learning: An Information Theoretic Approach". Peter Frazier was supported by NSF CAREER CMMI-1254298, NSF IIS-1247696, AFOSR YIP FA9550-11-1-0083, and AFOSR FA9550-12-1-0200.

A Definition of the Sequential Bifurcation Policy

In this appendix, we define the sequential bifurcation policy used as a benchmark in Figure 1. This policy is based on the sequential bifurcation policy of [25], but adapted slightly to the setting considered in this paper.

We define the sequential bifuration (SB) policy as follows. At each point in time n, SB maintains a disjoint collection of intervals $\mathcal{D}_n = \{D_{n,1}, ..., D_{n,m_n}\}$. At time 0, $\mathcal{D}_0 = \{\mathbb{R}\}$, and for each n, SB obtains $\mathcal{D}_{n+1}$ and A_{n+1} recursively as follows. First, SB chooses the interval D_n^* in $\mathcal{D}_n$ with the largest mass under the prior, i.e.,

$$D_n^* \in \arg\max_{D \in \mathcal{D}_n} \int_D f_0(u)\, du. \tag{82}$$

Then, SB obtains A_{n+1} by splitting D_n^* at its conditional median under the posterior, and taking the left-hand portion. SB then creates $\mathcal{D}_{n+1}$ by adding to $\mathcal{D}_n \setminus D_n^*$ those intervals A_{n+1} and $D_n^* \setminus A_{n+1}$ shown by X_{n+1} to have at least one object.

This version of the sequential bifurcation policy differs slightly from the policy presented in [25] in that (1) it is designed for the continuum rather for a discrete domain; (2) it is designed for the case with known k, while running it for unknown k (as does [25]) would require an additional query of the number of objects in $\mathbb{R}$ at the start; (3) it is generalized for the case of a non-uniform prior distribution.

References

[1] S. Ulam, *Adventures of a Mathematician.* New York: Charles Scibner's Sons, 1976.

[2] A. Renyi, *A Diary on Information Theory.* Akademiai Kiado, 1984.

[3] C. Marini and F. Montagna, "Probabilistic Variants of Renyi-Ulam Game and Many-Valued Logic," *Task Quarterly*, vol. 9, no. 3, pp. 317–335, 2005.

[4] A. Pelc, "Searching with known error probabiity," *Theoretical Computer Science*, vol. 63, no. 185-202, 1989.

[5] B. Jedynak, P. I. Frazier, and R. Sznitman, "Twenty Questions with Noise: Bayes Optimal Policies for Entropy Loss," *Journal of Applied Probability*, no. 1, pp. 114–136, 2012.

[6] M. Horstein, "Sequential transmission using noiseless feedback," *IEEE Transactions on Information Theory*, vol. 9, no. 3, pp. 136–143, Jul 1963.

[7] R. Castro and R. Nowak, "Active sensing and learning," *Foundations and Applications of Sensor Management*, 2007.

[8] R. Waeber, P. I. Frazier, and S. G. Henderson, "Bisection search with noisy responses." *SIAM J. Control and Optimization*, vol. 51, no. 3, pp. 2261–2279, 2013.

[9] T. Tsiligkaridis, B. M. Sadler, and A. O. Hero III, "Collaborative 20 questions for target localization," *CoRR*, vol. abs/1306.1922, 2013.

[10] D. Du and F. Hwang, *Combinatorial Group Testing and its Applications*. World Scientific Pub Co., 2000.

[11] D. R. Stinson, T. V. Trung, and R. Wei, "Secure frameproof codes, key distributiong patterns, group testing algorithms and related structures," *Journal of Statistical Planning and Inference*, vol. 86, pp. 595–617, May 2000.

[12] D. Eppstein, M. T. Goodrich, and D. S. Hirschberg, "Improved combinatorial group testing algorithms for real-world problem sizes," *SIAM Journal on Computing*, vol. 36, pp. 1360–1375, January 2007.

[13] N. J. A. Harvey, M. Pătraşcu, Y. Wen, S. Yekhanin, and V. W. S. Chan, "Non-adaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs," *IEEE International Conference on Computer Communications*, pp. 697–705, May 2007.

[14] E. Porat and A. Rothschild, *Automata, Languages and Programming*. Berlin Heidelberg: Springer, 2008, vol. 5125, ch. Explicit Non-adaptive Combinatorial Group Testing Schemes, pp. 748–759.

[15] S. Kauffman, *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford University Press, 1996.

[16] J. S. Buzas and G. S. Warrington, "Optimized random chemistry," February 2013, arXiv:1302.2895 [math.PR].

[17] F. Chung, R. Graham, and T. Leighton, "Guessing secrets," *The Electronic Journal of Combinatorics*, vol. 8, p. 13, 2001.

[18] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes*, 2nd ed. Cambridge: MIT Press, 1972.

[19] D. L. desJardins, "Precise coding with noiseless feedback," Ph.D. dissertation, University of California at Berkeley, 2002.

[20] M. J. Eppstein and P. D. H. Hines, "A "random chemistry" algorithm for identifying collections of multiple contingencies that initiate cascading failure," *IEEE Transactions on Power Systems*, vol. 27, pp. 1698–1705, February 2012.

[21] R. Sznitman, A. Lucchi, P. Frazier, B. Jedynak, and P. Fua, "An optimal policy for target localization with application to electron microscopy," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

[22] R. Sznitman and B. Jedynak, "Active testing for face detection and localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1914–1920, July 2010.

[23] R. Sznitman, R. Richa, R. H. Taylor, B. Jedynak, and G. D. Hager, "Unified detection and tracking of instruments during retina microsurgery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1263–1273, 2013.

[24] P. Frazier, B. Jedynak, and L. Chen, "Sequential screening: A bayesian dynamic programming analysis," in *Proceedings of the 2012 Winter Simulation Conference*, 2012.

[25] B. W. M. Bettonvil and J. P. C. Kleijnen, "Searching for important factors in simulation models with many factors: sequential bifurcation," *European Journal of Operational Research*, vol. 96, no. 1, pp. 180–194, 1997.

[26] P. Frazier, *Wiley Encyclopedia of Operations Research and Management Science*. Wiley, 2010, ch. Learning with Dynamic Programming.

[27] S. D. Poisson, *Recherches sur la Probabilitè des jugements en matié criminelle et en matiére civile*. Paris: Bachelier, 1837.

[28] P. Billingsley, *Probability and Measure*, anniversary ed. Hoboken, NJ: Wiley, 2012.

Chapter 5

Active testing for face detection and localization

Active Testing for Face Detection and Localization

Raphael Sznitman and Bruno Jedynak

Abstract—We provide a novel search technique which uses a hierarchical model and a mutual information gain heuristic to efficiently prune the search space when localizing faces in images. We show exponential gains in computation over traditional sliding window approaches, while keeping similar performance levels.

Index Terms—Active testing, face detection, visual search, coarse-to-fine search, face localization.

1 Introduction

In recent years, face detection algorithms have provided extremely accurate methods to localize faces in images. Typically, these have involved the use of a strong classifier which estimates the presence of a face given a particular subwindow of the image. Successful classifiers have used Boosted Cascades (BCs) [1], [2], [3], [4], Neural Networks [5], [6], [7], and SVMs [8], [9], among others.

In order to localize faces, the aforementioned algorithms have relied on a sliding window approach. The idea is to inspect the entire image by sequentially observing each and every location a face may be in by using a classifier. In most face detection algorithms [1], [3], [4], [6], this involves inspecting all pixels of the image for faces, at all possible face sizes. This exhaustive search, however, is computationally expensive and in general not scalable to large images. For example, for real-time face detection using modern cameras ($4{,}000 \times 3{,}000$ pixels per image), more than *100 million* evaluations are required, making it hopeless on any standard computer.

To overcome this problem, previous works in object and face localization have simply reduced the pose space by allowing only a coarse grid of possible locations [1], [5], [10]. An elegant improvement to object detection was proposed in [2], where "feature-centric" evaluation are performed as opposed to "window-centric," allowing previous computation to be reused. Such a method, however, relies on strong knowledge of the classifier used. More recently, a globally optimal branch-and-bound subwindow search method for objects in images was proposed [11] and extended to videos [12]. Here, the classifier and the feature space used to locate the object are dependent on a single robust feature (e.g., SIFT [13]), making it difficult to use in the context of faces.

In this paper, we propose a novel search strategy which can be combined with any face classifier in order to significantly reduce the computational cost involved with searching the entire space. The design principle is as follows: We assume that a *perfect* face classifier is available, i.e., one which always provides the correct answer. In practice, however, such a classifier does not exist and an accurate one (as in [1], [3], [4], [6]) will be used instead. Our goal is then to reduce the total number of classifier evaluations required to detect and locate faces in images while still providing similar performance levels when compared with an exhaustive search.

A proposed strategy for computational shape recognition [14] argues that the task of visually recognizing an object can be accomplished by querying the image in a sequential and adaptive way. In general, this can be regarded as a coarse-to-fine approach to perception [1], [15], [16], [17]. This "twenty questions" approach can be described as follows: there is a fact to be verified, e.g., "is there a face in the field of view," and each query, which consists of evaluating a particular function of the image, is chosen to maximally reduce the expected uncertainty about this fact. In the context of computer vision, such approaches have led to two different types of search algorithms: offline and online. In the offline versions, the "where to look next" strategy is computed once and for all, anticipating all possible queries. It has led to efficient algorithms for symbol recognition [15], face [16], and cat [17] detection. In the online version, the strategy is computed sequentially as information is gathered. It has led to a road tracking algorithm [14], [18]: This approach is known as *Active Testing* (AT).

In this paper, we extend the active testing framework in order to do fast face detection and localization. We provide a way to ask questions that are general and specific with regard to the face pose and span different feature spaces. Similarly to the "twenty questions" game, questions such as "is the object at this location with this size?" are asked by means of an accurate face classifier [1], [4], [6], [9], independently of what features are used to guide the search. We show here that this approach provides a coherent framework, with few parameters to choose or tune, which significantly reduces the number of classifier evaluations necessary to localize faces. Comparison of our method with state-of-the-art face detection algorithms and the traditional sliding window approach indicates that our framework reduces, by several orders of magnitude, the number of classifier evaluation needed while maintaining similar accuracy levels on localization and detection tasks. Even though this paper specifically focuses on frontal faces, this approach can be extended to faces in general [19], [20], [21], [22], [23], other object categories [24], and to most classifiers in the machine learning literature.

The remainder of this paper is organized as follows: In Section 2, the general framework of our method is presented along with implementation details. Section 3 describes localization experiments, and in Section 4 we compare the performance with state-of-the-art methods on a detection and localization task. Concluding remarks are provided in Section 5.

2 Active Testing

The goal set forth is to detect and localize a single frontal face of unknown size, which may or may not be present in the image. We define the pose of a face as the pixel location of the face center and a face scale. That is, we treat localization as placing a bounding box around a face. In Section 4, we detail how this can be extended to searching for multiple faces.

AT can be regarded as a search algorithm which uses an information gain heuristic in order to find regions of the search space which appear promising. The region which is to be observed next is determined as information is gathered, and thus can be viewed as an *online* variation of the "twenty questions" game. The general approach is as follows: We are looking for a face in an image, and are provided with a set of questions which help us determine where the face is located. Questions are answered with some uncertainty, reducing the search space and eventually leading to the face pose.

In addition, it is also assumed that a special question regarding the exact face pose is available. This question is treated as an "Oracle," always providing a *perfect* answer when queried, but is computationally expensive relative to other questions. Querying

- *R. Sznitman is with the Department of Computer Science, The Johns Hopkins University, CSEB Room 136, 3400 North Charles Street, Baltimore, MD 21218. E-mail: sznitman@jhu.edu.*
- *B. Jedynak is with the Center for Imaging Science (JHU) and Department of Applied Mathematics and Statistics, The Johns Hopkins University, Whitehead 208B, 3400 North Charles Street, Baltimore, MD 21218-2686, and the Laboratoire de Mathématiques Paul Painlevé and Institut Universitaire de Technologie, Université des Sciences et Technologies de Lille, France. E-mail: bruno.jedynak@jhu.edu.*

Manuscript received 2 Dec. 2009; revised 2 Mar. 2010; accepted 22 Mar. 2010; published online 13 May 2010.
Recommended for acceptance by A. Martinez.
For information on obtaining reprints of this article, please send E-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-12-0792.
Digital Object Identifier no. 10.1109/TPAMI.2010.106.

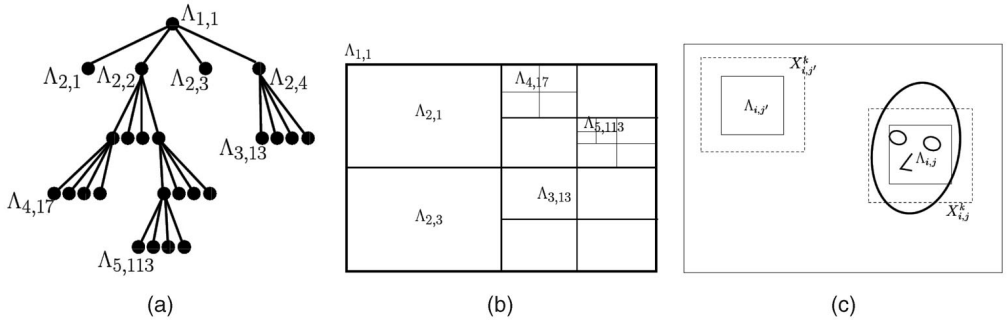


Fig. 1. (a) Each node in the tree corresponds to (b) a subwindow in the image. The root of the tree, $\Lambda_{1,1}$, represents the entire image space and has four children $(\Lambda_{2,1}, \Lambda_{2,2}, \Lambda_{2,3}, \Lambda_{2,4})$. (c) Example query: Here, the face center is, $Y = l \in \Lambda_{i,j}$. The query $X^k_{i,j}$ counts the proportion of edges in a window twice the size of $\Lambda_{i,j}$, centered on $\Lambda_{i,j}$. k indicates that we count the proportion of edges on a surface twice the size of the subwindow $\Lambda_{i,j}$, while $\{i, j\}$ provides the pose subset in Λ.

the oracle at every location would provide the face pose but is expensive and inefficient as certain questions are more informative than others and help reduce the search space faster. Consequently, a subgoal is to determine face pose with as few questions as possible.

2.1 Model and Algorithm

Let $Y = (L, S)$ be a discrete random variable defining the face pose, where L is the location of the face center (i.e., pixel coordinates) and S is the face scale such that S can take values $\{1, \ldots, M\}$ corresponding to M face size intervals. Additionally, Y can take one extra value when the face is not in the image. Let

$$\Lambda = \{\Lambda_{i,j}, i = 1, \ldots, D, j = 1, \ldots, 4^{i-1}\}$$

be a quadtree of finite size, which decomposes the image space; i indexes the level in the tree and j designates the cell at that level (see Fig. 1a). Every leaf is associated with a pixel in the image and each nonterminal node corresponds to a unique subwindow in the image, representing a subset of poses (Fig. 1b). When no face is present in the image, then $Y \in \bar{\Lambda}_{1,1}$, where $\bar{\Lambda}_{1,1}$ denotes the complement of $\Lambda_{1,1}$.

We are interested in refining the estimate of where the face is located iteratively and hence denote π_t as the probability density of Y at iteration step t. Let $u_{i,j,s} = P(L \in \Lambda_{i,j}, S = s)$, $\Lambda_{i,j} \subset \Lambda$, $s \in \{1, \ldots, M\}$. By construction, calculating $u_{i,j,s}$ can be achieved by summing the probability of $\Lambda_{i,j}$'s children. Clearly, $u_{1,1,s} = u_{2,1,s} + u_{2,2,s} + u_{2,3,s} + u_{2,4,s}$ and similarly for any other $u_{i,j,s}$. For any node, we also denote $u_{i,j} = \pi(\Lambda_{i,j}) = \sum_{s=1}^{M} u_{i,j,s}$. Let $\mathcal{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^K\}$ be a set of question families, such that, for each family k, $\mathbf{X^k} = \{X^k_{i,j}, i = 1, \ldots, D, j = 1, \ldots, 4^{i-1}\}$, where $X^k_{i,j}$ is a query from family k, about the pose subset $\Lambda_{i,j}$.

The generic AT algorithm (Algorithm 1) can then be seen as following: To begin, π_0 and the first query are initialized (lines 1 and 2). Three operations are then repeated: The response is observed (line 4); the belief of the location of Y is updated using the latest observation (line 5); a new query is chosen for the next iteration (line 6). The iteration is stopped when a terminating criteria is achieved (line 7). Each line is explained in detail in the following sections.

Algorithm 1. Active Testing (AT)

1: Initialize: $i \leftarrow 1, j \leftarrow 1, k \leftarrow 1, t \leftarrow 0$
2: Initialize: $\pi_0(\Lambda_{1,1}) = \pi_0(\bar{\Lambda}_{1,1}) = \frac{1}{2}$
3: **repeat**
4: Compute the test $x = X^k_{i,j}$
5: Compute π_{t+1} using π_t and x
6: Choose the next subwindow and test:

$$\{i, j, k\} = \arg\max_{i',j',k'} I\big(Y; X^{k'}_{i',j'}\big)$$

7: **until** $H(\pi_{t+1}) > 1 - \epsilon$ and/or $t < \gamma$.

2.2 Queries

The AT algorithm requires a set of query families, $\mathcal{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^K\}$, to be specified. Each query family, $\mathbf{X}^k$, consists of evaluating a specific type of image functional indexed by k. Members of a family $\mathbf{X^k} = \{X^k_{i,j}, i = 1, \ldots, D, j = 1, \ldots, 4^{i-1}\}$ are indexed by a pose index in Λ (as in [17]). That is, $X^k_{i,j}$ is an image functional, where k defines a particular computation and $\{i, j\}$ specifies the pose subset. Note that these queries are generic and need not be binary. Example queries can be seen in Fig. 1c.

In addition, *perfect* tests—which precisely predict the presence of a face by using a classifier—are included in $\mathcal{X}$. When this test is used at a specific pose, either the classifier responds positively and the face is deemed found, or conversely, the response is negative and the face is assumed not to be at this pose. That is, we assume no uncertainty with regard to the response of this classifier.

In order to specify the joint distribution between the face pose Y and queries $\mathcal{X}$, we make the following heuristic assumptions:

Conditional independence:

$$P\big(\{X^k_{i,j} = x\}, i = 1 \ldots D, j = 1 \ldots 4^{i-1}, k = 1 \ldots K | Y = (l, s)\big) = \prod_{i,j,k} P\big(X^k_{i,j} = x | Y = (l, s)\big). \quad (1)$$

Homogeneity:

$$P\big(X^k_{i,j} = x | Y = (l, s)\big) = \begin{cases} f^k_s(x; i), & \text{if } l \in \Lambda_{i,j}, \\ f^k_0(x; i), & \text{otherwise.} \end{cases} \quad (2)$$

Here, f^k_s characterizes the "response" to the query $X^k_{i,j}$ when the center of the face is within $\Lambda_{i,j}$ with size s. Similarly, f^k_0 is the "response" when the center is not in $\Lambda_{i,j}$. Additionally, even though KN queries are specified, where N is the number of nodes in Λ, the number of densities needed is only KD. That is, for each test family, only one density per level of Λ needs to be specified. This is why $f^k_s(\cdot, i)$ is only indexed by i.

Note that these assumptions are a simple way to make the problem tractable: For example, the conditional independence of queries given the location of the object Y assumption is clearly a simplification as the same pixel values are used to compute many queries at different levels of Λ. Similarly, the actual responses to tests might in fact depend on the precise location of the face within $\Lambda_{i,j}$. The homogeneity assumption simplifies the response model by assuming a single model for all cases. Even when using these assumptions, however, the experiments conducted here (Sections 3 and 4) indicate that these simplifications provide a good way to solve the problem at hand. In addition, this model should be taken into account when choosing queries to use: Similarly to a Naive Bayes model, queries should be individually informative.

2.3 Belief Update

Once an observation has been made, the new distribution of the face location Y must be calculated (line 5 of AT). At initialization

(line 1 of AT), $\pi_0(\Lambda_{1,1}) = \pi_0(\bar{\Lambda}_{1,1}) = \frac{1}{2}$, indicating that a face is believed to be in the image with probability $1/2$. Note that the probability $\pi_0(\Lambda_{1,1})$ is uniformly distributed within $\Lambda_{1,1}$ by construction. Given π_t and the query response $X^k_{i,j} = x$ at time step t, the updated distribution π_{t+1} can then be calculated by using Bayes formula:

$$\pi_{t+1}(l,s) = \frac{P\big(X^k_{i,j} = x | Y = (l,s)\big)\pi_t(l,s)}{\sum_{s'} \int_{l'} P\big(X^k_{i,j} = x | Y = (l',s')\big)\pi_t(l',s')dl'}. \quad (3)$$

Using Assumptions 1 and 2, then

$$P\big(X^k_{i,j} = x | Y = (l,s)\big) = f^k_0(x,i)\mathbb{1}_{\bar{\Lambda}_{i,j}}(l) + f^k_s(x,i)\mathbb{1}_{\Lambda_{i,j}}(l). \quad (4)$$

Let us now define the likelihood ratio as

$$r(x,s) = \frac{f^k_s(x,i)}{f^k_0(x,i)},\ s = 1 \dots M, \quad (5)$$

then (3) can be written as,

$$\pi_{t+1}(l,s) = \frac{1}{\mathcal{Z}(x)}\Big(\mathbb{1}_{\bar{\Lambda}_{i,j}}(l) + \mathbb{1}_{\Lambda_{i,j}}(l)r(x,s)\Big)\pi_t(l,s), \quad (6)$$

where $\mathcal{Z}(x)$ is the normalizing constant. Note that the evolution from π_t to π_{t+1} only relies on $r(x)$ and allows for probability mass to be shifted onto or away from $\Lambda_{i,j}$, depending on the response of $X^k_{i,j}$.

In order to reduce the number of nodes to update, only a subtree is maintained, where only nodes which have probability greater than some threshold τ are included. By construction of Λ, parent nodes have probability equal to the sum of their children, hence any node which has probability larger than τ also has parent with probability greater than τ. This guarantees that applying this threshold forms a subtree within Λ containing $\Lambda_{1,1}$. This approximation of π_t allows for a compact representation of the distribution.

2.4 Query Selection

We choose to select the next query by maximizing the mutual information gain between Y and the possible queries $X^k_{i,j}$ (line 6 of AT). This can be written as

$$I\big(Y; X^k_{i,j}\big) = H\big(X^k_{i,j}\big) - H\big(X^k_{i,j} | Y\big), \quad (7)$$

where

$$H\big(X^k_{i,j}\big) = h\left(\sum_{s=0}^{M} u_{i,j,s} f^k_s(\cdot)\right). \quad (8)$$

Here, $h(f)$ is the differential Shannon entropy of the density f. We simplify this expression by substituting $h(f)$ with the Gini Index [25]. The mutual information then becomes

$$I\big(Y; X^k_{i,j}\big) = \sum_{s=0}^{M}\sum_{m>s}^{M} u_{i,j,s}u_{i,j,m}\int \big(f^k_s - f^k_m\big)^2, \quad (9)$$

where $u_{i,j,0} = 1 - u_{i,j}$. Note that the term $\int (f^k_s - f^k_m)^2$ is the euclidean distance between the densities f^k_s and f^k_m, and only needs to be computed once and then stored for fast evaluation.

Since we are interested in choosing both the region $\Lambda_{i,j} \in \Lambda$ and a query family k which maximizes the information gain, one can simply evaluate $I(Y; X^k_{i,j})$ for all possible values of the triple (i,j,k) and select the parameters providing the largest gain. However, as described in Section 2.3, only a small subset of poses is ever considered at any iteration. For example, nodes which have little probability will surely only provide a small information gain. Consequently, we only need to evaluate (9) for the explicitly maintained subtree (Fig. 1a). Additionally, once a query has been chosen, it is removed from the set of possible queries, further reducing the amount of computation.

2.5 Terminating Criteria

At line 7 of the AT algorithm, two terminating criteria are presented: 1) The algorithm runs until the entropy of π, $H(\pi)$, is very high, and 2) the algorithm iterates for a fixed number of steps, γ. In the first case, running until the entropy is high corresponds to two possible outcomes: Either a face has been found and most of the probability mass is at a single leaf of Λ or most of the mass is outside the image $\bar{\Lambda}_{1,1}$ and no face is believed to be present in the image. In general, the choice of which criteria to use (1), 2), or both) is for the user to decide. Sections 3 and 4 show the behavior of these scenarios.

In addition, for all cases, the total number of queries is bounded by the size of the tree and the number of query families. As the algorithm iterates and the classifier is queried, the number of poses with strictly positive probability decreases. This provides a guarantee that, in the worst case, the face will be found after having observed all the poses.

2.6 Implementation

We now provide some implementation details and give a more in-depth algorithm for updating π (see Algorithm 2) and choosing queries.

Before the AT algorithm begins, all features necessary to evaluate queries from $\mathcal{X}$ for a given image are computed and stored in the form of an integral image making the evaluation of a query $O(1)$ operations (similarly to [11]). This is particularly efficient since queries $X^k_{i,j}$ compute nested subwindows.

In order to form and maintain the subtree of Λ (line 7), only nodes which are above a threshold ($\tau = 0.001$) are explicitly stored. To do this, we construct Λ as a quadtree, and maintain a frontier set $\mathcal{F}$. $\mathcal{F}$ consists of any node $\Lambda_{i,j}$ with $u_{i,j} > \tau$ and with all children having $u_{i+1,j'} < \tau$. Applying this rule at each iteration ensures that the maintained subtree is relevant to where the face is believed to be located. Additionally, since the probability associated at any node in the tree is equal to the sum of its children, we only need to update nodes in $\mathcal{F}$ and recurse through the tree to update the remaining nodes in Λ.

After having computed the query $X^k_{i,j}$, updating any node $\Lambda_{i',j'} \in \mathcal{F}$ is simple: If $\Lambda_{i',j'} \in \Lambda_{i,j}$, then $u_{i',j'} = r(y)u_{i',j'}/\mathcal{Z}$; otherwise, $u_{i',j'} = u_{i',j'}/\mathcal{Z}$. Doing so updates π as described in (6) in an efficient way. In addition, at any point in the updating of π, the next best query, S, seen so far is maintained. The denominator $\mathcal{Z}$ is calculated once and for all, and used to calculate (9) when each node is visited. Only the best score is kept and ultimately chosen for the following iteration of the AT algorithm. That is, we compute (6) and (9) one after the other, requiring only one pass through the subtree per iteration.

Algorithm 2. Update($\Lambda_{i',j'}, \Lambda_{i,j}, x, S, \mathcal{F}$)

1: **if** $\Lambda_{i',j'} \in \mathcal{F}$ **then**
2: **if** $\Lambda_{i',j'} \subset \Lambda_{i,j}$ **then**
3: $u_{i',j'} \leftarrow r(x)u_{i',j'}/\mathcal{Z}$
4: **else**
5: $u_{i',j'} \leftarrow u_{i',j'}/\mathcal{Z}$
6: **end if**
7: Maintain $\mathcal{F}$
8: **else**
9: **for** Each child, $\Lambda_{i'+1,j''}$, of $\Lambda_{i',j'}$ **do**
10: Update($\Lambda_{i'+1,j''}, \Lambda_{i,j}, x, S, \mathcal{F}$)
11: **end for**
12: $u_{i',j'} \leftarrow \sum_{j''} u_{i'+1,j''}$
13: **end if**
14: $S = \max(S, \max_k I(Y; X^k_{i',j'}))$

3 Face Localization

To demonstrate that this framework can be used to significantly reduce the number of classifier evaluations required when searching for a face in an image, we begin by evaluating the AT algorithm on a pure localization task (as done in [11]). In the following set of experiments, each image contains exactly one face. We describe in Section 3.1 the queries used to localize faces. In Section 3.3, we show how AT performs in terms of time, number of classifier evaluations, and accuracy.

We perform the following experiments on the Caltech Frontal Face data set [26], which consists of 450 images (896×592 pixels), each containing exactly one of 27 different faces in variously cluttered environments and illuminations. Face sizes range from approximately 100 to 300 pixels in width. We choose $M = 4$ possible face size intervals ($[100, 150]$, $[150, 200]$, $[200, 250]$, $[250, 300]$). All experiments are conducted on a 2.0 Gigahertz machine.

3.1 Face Queries

To locate faces, we first specify the following set of test families, $\mathcal{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^K\}$, and their associated distributions (f_s^k, f_0^k). In the following experiments, $K = 30$.

The first family of tests, $\mathbf{X}^1$, calculates the proportion of edge pixels (defined and computed as in [15] by means of an edge oriented integral image) in a window associated with the pose $\Lambda_{i,j}$. That is, $X_{1,1}^1$ is the proportion of pixels which are edges within $\Lambda_{1,1}$ and similarly for all $\Lambda_{i,j}$. Test families $\mathbf{X}^2$-$\mathbf{X}^5$ are similar to $\mathbf{X}^1$ in that they compute the proportion of edge pixels in a window centered on $\Lambda_{i,j}$, but of larger size, by a factor $F = \{2, 3, 4, 5\}$ (see Fig. 1c). Note that this factor is different from the scale S. Using these pose-indexed tests provides a way to test arbitrarily large regions, even when $\Lambda_{i,j}$ is a small subwindow. These tests also allow for overlapping $\Lambda_{i,j}$ regions and more precise estimation of the face scale.

Families $\mathbf{X}^6$-$\mathbf{X}^9$ are similar to $\mathbf{X}^1$ but compute the proportion of edge pixels in a particular direction (four possible directions). Similarly to families $\mathbf{X}^2$-$\mathbf{X}^5$, families $\mathbf{X}^{10}$-$\mathbf{X}^{25}$ allow for a scale factor for tests in a particular direction (four directions $\times$ four factors). Using integral images allows for computation of these tests with only four additions, making them very efficient.

We choose to model all the f_s^k for $s \in \{0, ..., M\}$ using Beta distributions. The Beta family permits to model a wide range of smooth distributions over the interval $[0, 1]$ with only two parameters. The parameters of each distribution are determined offline from a small training data set where the face location and scale is known (more details are given in Section 3.2).

Finally, families $\{\mathbf{X}^{26}, \ldots, \mathbf{X}^{30}\}$ are the *perfect* tests and involve testing for a face using a BC. Each family specifies testing for a face at all scales within a given interval ($s \in \{1, ..., M\}$). For each interval, we test for face sizes in increments of 10 percent of the smallest face size (total of 13 face sizes in the range $[100, 300]$). In terms of operations, evaluating this test requires, on average, 56 additions, one multiplication, and one comparison, per face size, making it significantly more costly than other queries. Since the BC is only informative when the pose is very specific, we restrict this test to leaves in Λ. These BCs are trained and provided by OpenCv [27], but modified to restrict testing to specific regions and face sizes. Even though better classifiers have recently been developed, we choose this one as it is publicly available and widely used.

3.2 Offline Training

We choose to model each $f_s^k(\cdot, i)$ with a Beta distribution with parameters (α, β). To do this, we randomly selected 50 images, from the Caltech Frontal Face Data set [26]. Note that far fewer images are used for training here when compared to other search methods (see [11], [12]) which typically use on the order of 10^3 images to train their systems. The estimation of the $f_s^k(\cdot, i)$ parameters is broken into two parts.

We first estimate all the background densities. That is, for each k and i, we randomly select $100j$s per image such that the face center is not in $\Lambda_{i,j}$. We then compute the tests $X_{i,j}^k = x$ and use these to compute the parameters using maximum likelihood estimation with 5,000 datapoints.

To estimate the foreground densities, a similar procedure is used. We describe the case $s = 1$. For each k and i, we randomly select $100j$s in each image such that the face center is in $\Lambda_{i,j}$. The parameters of $f_1^k(\cdot, i)$ are then estimated from the tests $X_{i,j}^k = x$. As before, 5,000 datapoints are used to estimate (α, β). In order to estimate $f_s^k(\cdot, i)$ for $1 < s \leq 4$, we subsample the images and repeat the same procedure (similar to [16]). Additionally, the $\int (f_s^k - f_m^k)^2$ term from (9) is then calculated by using a Monte Carlo approximation, and stored in a lookup table.

3.3 Single Face Localization

We set up the AT algorithm with BCs $(\mathrm{AT} + \mathrm{BC})$ to run until a face is found or until 5×10^5 classifier evaluations have been performed (see Fig. 4 for details on how this was chosen). We compare this with a sliding window approach using the identical BCs $(\mathrm{SW} + \mathrm{BC})$ and letting it run until a face is found or until all poses have been observed. Note that both $(\mathrm{AT} + \mathrm{BC})$ and $(\mathrm{SW} + \mathrm{BC})$ have the same pose space: all pixels and face sizes (e.g., pose space size $= 896 \times 592 \times 13 = 6{,}895{,}616$). In order to avoid any unfair bias as to where faces may be located, we randomly pick initial starting locations in the image for $(\mathrm{SW} + \mathrm{BC})$, looping around the image in order to observe all the poses. We report that $(\mathrm{AT} + \mathrm{BC})$ allows for exponential computational gains over the sliding window approach while keeping similar performance levels.

Fig. 2 shows a typical behavior of the AT algorithm on a given image. In general, the order in which queries are posed is complex and, in some cases, counterintuitive—validating the need for an *online* search strategy.

In Fig. 3a, we compare the accuracy of $(\mathrm{AT} + \mathrm{BC})$ and $(\mathrm{SW} + \mathrm{BC})$ on the remaining unused 400 images of the data set using an ROC curve. We observe that generally $(\mathrm{AT} + \mathrm{BC})$ does not suffer much from a loss in performance compared to the brute force sliding window approach. Note that the difference between the two methods is not significant.

To compare how much time $(\mathrm{AT} + \mathrm{BC})$ and $(\mathrm{SW} + \mathrm{BC})$ take to locate a face depending on the size of the pose space, we randomly selected a subset of 50 images from the testing set, subsampled these to have images of sizes 112×74, 224×148, 448×296, 672×444, and 896×592. Fig. 3b shows the average time of both methods for each image size. Note that the overhead of $(\mathrm{AT} + \mathrm{BC})$—the time to evaluate all queries tested, the update mechanism, and the query selection—is included in this plot (the additional time to compute an integral image for oriented edges is not included as it is negligible). As expected, we see that $(\mathrm{SW} + \mathrm{BC})$ is linear in the number of poses. However, the total time $(\mathrm{AT} + \mathrm{BC})$ takes to complete is significantly lower than $(\mathrm{SW} + \mathrm{BC})$ and even more so at large image sizes. In fact, $(\mathrm{AT} + \mathrm{BC})$ remains almost logarithmic even as the number of poses increases. This suggests that AT uses a form of "Divide and Conquer" search strategy. Note, that at image sizes smaller than (112×74), $(\mathrm{AT} + \mathrm{BC})$ is slower than $(\mathrm{SW} + \mathrm{BC})$ due to the overhead.

Fig. 3c shows the average number of classifier evaluations both $(\mathrm{AT} + \mathrm{BC})$ and $(\mathrm{SW} + \mathrm{BC})$ perform when changing the image size. Notice that the difference between $(\mathrm{AT} + \mathrm{BC})$ and $(\mathrm{SW} + \mathrm{BC})$ is even larger than the difference reported in Fig. 3b and that the AT algorithm significantly reduces the number of classifier evaluations. For the largest image size, AT requires 100 times fewer evaluations than SW.

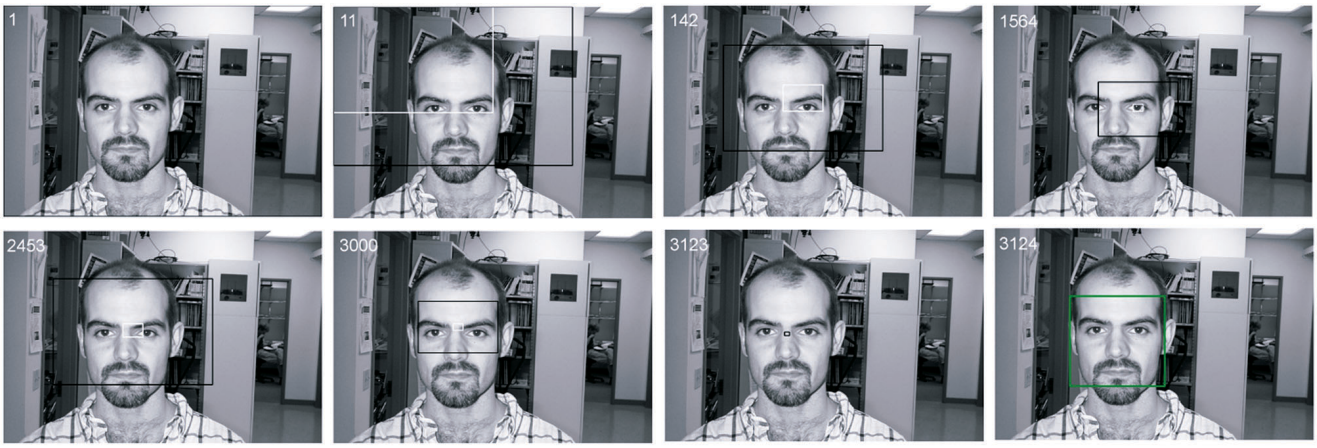


Fig. 2. Sequence of queries posed by the Active Testing algorithm on a test image from the Caltech Frontal Face Data set. In each image, a test $X_{i,j}^k$ is computed: White boxes show the pose, $\Lambda_{i,j}$, queried while black boxes show the subimage queried. The number indicated in the top left of each image is the iteration number of the AT algorithm. In image 3123, the Boosted Cascade is evaluated and a face is found at a given scale (green box).

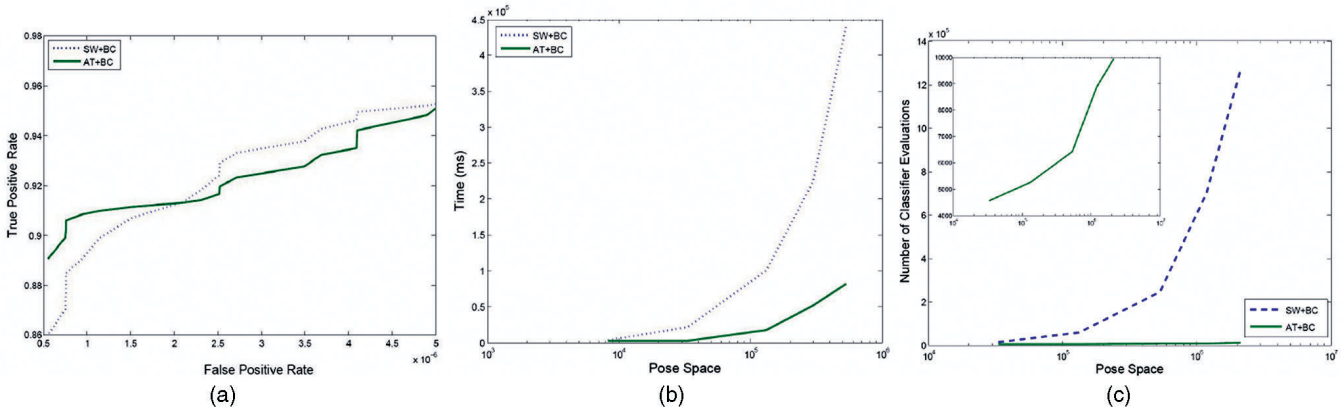


Fig. 3. (a) ROC curve of both $SW + BC$ and $AT + BC$ to find a face in the Caltech Frontal Face data set. The performance of both methods is approximately identical. (b) Average computation time with varying pose space size. Note that image size is in logarithmic scale. The AT algorithm performs in almost logarithmic time compared to SW. (c) Average number of classifier evaluations when the pose space increases. Additionally, a zoom of the AT performance is provided.

In Fig. 4a, we show how the accuracy of $(\mathrm{AT} + \mathrm{BC})$ is affected by the total number of classifier evaluations allowed. The dotted line indicates the performance of $(\mathrm{SW} + \mathrm{BC})$ when the entire pose space is observed. We see that after observing the entire pose space ($O(10^6)$ evaluations), 98 percent accuracy is achieved. Performance results are shown when $(\mathrm{AT} + \mathrm{BC})$ is stopped when either a face has been located or after (10^3, 10^4, 10^5, and 10^6) classifier evaluations have to be performed. After only 10^4 classifier evaluations are nearly 90 percent of detectable faces found. By 10^5 evaluations, AT performs at the same accuracy level as SW. In general, we can see in Fig. 4b that the number of evaluations required is approximatively Geometric $(p = 10^{-4})$. Hence, on average, 0.0014 of the total pose space is evaluated by the classifier.

As in [15], Fig. 4c shows a randomly selected test image, and the corresponding computational image associated (right). The computational image is a gray-scale image, which indicates the number of times each pixel has been included in a queried window (all types of queries included). Darker regions show areas where little computation has taken place, while white regions show important computation. As expected, we can see that regions of the image which contain few features (left part of the image) are not considered for much computation.

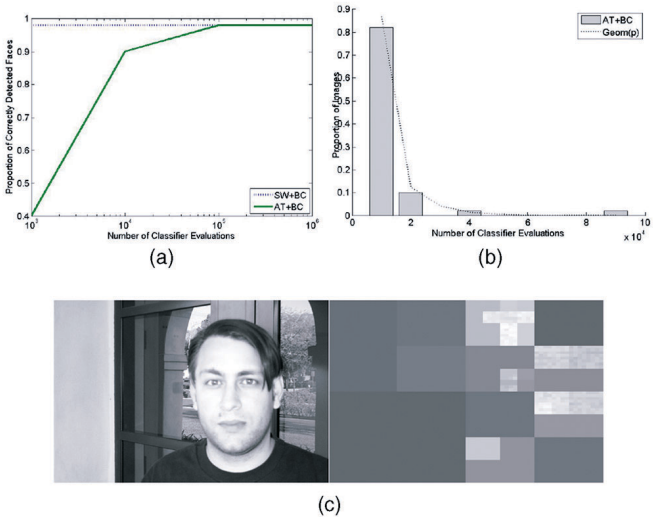


Fig. 4. (a) The proportion of faces detected increases with the number of classifier evaluations: 90 percent of faces are correctly detected with only 10^4 evaluations and with 10^5 classifier evaluations, the AT algorithm performs as well as SW, but much faster. (b) Histogram of the number of classifier evaluations. The dotted black line represents the point mass function of the Geometric distribution with parameter $p = 1/9,248$. (c) Face image and associated computation image. This gray-scaled image indicates the number of times each pixel has been included in a queried window.

4 Face Detection and Localization

We now test the AT algorithm in a much harder setting—a detection and localization task. We do this by looking for faces in

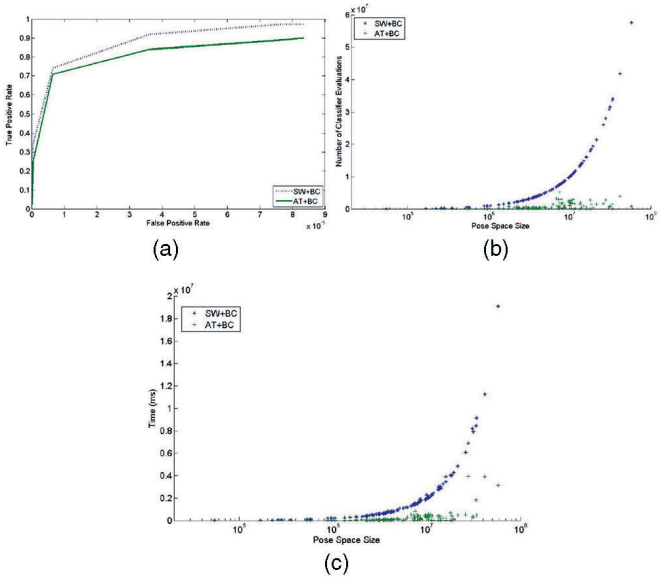


Fig. 5. (a) ROC for both the sliding window and the Active Testing approaches on the MIT+CMU frontal face data set. The AT algorithm achieves similar performance levels to the exhaustive search. (b) Number of classifier evaluations for each image in the test set. Clearly the AT approach does not suffer as much from the increase in pose space. (c) Time performance for each image in the test set.

the MIT+CMU data set [28]. This data set contains 130 images, of various sizes, where some images contain no faces and others contain an unknown number of faces. Face sizes range from 20 pixels to the width of images. As in the previous experiment, we initialize the AT algorithm similarly to that in Sections 2 and 3.

To find multiple face instances, we assume that at any point in time, the remaining number of faces to be found in an image follows a Poisson distribution with parameter λQ, where Q is the number of pixels unobserved in the image and λ is a face rate. We have chosen $\lambda = 10^{-4}$, corresponding to one face per 100×100 pixel image on average and hence $\pi_0(\Lambda_{1,1}) = e^{-\lambda Q}$. We then run the AT algorithm until $\pi_t(\Lambda_{1,1}) < \epsilon = 10^{-5}$. When a face is found, edges from the detected face region are removed from the integral images and the remaining poses are assigned uniform probability. The algorithm is then restarted with the updated $\pi_0(\Lambda_{1,1})$.

Fig. 5a shows the ROC curve of both the $(\mathrm{AT} + \mathrm{BC})$ and $(\mathrm{SW} + \mathrm{BC})$ methods on the MIT+CMU data set. In both cases, no postprocessing step was applied to these results (i.e., No NonMaximum suppression). First, we note that the MIT+CMU testset is much harder than the Caltech Frontal Face set. In general, the performance of the AT algorithm is comparable to the brute force approach. There is, however, a slight performance decrease in $(\mathrm{AT} + \mathrm{BC})$ when compared to the exhaustive search. That is, we notice that even though the classifier used (BC) is not very good (when compared to state-of-the-art classifiers), little accuracy loss is observed when used in the AT framework.

From this experiment, $(\mathrm{AT} + \mathrm{BC})$ required $O(10^8)$ classifier evaluations over the entire testset, while $(\mathrm{SW} + \mathrm{BC})$ required $O(10^9)$ evaluations. Fig. 5b shows the number of classifier evaluations required by both $(\mathrm{AT} + \mathrm{BC})$ and $(\mathrm{SW} + \mathrm{BC})$ on each image. Generally, we see that AT is still able to significantly reduce the total number of evaluations required even though the number of faces in the images is a priori unknown. Fig. 5c shows a similar result in terms of time. Again, computational gains are of one order of magnitude over the entire testset.

Notice in Figs. 5b and 5c that for images of the same pose space size, the number of classifier evaluations and time necessary for $(\mathrm{AT} + \mathrm{BC})$ to terminate vary. This variance is due to the fact that $(\mathrm{AT} + \mathrm{BC})$ stops when the estimate of having a face in the image is very low: $\pi_t(\Lambda_{1,1}) < \epsilon = 10^{-5}$. Hence, in images which contain many face-like features, the algorithm will need to visit many more locations to see if faces are still present. This is precisely what is observed in Figs. 5b and 5c.

5 Conclusion

We have proposed an Active Testing framework in which one can perform fast face detection and localization in images. In order to find faces, we use a coarse-to-fine method while sampling subwindows which maximize information gain. This allows us to quickly find the face pose by focusing on regions of interest and pruning large image regions. We show through a series of experiments that the active testing framework can be used to significantly reduce the number of classifier evaluations when searching for an object. Exponential speedup is observed when detecting and locating faces compared to the traditional sliding window approach (particularly on large image sizes), without significant loss in performance levels, indicating that this method is scalable to larger image sizes.

Acknowledgments

Funding for this research was provided in part by US National Institutes of Health Grant 1 R01 EB 007969-01 and the Duncan Fund for the Advancement of Statistics Research, Award 08-19.

References

[1] P. Viola and M. Jones, "Robust Real-Time Face Detection," *Int'l J. Computer Vision*, vol. 57, no. 2, pp. 137-154, 2004.
[2] H. Schneiderman, "Feature-Centric Evaluation for Efficient Cascaded Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.
[3] S. Li and Z. Zhang, "Floatboost Learning and Statistical Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112-1123, Sept. 2004.
[4] S. Yan, S. Shan, X. Chen, and W. Gao, "Locally Assembled Binary (LAB) Feature with Feature-Centric Cascade for Fast and Accurate Face Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-7, 2008.
[5] H. Rowley, S. Beluja, and T. Kanade, "Human Face Detection in Visual Scenes," *Proc. Advances in Neural Information Processing Systems*, pp. 875-881, 1996.
[6] M. Osadchy, Y. LeCun, and M. Miller, "Synergistic Face Detection and Pose Estimation with Energy-Based Models," *J. Machine Learning Research*, vol. 8, pp. 1197-1215, 2007.
[7] C. Gracia and M. Delakis, "Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1408-1423, Nov. 2004.
[8] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 130-136, 1997.
[9] S. Shah, S.H. Srinivasan, and S. Sanyal, "Fast Object Detection Using Local Feature-Based SVMs," *Proc. Int'l Workshop Multimedia Data Mining*, pp. 1-5, 2007.
[10] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 886-893, 2005.
[11] C. Lampert, M. Blaschko, and T. Hofmann, "Beyond Sliding Windows: Object Localization by Efficient Subwindow Search," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
[12] J. Yuan, Z. Liu, and Y. Wu, "Discriminative 3D Subvolume Search for Efficient Action Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
[13] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
[14] D. Geman and B. Jedynak, "Shape Recognition and Twenty Questions," INRIA Technical Report 2155, 1993.
[15] Y. Amit and D. Geman, "A Computational Model for Visual Selection," *Neural Computation*, vol. 11, pp. 1691-1715, 1999.
[16] F. Fleuret and D. Geman, "Coarse-to-Fine Face Detection," *Int'l J. Computer Vision*, vol. 41, pp. 85-107, 2001.
[17] F. Fleuret and D. Geman, "Stationary Features and Cat Detection," *J. Machine Learning Research*, vol. 1, pp. 2549-2578, 2008.
[18] D. Geman and B. Jedynak, "An Active Testing Model for Tracking Roads from Satellite Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 1-14, Jan. 1996.

[19] F.D. la Torre, J. Campoy, Z. Ambadar, and J. Cohn, "Temporal Segmentation of Facial Behavior," *Proc. Int'l Conf. Computer Vision*, 2007.
[20] X. Jiang, B. Mandal, and A. Kot, "Eigenfeature Regularization and Extraction in Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 383-394, Mar. 2008.
[21] L. Ding and A.M. Martinez, "Features versus Context: An Approach for Precise and Detailed Detection and Delineation of Faces and Facial Features," *IEEE Trans. Pattern Analysis and Machine Intelligence*, preprint, 2010, doi:10.1109/TPAMI.2010.28.
[22] P. Li and S. Prince, "Joint and Implicit Registration for Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
[23] V. Belle, T. Deselaers, and S. Schiffer, "Randomized Trees for Real-Time One-Step Face Detection and Recognition," *Proc. Int'l Conf. Pattern Recognition*, 2008.
[24] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results," http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html, 2010.
[25] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, Aug. 2001.
[26] M. Weber, "Frontal Face Dataset," California Inst. of Technology, http://www.vision.caltech.edu/html-files/archive.html, 1999.
[27] Intel, "Opencv Open Source Computer Vision Library."
[28] H. Schneiderman and T. Kanade, "Frontal Face Images," 2000.

Chapter 6

Unified detection and tracking of instruments during retinal microsurgery

Unified detection and tracking of instruments during retinal microsurgery

Raphael Sznitman, Rogerio Richa, Russell H. Taylor *Fellow, IEEE*, Bruno Jedynak and Gregory D. Hager, *Fellow, IEEE*

Abstract—Methods for tracking an object have generally fallen into two groups: tracking by detection, and tracking through local optimization. The advantage of detection-based tracking is its ability to deal with target appearance and disappearance, but does not naturally take advantage of target motion continuity during detection. The advantage of local optimization is efficiency and accuracy, but requires additional algorithms to initialize tracking when the target is lost.

To bridge these two approaches, we propose a framework for unified detection and tracking as a time-series Bayesian estimation problem. The basis of our approach is to treat both detection and tracking as a sequential entropy minimization problem, where the goal is to determine the parameters describing a target in each frame. To do this we integrate the Active Testing paradigm with Bayesian filtering, and this results in a framework capable of both detecting and tracking robustly in situations where the target object enters and leaves the field of view regularly. We demonstrate our approach on a retinal tool tracking problem and show through extensive experiments that our method provides an efficient and robust tracking solution.

Index Terms—Unified object detection and tracking, Active Testing, Instrument tracking, Adaptive Sensing, Retinal microsurgery.

1 Introduction

Visual tracking has been intensely studied in computer vision over the past two decades [1]. Informally, the objective of visual tracking is to provide an accurate estimate of the configuration of a target across time, where the term "configuration" denotes parameters describing the position, pose, or shape of an object. A general solution involves solving two subtasks: (i) detecting the target in the initial image in which it appears, and (ii) predicting and refining (*i.e.*, tracking) the configuration of the detected target in subsequent images [1], [2]. While extensive research in this area has produced excellent tracking systems, combining these two subtasks remains difficult when the target appearance is complex or when the target enters and leaves the field of view frequently.

Indeed, the initial detection of the target is often the most difficult aspect of a tracking system. This is particularly the case when object appearance is complex and many configuration parameters are involved [3]–[5]. Even more so, performing detection with accuracy and at frame rate for objects that have many pose parameters is often infeasible due to the enormous size of the search space *i.e.*, easily over a billion hypotheses. In addition, while object detection and localization algorithms, such as classifier cascades [6], [7] or branch-and-bound algorithms with SVM based cost functions [8] are expected to determine parameters that define an object (*i.e.*, a bounding box [6], or object segmentation [9]), incorporating prior object knowledge into these frameworks to improve detection in subsequent images is usually an ad-hoc adaptation of common filtering paradigms [10], [11].

Conversely, some tracking approaches have tried to place detection and tracking under the same umbrella. Approaches have included strategies for removing faulty detections by model validation and temporal non-maximal suppression [12]–[14]. However, in these cases only restricted regions of the image are considered when searching for the target, which often leads to tracking failures when motion models are violated. Other approaches have performed tracking by using cascades of detection processes that refine the possible object location [15], [16]. While these have generally been shown to be robust, the construction of these systems has been hand-crafted for each problem setting with no underlying principle. This in turn makes them challenging to implement in real-applications.

1.1 An Active Testing Approach

In this work, we propose an algorithmic solution for the task of detection and tracking. Our approach embodies and extends the *Active Testing* (AT) paradigm [3], [17] and allows both detection and tracking to be considered within the same framework. In particular, in both tasks, estimating the object parameters is achieved by using the same sequential entropy minimization procedure, and hence removes the need for two separate algorithms and the protocols necessary to join them. By using the AT framework and Bayesian filtering strategies, the entire search space of the object is always considered when searching for the object pose, and informative priors

R. Sznitman, R. Richa, R. H. Taylor and G. D. Hager are with the Dept. of Computer Science, B. Jedynak is with the Dept. of Applied Mathematics and Statistics Johns Hopkins University, Baltimore, MD, 21218, USA.
e-mail:{sznitman,richa,rht,bruno.jedynak,hager}@jhu.edu

Digital Object Indentifier 10.1109/TPAMI.2012.209

can effectively be used to weigh likely pose candidates in subsequent images. In addition, within the AT optimization, the parameters of the object are searched sequentially, requiring far fewer observation models when compared to [17]. Consequently, the learning stage of the framework is significantly simplified. Finally, central to the tracking problem, we detail how to incorporate traditional gradient-based tracking methods for this task [1], [18], [19] within our framework.

In summary, the unique aspects of our framework are: (i) an information-based heuristic is used to guide the search process during detection and tracking, allowing both to be solved using the same optimization strategy while considering the entire search space at all times, (ii) traditional local optimization tracking is incorporated into a larger class of image functions used to gather information regarding the location of the target and (iii) the process of learning observation models from training data is greatly simplified by introducing a new parametrization of these models.

1.2 Instrument Tracking in Surgery

We demonstrate our approach on the task of detecting and tracking a surgical instrument during retinal microsurgery. From a computer vision point of view, visual tracking of instruments is a challenging problem within a controlled environment. The instruments used during surgery are known *a priori*, making it possible to learn their appearance and geometry beforehand. However, the instruments are subject to a large variety of appearance changes during procedures, making tracking difficult. For example, an instrument may be partially blurred, the shadow of the instrument may be similar in appearance to itself, and local and global illumination conditions change with time. While one possibility would be to model the background and detect outliers to estimate the instrument pose as in [20], modeling the background in *in-vivo* settings remains challenging, particularly when the eye moves during the procedure.

In the context of surgical applications and with the goal of providing semi-automated assistance for clinicians during procedures, tool detection has received increased attention in recent years. Towards this end, a number of techniques have been proposed such as in [20], [21] where kinematic information and instrument templates were used to detect and track tools in image sequences. In [22]–[25], instrument models based on pixel, or local color are learned and used for detection and segmentation. Other methods, as in [26], [27] extracted edges and lines to ultimately infer tool tip position. Yet, to date a majority of methods have relied on the ability to alter the tool appearance directly by adding visible markers to facilitate tool detection [28]–[30].

Unlike other approaches for this task that demand prohibitively high computational costs [31] or extremely accurate initialization methods [19], [21], [32], our solution provides a feasible and automatic solution to tracking retinal instruments without the need for accurate instrument motion models. More importantly, this remains the case when the instrument enters and leaves the field of view often. To demonstrate this, our approach is validated on a microscope platform that uses a phantom eye and also on images from human retinal microsurgery. While a preliminary version of our method was presented in [33], this paper provides a full description of our approach, integrates gradient-based tracking methods in our framework, and presents extensive additional empirical results on both phantom eye data and human *in-vivo* data. While this work focuses on this particular application, the approach is relevant for a number of other applications as well.

The remainder of the paper is organized as follows: in Sec. 2 we first introduce some notation and describe the problem formulation. We then introduce tracking as a Bayesian sequential estimation problem and describe how our approach embodies this structure in Sec. 3. In Sec. 4 we describe the AT model for detecting retinal instruments. In Sec. 5, we perform extensive experimentation to validate our approach on both phantom and *in-vivo* data. Finally, we conclude with some closing remarks in Sec. 6.

2 Problem Formulation

The aim of this work is to locate a surgical instrument in a sequence of monocular images, gathered from the operating microscope. Similar to most detection and tracking approaches, we assume that the instrument's position and orientation[1], or *pose*, can be described by a relatively small number of parameters. As depicted in Fig. 1(*left*), we let the parameters representing the instrument be defined as $Y = (Y_1, Y_2, Y_3)$, where Y_1 corresponds to the instruments point of entry in the image (*i.e.*, a pixel location on the image boundary), Y_2 describes the angle the instrument makes with Y_1 and Y_3 is the instruments length measured in pixels. This particular parametrization is chosen as it is simple and intuitive to the retinal microsurgery application.

Ultimately, we are interested in determining the values of $Y^t = (Y_1^t, Y_2^t, Y_3^t)$, for all images in a sequence, $\mathcal{I}^T = (I^1, \ldots, I^T)$. For this reason, we treat Y^t as a random variable that must be inferred and where we want to compute $P(Y^t|\mathcal{I}^t), t = 1, \ldots, T$.

To do this, we first describe the pose space of the instrument. This is achieved in two steps. First, let the instrument's pose space when in the field of view, $\mathcal{S}_1$, be

$$\mathcal{S}_1 = \{[0, L] \times [-\pi/2, \pi/2] \times [\delta, D]\}$$

where L is the perimeter length of the image, δ and D are the minimum and maximum instrument lengths measured in pixels. In practice, δ is 10% of the width of the image.

1. The scale or size parameter is assumed to be known given that the tool must be in focus and microscopes used during procedures have very large focal lengths. Hence, we assume the tool scale is known.

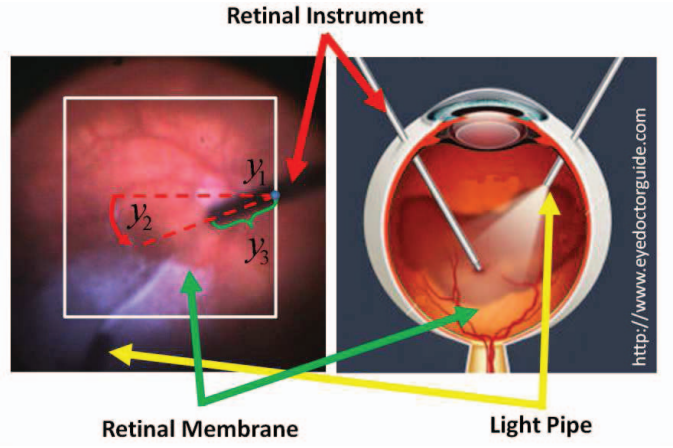


Fig. 1. (left) Tool parametrization: the retinal instrument has three parameters consisting of the point of entry of the instrument in the image, Y_1, the angle the instrument makes with the image boundary at the point of entry, Y_2 and the instruments length, Y_3. (right) Diagram of surgical environment, displaying the positioning of the light pipe and instrument during surgery.

Second, since the instrument may not be visible in the field of view of the camera, the separate space $\mathcal{S}_0 = \{\square\}$, is defined for this event (*i.e.*, the $\square$ is a token representing this case).

Finally, the complete pose space of the instrument is defined as $Y \in \mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_0$.

3 Active Testing for Tool Tracking

To detect and track an object, we cast the tracking problem in a Bayesian sequential estimation fashion [10], [11], [34]. That is, at time t, we must infer the random variable, Y^t, given the image sequence observed up to that time instant, $\mathcal{I}^t$. This can be expressed by,

$$P(Y^t|\mathcal{I}^t) = \int P(Y^t|Y^{t-1},\mathcal{I}^t)P(Y^{t-1}|\mathcal{I}^t)dY^{t-1} \quad (1)$$

$$\propto P(I^t|Y^t)\int P(Y^t|Y^{t-1})P(Y^{t-1}|\mathcal{I}^{t-1})dY^{t-1} \quad (2)$$

$$\propto P(I^t|Y^t)P(\hat{Y}^t) \quad (3)$$

where the conditional distribution given the observations can be rewritten as (1) by including the marginalization of Y^{t-1} and an application of Bayes theorem. (2) follows from (1) by another application of Bayes theorem, the assumption of Markov dynamics of the object, and the assumption that $P(Y^t)$ is a sufficient statistic for $\mathcal{I}^t$. In (3) we have defined $P(\hat{Y}^t) = \int P(Y^t|Y^{t-1})P(Y^{t-1}|\mathcal{I}^{t-1})dY^{t-1}$.

In most cases, various elements of the observation model, the dynamics and the distribution on Y have been approximated in order to allow both fast and feasible computation. For example, in methods based on Kalman filtering [10], the dynamics are assumed to be linear or are linearized, and both the distribution of Y and the observation model are Gaussian. In sampling-based methods [11], non-Gaussian distributions of Y are maintained by using particles. In our approach, we rely on a partitioning of $\mathcal{S}_1$ (using a conditional binary tree) to allow exact computation of posterior distributions and will be achieved by using a slightly adapted histogram filter [35], [36].

In the context of tracking, image observations are typically evaluated at a single location or a set of locations predicted by the distribution of the target (*i.e.*, at particle locations). But in the case of detection, following such a strategy is computationally hopeless as the number of hypotheses to evaluate is enormous. For example, when using particle filters, this would imply maintaining order the size of the pose space number of particles. For this reason, we need a mechanism to efficiently select which observations to make and the AT optimization scheme serves this purpose.

3.1 The Active Testing Model

The Active Testing (AT) [3], [17] can be viewed as an iterative stochastic optimization scheme aimed at reducing the uncertainty of a discrete random variable by sequentially asking "questions" or "queries" in an efficient fashion. In particular, this optimization scheme provides an approximation to the maximum likelihood estimate of the random variable when all possible questions are answered.

In general, the optimization process is as follows: one begins with a prior on the random variable to infer, p_0, and selects a subset of the pose space to query using a question regarding this subset. The question typically consists of computing a simple measurement on a region of the image, such as the proportion of pixels belonging to the object in some region of the image. Hence, a question is a coupling between a computation type *and* a region of the search space. The answer to the question is then used in a Bayesian way to recompute a new probability distribution or *posterior* distribution (*i.e.*, $p_1, p_2, p_3, \ldots$).

Selecting a new question for the following iteration, which we will denote as $\hat{X}$, is then achieved by choosing the question that reduces the expected entropy of the object as much as possible. This is equivalent to selecting the question that has the highest information gain,

$$\hat{X} = \arg\max_{X\in\mathcal{X}} MI(Y;X) \quad (4)$$

where MI is the mutual information [37] and $\mathcal{X}$ is the set of possible questions that are available. This procedure repeats until the entropy of the random variable drops below a pre-determined threshold, or a set number of questions have been asked.

Hence, in order to make use of the AT framework three pieces must be specified: (i) a prior distribution on the parameter, $P(\hat{Y}^t)$, (ii) a representation for the distribution of Y and (iii) a set of "questions" (and their

associated noise models), $\mathcal{X}$, pertaining to the parameter Y. We will specify these in Sec. 4.

3.2 Active Testing Filtering

At this point, we describe the general form of an Active Testing Filtering (ATF) algorithm (see Alg. 1). Here, the user initially provides an instrument dynamics model, $P(Y^t|Y^{t-1})$ and a prior on $P(Y^0)$. Then, for each image in the sequence, we first compute $P(\hat{Y}^t)$ (line 3) by using the provided dynamics model and the previous density. Depending on the model used, this can be computed in a number ways. We can then treat $P(\hat{Y}^t)$ as an initial prior for the AT optimization (line 4). That is, instead of using uninformative priors on the pose of the instrument for each image, we begin the AT optimization with $P(\hat{Y}^t)$, which carries information about where the instrument was previously located and how it may have moved.

Algorithm 1 Active Testing Filtering ($\mathcal{I}$)

1: Initialize: $P(Y^0), P(Y^t|Y^{t-1})$
2: **for all** I^t **do**
3: $\quad P(\hat{Y}^t) = \int P(Y^t|Y^{t-1})P(Y^{t-1})dY^{t-1}$
4: $\quad P(Y^t|\mathcal{I}^t) = ActiveTesting(I^t, P(\hat{Y}^t))$
5: **end for**

In this work, we select a simple linear instrument dynamics model of the form,

$$Y^{t+1} = AY^t + \mathcal{N}(0, \alpha) \tag{5}$$

where A is the dynamics transition matrix. In the experiments that follow, we use two different models: (i) A is the identity matrix which corresponds to assuming the tool has not moved from one frame to another. (ii) A is augmented to allow velocity estimates to be compounded in the new prior. Given that we know that the tool will enter and leave the field of view often, we expect both dynamic models to be consistently violated. While this may induce inappropriate priors $P(\hat{Y}^t)$, the active testing framework will still recover the pose of the instrument.

4 Active Testing Implementation

In this section, we describe the aspects of the AT optimization that must be specified: the representation of the probability distribution of Y and what "questions" will be available to localize an instrument. In particular, we will provide a set of questions, which can be viewed as a set of features that can be evaluated, combined and integrated by the framework and which are informative with respect to different coordinates of Y.

4.1 Probability Density Representation

To represent p_0 and the sequence of posterior distributions that will be computed, we make use of an abstract decomposition of the space $\mathcal{S}_1$. Let S denote a binary decomposition of the space $\mathcal{S}_1$. That is, S is a tree of subsets

$$S = \{S_{i,j}, i = 0, \ldots, H, j = 0, \ldots, 2^i - 1\} \tag{6}$$

The root of tree is $S_{0,0} = \mathcal{S}_1$ and $S_{i,j}$ is a subset of $S_{0,0}$. The decomposition of the tree is performed by splitting one coordinate of Y at a time, until a desired resolution is reached, at which point we repeat the procedure for another coordinate (*e.g.*, split Y_1, then Y_2 and so on). Simply put, the tree consists of a series of binary trees one after the other.

Fig. 2 depicts this decomposition visually. Here, we show a tree structure specified by (6), where blue nodes show where only Y_1 is being decomposed, red nodes show where Y_1 has been fully decomposed and Y_2 is in the process of being decomposed, and green nodes show those with both Y_1 and Y_2 fully decomposed and where only Y_3 is being refined.

Using this decomposition, describing the probability distribution of Y as a function of S is straightforward. If we denote the probability $P(Y \in S_{i,j})$ as $u_{i,j}$. Then since the node s$S_{i,j}$ are disjoint subsets, $u_{i,j} = u_{i+1,2j} + u_{i+1,2j+1}$ for every non-terminal (or leaf) node in S. Hence, observing the probability at a single level of S provides a piecewise constant representation of the distribution of Y.

Naturally, the space required to store this tree may be overwhelmingly large. For this reason, the tree will be generated in a lazy fashion and will allow us to represent the distribution of Y in a compact fashion. That is, the AT optimization will only begin with the root, $S_{0,0}$ and the tree will grow as questions are asked. This is closely related to Evolving Trees [38] which allow efficient organization of large amounts of data. Indeed, a key aspect of this method compared to classification trees, is that the construction of the tree is done online, dictated by the data at test time.

4.2 Set of Questions

To determine the pose of the instrument, the AT framework relies on the ability to ask "questions" about the content in the image. For a particular node $S_{i,j} \in S$, a "question" is a deterministic function of the image, $X_{i,j} : I_{S_{i,j}} \mapsto \mathbb{R}$, which computes a specific quantity from the image region specified by the pose subset $S_{i,j}$. The answers to the question $X_{i,j}$, denoted $Z_{i,j}$, is considered to be random and is interpreted in a probabilistic manner. In particular, when asking a question $X_{i,j}$ the answer, $Z_{i,j} = z$ is assumed to follow,

$$P(Z_{i,j} = z|Y) = \begin{cases} f_o(z; i, j) & \text{if } Y \in S_{i,j} \\ f_b(z; i, j) & \text{if } Y \notin S_{i,j} \end{cases} \tag{7}$$

where f_o and f_b are two distributions of responses, corresponding to the case where the instrument pose is in the space queried, and when it is not. These distributions are learned from representative labelled training

data and since the response, $Z \in \mathbb{R}$, both f_o and f_b are modelled as Gaussian (this will be detailed in the following subsection).

As in [17], we use two categories of questions in this framework: (i) noisy questions (Sec. 4.2.1) and (ii) noiseless or *oracle* questions (Sec. 4.2.2). These two categories of questions are motivated by the fact that oracle questions correspond to evaluating excellent, if not state-of-the-art, methods for detecting the target when looking at extremely small regions of the search space, while noisy questions help reduce the search space in order to ultimately use an oracle question. As shown in [17] this has the benefit of being computationally efficient when locating a target and allows for a simple mechanism to reject regions of the search space that have been observed fully. We now specify what questions are available in this application.

4.2.1 Noisy Questions

To detect our instrument we use five noisy questions such that each node of S only evaluates a single type of question. In particular, for $S_{i,j} \in S$ we denote the intervals for each coordinate of Y, as $Y_1 \in [a, b]$, $Y_2 \in [c, d]$ and $Y_3 \in [e, f]$. Also, we recall that δ is the minimum length the instrument must be protruding from the image boundary to be considered in the image, and let W be the width of the instrument. Fig. 2 visually depicts examples of where each question type (A through E) is evaluated in our decomposition and what computation is performed:

(A): In this question, $X_{i,j}$ computes the proportion of tool-like pixels in the region defined by $[a, b]$. This consists of a rectangular image patch along the boundary of the image, where the width of the patch is δ and is of length $[a, b]$.

Evaluating if a pixel belongs to the tool is achieved by evaluating if the RGB color of the pixel is likely to have come from a 3 dimensional Gaussian representing the instrument color. The parameters of the Gaussian are estimated using labeled training data, and pixels are classified as tool-like if their RGB color is within a fixed Mahalanobis distance to the Gaussian mean. The computed score is the proportion of tool-like pixels in the evaluated δ by $[a, b]$ patch.

(B): This question evaluates a series of template matches in order to estimate the precise location of the instrument entry point, Y_1. Centered on the boundary at the point $(a + b)/2$ and in increments of five degrees, we rotate a template of size $\delta \times 3W$, consisting of three $\delta \times W$ strips stacked together (*i.e.*, similar to [21]). At each rotation, we evaluate a template match, and then return maximum score observed over all evaluations.

The template match consists in evaluating a Haar-like feature [6] such that on each strip we sum the number of tool-like pixels using the color model described in question type (A), and subtract the sums of the two outer strips to that of the center region.

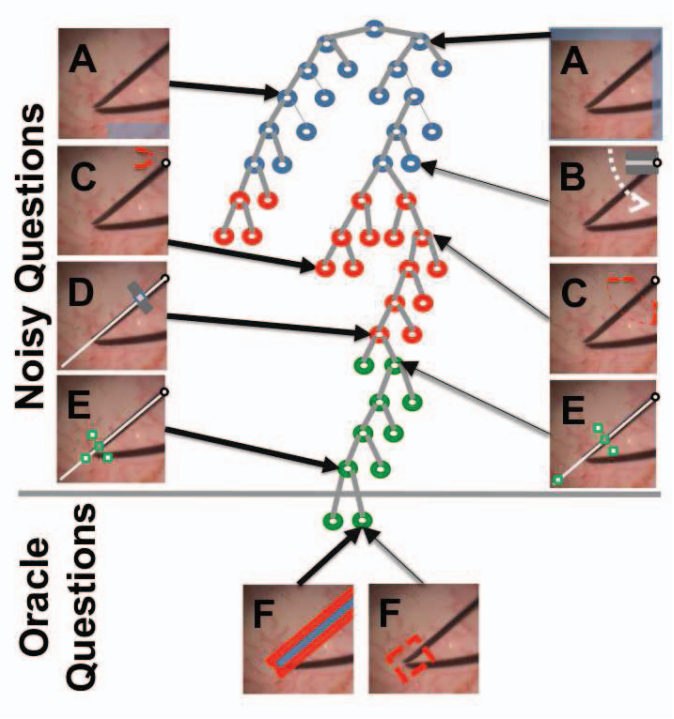


Fig. 2. Search space decomposition, density representation and image questions. The above tree represents a binary decomposition of the search space, where each node splits half the search space in two, by only spiting one coordinate of Y at a time. Blue, red and green nodes show when Y_1, Y_2 and Y_3 are being decomposed, respectively. By assigning the likelihood of the instrument parameters being within the search space of a given node, the tree S provides a representation of p_n. We also show examples of where different noisy question (types A through E) types can be evaluated, and what region of the image space they query. Oracle questions (type F) can only be evaluated at the leaf of the tree.

(C): This question also computes the proportion of tool-like pixels. As in (A), tool-like pixels are estimated by means of the same RGB color model. The region evaluated by this query type is defined by both $[a, b]$ and $[c, d]$. Defining the origin as the location on the boundary of the image $(a+b)/2$, we evaluate a restricted sector, by sweeping from c to d degrees and with length $\delta/2$ to δ to the origin. The proportion of tool-like pixels in this region is the computed score.

(D): Evaluates a template region, similar to that in (B), and returns the template matching score. A template of size $3W \times W$, consists of three $W \times W$ square regions stacked together. The template is positioned at a distance δ from boundary point $(a + b)/2$, and with an angle of $90 + (c + d)/2$ degrees (*i.e.*, perpendicular to the angle). The sum of tool-like pixels in the outer square regions is computed and subtracted to the sum in the inner square. Again, tool-like like pixels are computed as in (A).

(E): In this question, we evaluate a modified Haar-like feature. Along a line with intercept on the boundary at

$(a+b)/2$ and slope $(c+d)/2$, we position $W/2 \times W/2$ square regions at a distance of e and f pixels. The average intensity of each square is subtracted from each other. In addition, perpendicular to the slope and at a distance e to the boundary point, the average pixel intensity of two supplementary square regions of the same size are also computed and then subtracted to the previous two regions. This final score is then returned.

4.2.2 Oracle Questions

The second category of questions are assumed to be noiseless (*i.e.*, f_1 and f_0 do not overlap), and can only be evaluated at the leaves of our final tree, *i.e.*, when the pose of the instrument is explicitly hypothesized (see Fig. 2 question type (F)). Given that both the detection and tracking literature present a number of methods that appear to perform well in certain situations, we demonstrate the use of using two possible oracle questions: (i) template match which detects the pose of the instrument and (ii) gradient-based trackers typically used for traditional local tracking tasks. In our experiments, we show the effects and benefits of using either oracle:

Template Matching: We follow a similar approach to that of [21]. Given a specific hypothesis for the instrument location, we expect to find the instrument on the boundary at location $(a+b)/2$, with angle $(c+d)/2$ and length $(e+f)/2$. With the instrument width known, W, we perform a sum-of-squared difference (SSD) template match between the hypothesized pose and the projection of the tool-like color model on the image. That is, using the hypothesized pose, we construct an instrument template mask of width $3W$ and length $(e+f)/2+W$, with value 1 at instrument locations and 0 elsewhere. Placing the mask on the image, we apply the RGB color model to the overlapping regions of the image. The SSD is then computed from the projected tool-like pixel image and the constructed mask, and normalized by the total number of evaluated pixels. The final score is 1 if the normalized SSD is above a threshold and 0 otherwise.

Gradient-Based tracker: We use the recently developed Sum of Conditional Variance (SCV) objective function along with the ESM optimization strategy to refine the tool pose as proposed in [39]. The reference template used is an image patch describing the instrument tool tip, of size 40 by 40 pixels, extracted from the previous frame. Once the optimization scheme finishes, we apply a normalized cross-correlation template match (returning "yes or "no") to verify good convergence (which occurs when the score is above a threshold).

Naturally, many different classifiers or trackers could be substituted for those chosen here. Our aim is to show how to incorporate different oracles within our framework.

Note that if an oracle responds "yes" to any question regarding a hypothesized pose, then the posterior distribution becomes a Dirac (*i.e.*, all probability mass is concentrated on a single pose) because the noise models have non-overlapping support. Consequently, the entropy of the ensuing posterior distribution is zero and the algorithm terminates. Hence, having a good oracle questions is crucial to avoid the algorithm from finishing prematurely or erroneously.

4.3 Learning noise models

As described in the previous section, each node $S_{i,j} \in S$ has an associated question, $X_{i,j}$ with a corresponding noise model (densities (f_o, f_b) from (7)). Given that these densities are indexed by (i,j), it would appear as though a separate noise model for each node in S is required. Considering the size of the pose space, the quantity of training data to achieve this would be overwhelming.

In [17], the problem is somewhat avoided by using folded models, that take advantage of translational invariances within levels of the hierarchy. However, this trick is not possible in this setting given that the pose space is much larger and S does not maintain the same invariance properties.

To avoid the problem here, we propose to parametrize the noise models and interpolate the parameters based on the position of a node in the tree. For example, let us consider the answer to the noisy question of type (A) in Fig. 2. Given that the size of the object is known, we can expect to see a certain number of object-like pixels in the queried region. Similarly, if no object were present, then we would expect a much smaller number of object-like pixels to be found in the queried region. In addition, if the queried region were twice as large, the same intuition would still apply. For this reason our noise models are of the form,

$$\begin{aligned} f_b(x;i,j) &= \mathcal{G}(x;\mu_0|X_{i,j}|,\sigma_0|X_{i,j}|^2) \\ f_o(x;i,j) &= \mathcal{G}(x;\mu_1|Y|+\mu_0(|X_{i,j}|-|Y|), \\ &\quad \sigma_1|Y|^2+\sigma_0(|X_{i,j}|-|Y|)^2) \end{aligned} \tag{8}$$

where $\mathcal{G}(\cdot)$ is a Gaussian distribution, $|X_{i,j}|$ and $|Y|$ are the number of pixels contained in $X_{i,j}$ and the estimated instrument size in the image, respectively.

The parameters (μ_1,σ_1) and (μ_0,σ_0) are the means and variances for the likelihood of observing tool-like pixels for a given question type (assumed to be Bernoulli random variables). As such, any blue node in Fig. 2 has the same noise model as any other node of the same color, with its parameters interpolated based on its placement in the tree. Modeling the noise this way has the added benefit of being invariant to the fineness of the decomposition the pose space. For example, if the depth of the tree changed (*e.g.*, image is twice as large), we would not need to learn new noise models.

In practice, this type of noise model is learned for each of the noisy questions. While this clearly does not benefit questions of type (B), (D) and (E) (since they are always computed over the same sized area), the number of parameters needed to learn is greatly reduced for questions of type (A) and (C). As such, only four

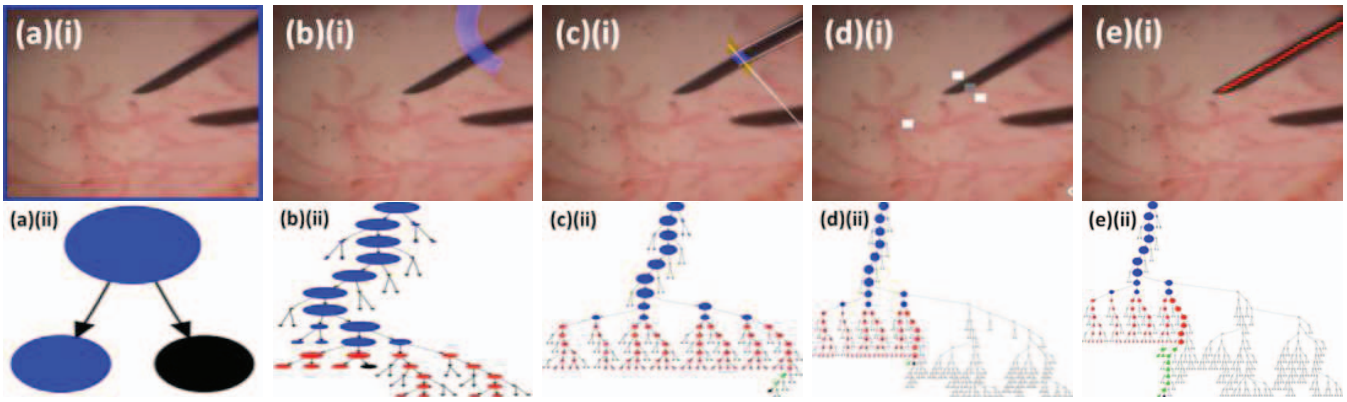


Fig. 3. Active Testing Iterations. Each image pair (top and bottom row) shows a question being evaluated and the corresponding state of the tree S at that point in time. See Video 1 for the entire image sequence.

parameters need to be learned for each question type and can be achieved by using an extremely small number of training images (*i.e.*, 20 labeled training images).

5 Experiments

In the following section we show how our approach performs on a live phantom eye platform, as well as on human *in-vivo* images. In both cases, we show qualitative and quantitative results of our method, and specify typical situations where our approach has difficulties maintaining accurate tracking.

Our framework was implemented on a Dell Precision PC with a Xeon 2.13GHz processor. The algorithm is coded in C++ and uses Opencv and the CISST library [40] for image acquisition and handling. Our PC is connected to receive video from a Grasshopper camera that is coupled to a microscope. The images acquired are 1600×1200 pixels large, and are captured at 30fps. The region of interest for the AT optimization is of size 256×256 and hence $Y_1 \in [1, L = 256 + (3 \times 255) = 1021]^2$.

The initial distribution of Y, $P(Y^0)$ is set to be an unbiased prior on the pose of the instrument. That is, $P(Y \in \mathcal{S}_1) = P(Y \in \mathcal{S}_0) = 1/2$, indicating that *a priori*, the instrument has equal likelihood of being in or not in the image. Note that we assign this probability at the root of S and assume uniform decomposition of the probability mass. While a small number of nodes may therefore be attributed with non-sensible probability, the practicality of this approximation is beneficial given that computing the exact probability is non trivial, and would be time consuming.

Finally, two versions of the algorithm are implemented. The first, **ATF-match**, uses the template match oracle question and the second, **ATF-track**, uses the gradient-based tracker oracle (as described in Sec. 4.2.2). With the exception of Sec. 5.1.2, where we observe the effect of different instrument motion models, we fix A to be the identity matrix (see Eq. (5)).

2. This is similar to the size of regions of interest during clinical procedures.

5.1 Phantom Eye Platform

We begin by providing some qualitative results as to how the proposed approach detects and tracks a surgical instrument in a phantom eye. To provide some intuition to the sequential nature of the AT algorithm, we have provided Video 1 (see additional videos) to visually depict both the questions asked and the evolution of S at each iteration of the AT optimization. Some snapshots of this video are shown in Fig. 3. The top row shows what question is being evaluated and the associated queried region (highlighted in each image) at given iterations of the optimization. The bottom row shows the corresponding evolution of the state of S. Here, the area of each node shown is proportional to the mass contained for that pose subset, and the color of each node represents which coordinate is being refined (as in Fig. 2). Additionally, the black node indicates which node is to be evaluated next.

Initially, only the root $S_{0,0}$ exists and is questioned. Having created children (Fig. 3(a)i-ii), the size of S is of three nodes. After a few questions, the tree has grown and refined itself past the first coordinate Y_1 and onto Y_2 and Y_3 (Fig. 3(b-c)i-ii). Eventually the correct Y_2 parameter (Fig. 3(d)i-ii) is refined, leading to a valid tool detection (Fig. 3(e)i-ii). Note that nodes which are extremely small are pruned as in [17]. This allows our search space to remain tractable and computationally manageable.

We also provided Videos 2-7, which show how our algorithm detects and tracks retinal instruments in our phantom environment. Fig. 4 shows a few snapshots from these sequences. The recorded sequences cover a wide range situations typically observed during retinal microsurgery: different types of instruments to track, severely blurred instruments, challenging non homogeneous illumination, no instrument in the field of view of the camera and the instrument shadow being present. In each image we have overlayed the AT search domain with a green box (except for (b) and (g) where the AT search domain is the entire image shown).

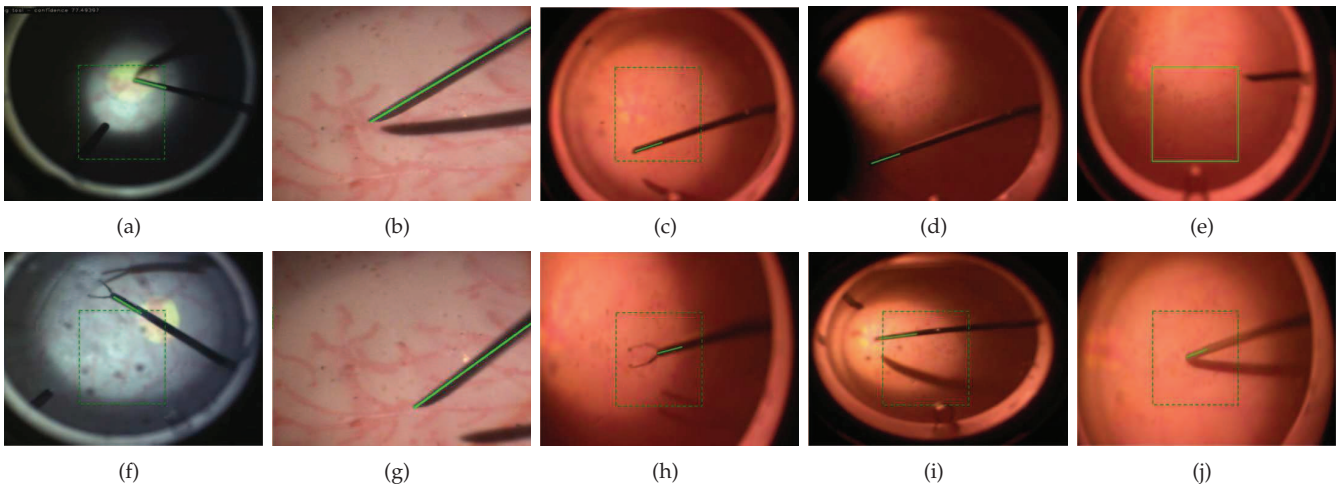


Fig. 4. Visual tracking examples in phantom environment. A variety of visual conditions typically encountered during clinical procedures are shown. The green box depicts the AT search region on our phantom platform.

5.1.1 Empirical Comparison

To evaluate the performance of the ATF approach, we compared it with several other methods: the Active Testing (AT-match) approach described above with a template match oracle but without any filtering (following [17], a Color-based Detector (CBD) as that in [41]), a Line-based tracker (LBT) (similar to [21]), and a particle filter [11] using our template matching oracle. We now further detail these methods.

The CBD [41] is a color based detector that requires a color model to evaluate the presence of tool-like pixels. Here, we set this color model to be the same as that used by our framework and at evaluation time, the instrument is segmented using this model. In addition to this, we also estimate the tool tip position by marching from the segmented centroid, along the direction of largest variance until a segmentation boundary is encountered. This position is considered to be the instrument tip. The LBT (following [21]) is a strip tracker that performs gradient based tracking on the color segmented image and a binary mask of the instrument. Tool motion is modelled with an image-plane rotation and translation vector. The particle filter [11] was set to use 1000 particles to maintain the distribution of the instrument, which was parametrized as in this application. The observation model consisted of using the same template match as in our framework, and used the same motion model as well. For both the LBT and the particle filter, initialization was performed by using the CBD.

To compare performances, we annotated by hand the location of the tool (*i.e.*, $Y = (Y_1, Y_2, Y_3)$) in an image sequence of over 400 images. These annotations provided ground truth for quantitative algorithm comparison. We then evaluate each approach by observing the error in the estimates of each parameter and the tool tip position, as well as the *true positive rate* (TPR), the *false positive rate* (FPR) and the precision for each approach (where a correct detection is where the estimated the tool tip location is within 10 pixels of the ground truth). The average time required by each method to find the location of the target for a single frame was also computed. Table 1 summarizes these results for each evaluated method. For the accuracy errors, we report the means and standard errors (in bracket) for each instrument parameter and tip.

In terms of coordinate accuracy, we can observe that the ATF methods generally performs better than the alternative methods. In particular, ATF-track performs better than other approaches when estimating the instrument tip position. This overall improvement can be attributed to the sequential parameter estimation approach that the active testing framework conducts. By estimating the first parameter, then the second and so on, each parameter is individually estimated accurately. This is in sharp contrast to the more direct LBT approach which locates the tool tip, and then estimates the necessary parameters, or the particle filter which simply samples the space directly.

In terms of detection accuracy, we notice that all methods tested provide more or less the same detection accuracy, with the exception of ATF-track which is significantly better than the others. This increase in precision is most likely due to the gradient-based tracker oracle question used. Also, we see that detection is significantly slower than tracking, as both AT-match and CBD run at much slower rates than the tracking algorithms. This confirms the advantage of tracking strategies over tracking by pure detection.

When comparing AT-match and ATF-match, we note that both methods perform similarly from an accuracy and detection point of view. However, we note that their speeds differ. Indeed, ATF-match is significantly faster than AT-match. This is most likely due to the use of informative priors. In fact, counting the number of nodes in final trees across all images, ATF-match trees have on average 75 nodes, while AT-match trees have around 210

TABLE 1
Comparison of algorithms. We show pixel accuracy of each method when estimating different parameters of the instrument, as well as the detection accuracy and time necessary for each method to process one frame.

	Accuracy Error				Detection Accuracy			
Method	Y_1	Y_2	Y_3	Tip	TPR	FPR (10^{-6})	Precision	Time (ms) per frame
ATF-track	4.13 (24)	2.42 (0.07)	**4.94 (0.27)**	**6.78 (0.6)**	**0.975**	**2.8**	**0.811**	8.0
ATF-match	2.97 (0.4)	2.37 (0.1)	14.11 (0.9)	11.03 (0.9)	0.839	6.6	0.631	8.54
AT-match	3.74 (0.4)	**2.01 (0.1)**	12.89 (0.8)	13.45 (0.8)	0.812	6.1	0.618	25.21
CBD	83.73 (2.4)	29.44 (0.8)	20.25 (0.7)	15.15 (0.79)	0.783	7.1	0.548	26.67
LBT	50.94 (7.3)	11.03 (0.9)	21.27 (2.1)	11.42 (0.2)	0.839	7.4	0.597	**4.8**
Particle Filter	**1.08 (0.05)**	11.53 (0.35)	6.64 (0.34)	6.91 (0.31)	0.841	2.9	0.783	6.28

nodes. This is a significant difference in the number of operations required to update the posterior distribution at each iteration of the AT optimization and accounts for the difference in speed between AT-match and ATF-match.

Given that our goal is to provide a tracking system, we would also like to have an understanding of how our system performs consecutively. To summarize this ability, we consider the event of correctly detecting a number of consecutive frames to follow a Geometric probability distribution. That is, with some probability ϵ, we correctly find the pose of the instrument in the next frame. Hence good tracking should be characterized by large values of ϵ. Computing this for each method, we find that ATF-tracker has the largest value with 0.98, followed by ATF-match (0.94), Particle Filter (0.93), LBD (0.92), AT(0.82) and CBD (0.82).

5.1.2 Alternative Tool Dynamics Model

We now briefly explore the effect of different instrument dynamics models (see (5)). As described in Sec. 3.2, we propose using two A matrices: (i) the identity (used until now) or (ii) augmented with velocity information.

Table 2 also shows a resume of the performance differences between the two proposed models. One can notice that in either case, the performances of the algorithms are extremely similar to each other. Most noticeably, we see that in terms of time, both methods run at approximately the same speed. This suggests that the dynamic models used in either case do not inhibit instrument localization and nor do they improve performance. This leads us to believe that the AT optimization ultimately is what provides timely solutions, rather than precise instrument motion models. Note that it could still be the case that alternative dynamics models could provide improvements in some cases.

5.2 Human In-Vivo Images

To validate the suitability of this approach for clinical settings, we evaluated our system on a human *in-vivo* image sequence. Our system was setup with the same parameters as previously described and then evaluated on 850 images. The initial 40 frames of the sequence were used for training purposes and were not included in the testing of our method.

Video. 8 show how our framework performs on this data and snapshots of this video are shown in Fig. 5. Here, we can see that even in situations where smoke is present, or when shadows overlap instrument regions considerably tracking is maintained and the instrument tip is accurately found. While this sequence is significantly more challenging than those acquired in our phantom experiments, reliable tracking is achieved for significant portions of this sequence.

However, as shown in Fig. 5, there are situations where our system fails to provide correct instrument pose. In particular, we can identify two such causes:

- Tool appearance changes due partial illumination variation. In some cases, the illumination on the instrument is not regular. Coupling this with the instrument tip appearing blurry (out of focus), our algorithm has difficulties precisely localizing the instrument tip, as depicted in Fig. 5(e).
- Poor oracle question. When using the gradient-based tracker oracle question, a threshold is used to validate valid convergence. Incorrect thresholds can lead to saying that the instrument is at a particular location when it is in fact not. As shown in Fig. 5(f), this may lead to being "stuck" on irrelevant image regions.

To relate the effectiveness of our method in this scenario to that reported on phantom data, we computed similar performance measures as done previously[3]. In all categories computed AT-track performed better than ATF-match (TPR; 0.6 vs 0.1. FPR; 5.1 vs 7.8 $\times 10^{-6}$. Precision; 0.49 vs 0.12. Accuracy Tip; 37.3 vs 82.3. ϵ; 0.65 vs 0.49.). These results indicate two distinct points. First, the task of detecting and tracking instruments is substantially more difficult in *in-vivo* sequences than in phantom sequences and is apparent from the drop in performances across all measures when compared to Table. 1. Second, the template match oracle, ATF-match, is in effect not capable of accurately detecting and tracking the instrument in this sequence. For this reason, in challenging tracking tasks such as the one at hand, the possibility of relying on successful gradient-based trackers is of great benefit.

3. Note that the CBD could not locate the instrument in an overwhelming number of images in the *in-vivo* sequence. Given that it initializes both the LBT and the particle filter, quantitative evaluation of these methods has been omitted here.

TABLE 2
Comparison of instrument dynamics models. Two instrument transition models, A, are tested.

A	Identity	Augmented	A	Identity	Augmented
Y_1	2.97 (0.4)	**2.63 (0.1)**	TPR	0.839	**0.842**
Y_2	2.37 (0.1)	**1.72 (0.08)**	FPR (10^{-6})	6.6	**6.4**
Y_3	14.11 (0.9)	**11.65 (0.7)**	Precision	0.631	**0.662**
Tip	11.03 (0.9)	**10.51 (0.7)**	Time (ms)	8.54	**7.21**

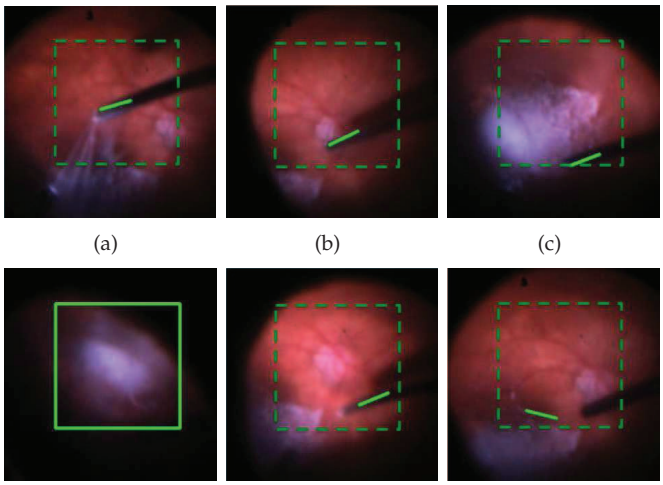


Fig. 5. Visual tracking example in a human *in-vivo* image sequence. The green region depicts the region considered by the AT optimization. (e-f) show two different cases where our system fails (see text for details).

6 Conclusion and Future Work

In this paper, we have a proposed a novel approach for the task of instrument detection and tracking in retinal microsurgery. By using the Active Testing paradigm, both these tasks can be treated as the same sequential parameter estimation problem, as opposed to two separate algorithmic tasks. Using filtering techniques, we have also shown how to effectively incorporate previous instrument information for the task of tracking. We have experimentally shown that the presented algorithm is capable of detecting and tracking retinal tools efficiently and robustly in cases where the object enters and leaves the field of view frequently. This has been demonstrated on both a live platform and on human *in-vivo* images. While presented in the context of retinal microsurgery, we are confident that this approach may apply to other surgical procedures, as well as for other object categories. Future work in this area will be directed to extending this method to stereo image sequences, as well as modeling illumination changes more consistently.

Acknowledgment

Funding for this research was provided in part by NIH Grant R01 EB 007969-01 and internal JHU funds. We would also like to thank Dr. Jim Handa MD and Dr. Peter Gehlbach MD for their insightful help. Marcin Balicki and Kevin Olds are credited for the development of the phantom eye used.

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Survey*, vol. 38, no. 4, 2006.

[2] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2000, pp. 142 –149.

[3] D. Geman and B. Jedynak, "An active testing model for tracking roads from satellite images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 1–14, 1996.

[4] O. Bernier, P. Cheung Mon Chan, and A. Bouguet, "Fast non-parametric belief propagation for real-time stereo articulated body tracking," *Journal of Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 29–47, 2009.

[5] W. Geng, P. Cosman, C. C. Berry, Z. Feng, and W. R. Schafer, "Automatic tracking, feature extraction and classification of c. elegans phenotypes," *IEEE Transaction on Biomedical Engineering*, vol. 51, pp. 1811–1820, 2004.

[6] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[7] A. Vedaldi, G. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009, pp. 606–613.

[8] C. Lampert, M. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[9] D. Aldavert, A. Ramisa, R. Toledo, and R. Mantaras, "Fast and robust object segmentation with the integral linear classifier," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2010, pp. 1046–1053.

[10] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[11] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 28, no. 1, pp. 5–28, 1998.

[12] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[13] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009, pp. 1515–1522.

[14] S. Shahed Nejhum, J. Ho, and Y. Ming-Hsuan, "Visual tracking with histograms and articulating blocks," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[15] R. Verma, C. Schmid, and K. Mikolajczyk, "Face detection and tracking in a video by propagating detection probabilities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1215–1228, 2003.

[16] K. Toyama and G. D. Hager, "Incremental focus of attention for robust vision-based tracking," *International Journal of Computer Vision*, vol. 35, no. 1, pp. 45–63, 1999.

[17] R. Sznitman and B. Jedynak, "Active testing for face detection and localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1914–1920, 2010.

[18] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework part 1: The quantity approximated, the warp update rule, and the gradient descent approximation," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[19] R. Richa, M. Balicki, E. Meisner, R. Sznitman, R. H. Taylor, and G. D. Hager, "Visual tracking of surgical tools for proximity detection in retinal surgery," in *Information Processing in Computer Assisted Interventions*, vol. 6689, 2011, pp. 55–66.

[20] R. Sznitman, H. Lin, M. Gupta, and G. Hager, "Active background modeling: Actors on a stage," in *International Conference on Computer Vision Workshops*, 2009, pp. 1222–1228.

[21] D. Burschka, J. Corso, M. Dewan, W. Lau, H. Li, H. Lin, P. Marayong, N. Ramay, G. Hager, B. Hoffman, D. Larkin, and C. Hasser, "Navigating inner space: 3-d assistance for minimally invasive surgery," *Robotics and Autonomous Systems*, vol. 52, pp. 5–26, 2005.

[22] D. Uecker, C. Leem, Y. Wang, and Y. Wang, "Automated instrument tracking in robotically assisted laparoscopic surgery," *Journal of Image Guided Surgery*, vol. 22, no. 6, pp. 429–437, 1995.
[23] Y. Wang, D. Uecker, and Y. Wang, "A new framework for vision-enabled and robotically assisted minimally invasive surgery," *Computerized Medical Imaging and Graphics*, vol. 1, no. 6, pp. 308–325, 1998.
[24] C. Doignon, F. Nageotte, and M. de Mathelin, "Detection of grey regions in color images: application to the segmentation of a surgical instrument in robotized laparoscopy," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, 2004, pp. 3394–3399.
[25] S. J. McKenna, H. Nait-Charif, and T. Frank, "Towards video understanding for laparoscopic surgery: instrument tracking," in *Image and Vision Computing New Zealand Conference*, 2005.
[26] J. Climent and P. Mares, "Automatic instrument localization in laparoscopic surgery," *Electronic Letters on Computer Vision and Image Analysis*, vol. 4, no. 1, pp. 21–31, 2004.
[27] S. Voros, J. A. Long, and P. Cinquin, "Automatic detection of instruments in laparoscopic images: A first step towards high-level command of robotic endoscopic holders," *International Journal of Robotic Research*, vol. 26, no. 11-12, pp. 1173–1190, 2007.
[28] A. Casals, J. Amat, and E. Laporte, "Automatic guidance of an assistant robot in laparoscopic surgery," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1996, pp. 895–900.
[29] G. Wei, K. Arbter, and G. Hirzinger, "Real-time visual servoing for laparoscopic surgery. controlling robot motion with color image segmentation," *IEEE Engineering in Medicine and Biology Magazine*, vol. 16, no. 1, pp. 40–45, 1997.
[30] O. Tonet, T. U. Ramesh, G. Megali, and P. Dario, "Image analysis-based approach for localization of endoscopic tools," in *Proceedings of Surgetica*, 2005, pp. 221–228.
[31] Z. Pezzementi, S. Voros, and G. Hager, "Articulated object tracking by rendering consistent appearance parts," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009, pp. 3940–3947.
[32] M. Dewan, P. Marayong, A. M. Okamura, and G. D. Hager, "Vision-based assistance for ophthalmic microsurgery," in *Medical Image Computing and Computer Assisted Intervention*, 2002, pp. 49–57.
[33] R. Sznitman, A. Basu, R. Richa, J. Handa, P. Gehlbach, R. T. H., B. Jedynak, and G. D. Hager, "Unified detection and tracking in retinal microsurgery," in *Medical Image Computing and Computer Assisted Intervention*, 2011, pp. 1–8.
[34] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
[35] G. Hager, *Task-Directed Sensor Fusion and Planning: A Computational Approach*. Springer, 1990.
[36] N. S. Peng, J. Yang, and Z. Liu, "Mean shift blob tracking with kernel histogram filtering and hypothesis testing," *Pattern Recognition Letters*, vol. 26, no. 5, pp. 605–614, 2005.
[37] T. M. Cover and J. A. Thomas, *Elements of information theory*. Wiley-Interscience, 1991.
[38] J. Pakkanen, J. Iivarinen, and E. Oja, "The Evolving Tree — a novel self-organizing network for data analysis," *Neural Processing Letters*, vol. 20, no. 3, pp. 199–211, 2004.
[39] R. Richa, R. Sznitman, R. H. Taylor, and G. D. Hager, "Visual tracking using the sum of conditional variance," in *IEEE Conference on Intelligent Robots and Systems*, 2011, pp. 2953–2958.
[40] P. Kazanzides, S. DiMaio, A. Deguet, B. Vagvolgyi, M. Balicki, C. Schneider, R. Kumar, A. Jog, B. Itkowitz, C. Hasser, and R. H. Taylor, "The surgical assistant workstation (saw) in minimally invasive surgery and microsurgery," in *MICCAI Workshop - Systems and Architectures for Computer Assisted Interventions*, 2010.
[41] R. Sznitman, D. Rother, J. Handa, P. Gehlbach, G. D. Hager, and R. H. Taylor, "Adaptive multispectral illumination for retinal microsurgery," in *Medical Image Computing and Computer Assisted Intervention*, 2010, pp. 465–472.

Raphael Sznitman received his B.Sc. in cognitive systems from the University of British Columbia in 2007. Following this, he received his M.Sc and PhD in computer science from the Johns Hopkins University in 2011. Currently, he is a postdoctoral fellow at the Ecole Polytechnique Federale de Lausanne (Switzerland) where he works in the computer vision laboratory. His research interests are primarily in computational vision, probabilistic methods and statistical learning, applied to visual tracking and object detection.

Rogerio Richa Rogerio Richa received his Ph.D. from the LIRMM (Laboratoire d'Informatique, Robotique et Microelectronique de Montpellier) in 2010 for his work in robust visual tracking for beating heart surgery. He then joined the LCSR (Laboratory of Computational Sensing and Robotics) at Johns Hopkins where he is currently working on the development of a robotic platform for retinal surgery. His main interests are computer vision, medical imaging and surgical robotics.

Russell H. Taylor Russell H. Taylor (Fellow, 1994) received his Ph.D. in Computer Science from Stanford in 1976. He joined IBM Research in 1976, where he developed the AML robot language and managed the Automation Technology Department and (later) the Computer-Assisted Surgery Group before moving in 1995 to Johns Hopkins, where he is a the John C. Malone Professor of Computer Science with joint appointments in Mechanical Engineering, Radiology, and Surgery and is also Director of the Engineering Research Center for Computer-Integrated Surgical Systems and Technology (CISST ERC). He is the author of over 275 peer-reviewed publications, a Fellow of the IEEE, of the AIMBE, of the MICCAI Society, and of the Engineering School of the University of Tokyo. He is also a recipient of numerous awards, including the IEEE Robotics Pioneer Award, the MICCAI Society Enduring Impact Award, and the Maurice Müller Award for Excellence in Computer-Assisted Orthopaedic Surgery.

Bruno Jedynak received his doctorate in Applied Mathematics from the University Paris Sud. His dissertation was performed at INRIA (Rocquencourt, France). After spending a year as post-doc in the Department of Statistics at the University of Chicago, he was appointed Maitre de conferences at the Universite des Sciences et Technologies de Lille. He is currently a faculty member of the Department of Applied Mathematics and Statistics at The Johns Hopkins University (JHU). He is also a faculty member of the Center for Imaging Science at JHU.

Gregory D. Hager Gregory D. Hager is a Professor and Chair of Computer Science at Johns Hopkins University and the Deputy Director of the NSF Engineering Research Center for Computer Integrated Surgical Systems and Technology. His research interests include time-series analysis of image data, image-guided robotics, medical applications of image analysis and robotics, and human-computer interaction. He is the author of more than 220 peer reviewed research articles and books in the area of robotics and computer vision. In 2006, he was elected a fellow of the IEEE for his contributions in Vision-Based Robotics.

Chapter 7

Model-based classification trees

Model-Based Classification Trees

Donald Geman * Bruno Jedynak †

March 2000

Abstract

The construction of classification trees is nearly always top-down, locally optimal and data-driven. Such recursive designs are often globally inefficient, for instance in terms of the mean depth necessary to reach a given classification rate. We consider statistical models for which exact global optimization is feasible, and thereby demonstrate that recursive and global procedures may result in very different tree graphs and overall performance.

Keywords. Classification tree, optimal testing strategy, dynamic programming, pattern recognition.

*Department of Mathematics and Statistics, University of Massachusetts, Amherst, MA 01003; Email:geman@math.umass.edu. Supported in part by the NSF under grant DMS-9217655, ONR under contract N00014-97-1-0249, Army Research Office under MURI grant DAAH04-96-1-0445,

†Département de probabilités et statistiques, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq Cedex; Email:Bruno.Jedynak@univ-lille1.fr

1 Introduction

Most of the literature on classification (or decision) trees is about inducing them from a training set $\mathcal{L}$ of labeled feature vectors in order to classify unlabeled data. Usually a tree T_{loc} is built in a top-down, recursive fashion from a pool of "tests" ("experiments," "questions") which are functions of a single feature. First the root is assigned a test, then each child of the root, and so forth until a stopping rule is enforced. At each internal node, each test in the pool is ranked according to a criterion based on information gain (e.g., entropy reduction) and the best test is assigned to the node; the gains are estimated from $\mathcal{L}$. The construction $\mathcal{L} \rightarrow T_{loc}$ is then data-driven and based on local optimization. Performance is often measured by classification error, and sometimes also by the efficiency of the representation (for example expected depth). Two seminal works are [8] and [25], and applications are numerous in statistics, pattern recognition, machine learning and other fields.

An alternative approach - the one here - begins with a statistical model $\mathcal{M}$ for the joint distribution of the tests and the classes (labels); then a tree T_{glo} is characterized by a global criterion for efficient classification. The construction $\mathcal{M} \rightarrow T_{glo}$ is then model-driven and based on global optimization. The model $\mathcal{M}$ might be estimated from data or derived in a Bayesian sense from a "forward model" for the distribution of the data given the class together with a "prior model" for the marginal distribution of the class variable. The optimality criterion might involve a tradeoff between accuracy (e.g., measured by the average entropy at the leaves or misclassification error) and computation (e.g., the average number of tests performed). A different notion of optimality based on efficient coding is discussed in [26]. Generally, calculating optimal trees is computationally prohibitive, whether model-driven or data-driven, and the literature is correspondingly sparse; see [18], [22] and the approximations in [26].

Our goal is to demonstrate that, in the model-based situation, the performance of tree classifiers based on recursive designs, e.g., stepwise entropy reduction, can be markedly inferior to those based on global designs. (The same is true of data-

driven trees, although this is more difficult to demonstrate as explicitly.) Another analysis of this discrepancy appears in the work of Garey [17] and others in the special case in which the test outcomes are determined by the classes ("constrained twenty questions"). The difference is especially pronounced with skewed priors, i.e., when *a priori* some classes are much more likely than others.

A simple example is given in Figure 4 for a model $\mathcal{M}$ with two classes $\{a, b\}$, one of which (class a) is rare; T_{loc} is on the left and minimizes entropy level-by-level and T_{glo} is on the right and minimizes a criterion based on both accuracy and computation. Both trees have the same error rate, but the *expected* depth of T_{loc} is about twice that of T_{glo}, and the testing strategy in T_{glo} is virtually the "opposite" of the greedy one. The expected depth necessary to reach a given level of accuracy is of particular importance when the tests are costly or when $\mathcal{L}$ is small and hence the estimation of information gains quickly becomes unstable. In another example (see Figure 5) the prior is uniform, both trees have average depth around ten, but the error rate of T_{loc} is many times that of T_{glo}.

Exact computations of optimal strategies, whether by brute force or clever reductions, are scarce, at least apart from the work cited above and a few very special cases in which they can be expressed in closed form, analytically. The emphasis here is on direct computation when the tests are repeatable, conditionally independent given the classes and the cost of a tree is a linear combination of the average terminal entropy and the average depth. Computing T_{glo} is then sometimes feasible, although intensive, because the optimal test to perform at any interior node is determined by the depth of the node and the conditional distribution on classes at the node. In other words, the posterior distribution is a "sufficient statistic" in that it carries all the information in the previous tests which is relevant for deciding how to continue. Optimal trees can then be generated from dynamic programming and variants thereof.

The complexity of an exact computation depends on the number M of distinct tests (in distribution) and the maximum depth D of the tree. We focus on complexity

as a function of D for fixed, relatively modest values of M. In one variant, the complexity is of order D^{2M}, which is feasible, in contrast to M^{2^D}, which is the the total number of possible trees, i.e., the order of a brute force computation without exploiting the independence assumption. Some of these observations can be traced back to DeGroot's 1970 classic text [15], where *fixed-length* optimal trees are discussed under the above assumptions, although none are actually constructed, probably due to a lack of computing resources.

In the following two sections we review the stochastic framework for tree-structured classification and the standard construction by stepwise entropy reduction; we also introduce a cost functional which accounts for both mean depth and mean terminal entropy and describe a simple recursion that characterizes minimal cost trees. A special case in which the test results are determined by the class is considered briefly in Section 4. In Section 5, we specialize to the independent model. We present a simple characterization of the cost-minimizing testing strategy in terms of the posterior distribution as well as analyze the resulting complexity of global optimization; in fact, two algorithms are presented, one top-down and the other bottom-up, for computing minimal cost trees. Bounds on the information gain are given in Section 6 and in Section 7 examples are given which illustrate the superiority of global strategies in several cases. Finally, some concluding remarks are made in Section 8.

2 Tree-Structured Classification

The goal is to assign a class label from a finite set $\mathcal{Y} = \{a, b, c, ...\}$ to a "feature vector" $\xi = (\xi_1, \xi_2, ..., \xi_p)$. Classification is based on a finite tree graph $\mathcal{T}$. The terminal nodes (denoted $\partial\mathcal{T}$) are each labeled by a class. The internal nodes (denoted $\dot{\mathcal{T}}$) are each labeled by a "test" - a discrete function $X(\xi)$ of the feature vector. For simplicity we will use only *binary tests*, for example $X = I_{\{\xi_i > c\}}$, which is the standard form of the tests in CART [8] and other algorithms. We write $\mathbf{X} = \{X_1, X_2, ..., X_M\}$ for the pool

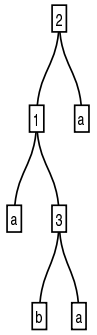


Figure 1: Example of a classification tree.

of available tests. The index of the test assigned to $t \in \dot{\mathcal{T}}$ is denoted $\pi(t) \in \{1, ..., M\}$, the depth of t by $d(t)$ (the depth of the root node is 0) and the set of observations preceding t by Q_t.

We regard the set of tests as random variables (relative to a background probability space) and we assume there is a true class $Y \in \mathcal{Y}$, another random variable. The class Y may or may not be determined by the tests or by the underlying feature vector. Let $\mathcal{M}$ denote the joint probability distribution of Y and $\mathbf{X}$; we will write $p_0(y), y \in \mathcal{Y}$, for the marginal (or "prior") distribution of Y and $g(\mathbf{x}|y)$ for the conditional distribution $P(\mathbf{X} = \mathbf{x}|Y = y), \mathbf{x} \in \{0, 1\}^M, y \in \mathcal{Y}$.

Let $T = T(\mathbf{X})$ denote the resulting random variable taking values in $\partial\mathcal{T}$; thus, Q_t is the event $\{T = t\}, t \in \partial\mathcal{T}$, and depends on the outcomes of the tests $X_{\pi(s)}$ at internal nodes s along the branch from the root to t. The classifier is denoted by $\hat{Y}_T = \hat{Y}_T(\mathbf{X})$ and takes values in $\mathcal{Y}$. The class assigned to $t \in \partial\mathcal{T}$ is always the *maximum a posteriori* estimator, i.e., the class y which maximizes $P(Y = y|T = t)$.

Figure 1 is an example of a classification tree with two classes a, b. As indicated, the test performed at the root is X_2. If $\{X_2 = 0\}$ is observed, test X_1 is performed; if $\{X_2 = 0\} \cap \{X_1 = 0\}$ is observed then class a is inferred; and so forth. The history of the terminal node $t \in \partial\mathcal{T}$ labeled b is $Q_t = \{X_2 = 0\} \cap \{X_1 = 1\} \cap \{X_3 = 0\}$.

3 Testing Strategies

For simplicity, we will write $P_t(.)$ for conditional probability $P(.|Q_t)$, and p_t for the posterior distribution of Y given Q_t: $p_t(y) = P(Y = y|Q_t)$. The conditional (Shannon) entropy of Y at node t is

$$H_t(Y) = H(Y|Q_t) = -\sum_y P_t(Y = y) \log_2 P_t(Y = y).$$

If $P(Q_t) = 0$, we set $H_t(Y) = 0$. The entropy at the root of $\mathcal{T}$ is $H(Y)$.

3.1 Local Optimization

If we perform test m at $t \in \dot{\mathcal{T}}$, the average class entropy given this test and the previous outcomes is

$$H_t(Y|X_m) = P_t(X_m = 0)H_{t_0}(Y) + P_t(X_m = 1)H_{t_1}(Y),$$

where t_0 and t_1 are the two descendents of t. (If t is the root node, we will write $H(Y|X_m)$ for $H_t(Y|X_m)$, and if $P(Q_{t_0}) = 0$ or $P(Q_{t_1}) = 0$, we set $H_t(Y|X_m) = H_t(Y)$.) The standard "one step ahead" testing strategy is

$$\pi(t) = \arg \min_{m=1,\ldots,M} H_t(Y|X_m). \tag{1}$$

It can also be characterized as choosing the test X_m which most reduces the *mean* Kullback-Liebler distance between the (random) conditional distributions $p_t(y|X_m)$ and $p_t(y|\mathbf{X})$. Together with a stopping rule, this is the recursive design for building T_{loc}.

3.2 Global Optimization

The aim of a global strategy is to build a tree classifier that balances error and computation. The former is measured by *average terminal entropy*

$$H(Y|T) = \sum_{t \in \partial \mathcal{T}} P(Q_t)H_t(Y)$$

and the later by *expected depth*:

$$Ed(T) = \sum_{t \in \partial \mathcal{T}} P(Q_t) d(t) = \sum_{s \in \dot{\mathcal{T}}} P(Q_s).$$

(The second equality results from writing $d(t) = \sum_{s \in \dot{\mathcal{T}}} I_{\{s<t\}}$, where $s < t$ indicates s precedes t in $\mathcal{T}$, and then interchanging the sums.) One could minimize entropy subject to a bound on expected depth, or vice-versa, but hard constraints are difficult to enforce. Instead we introduce a control parameter $\lambda > 0$ and define the *cost* of T as

$$C(T, \mathcal{M}) = H(Y|T) + \lambda Ed(T) \tag{2}$$

We write $C(T, \mathcal{M})$ to emphasize that the cost depends on the distribution of $(Y, \mathbf{X})$. One global optimization problem is then to minimize $C(T, \mathcal{M})$ over all T. Instead, we will minimize $C(T, \mathcal{M})$ subject to a maximum depth $D = \max_{t \in \partial \mathcal{T}} d(t)$.

The expected depth is of course the expected number of tests performed in order to reach a terminal node. Trees minimizing a cost function based instead on the maximal depth or the total number of tests would be very different. In fact, for a fixed error rate, reducing the expected depth necessitates increasing the allowable maximum depth. Notice also that tests with varying costs can easily be accommodated by replacing $Ed(T)$ by

$$\sum_{t \in \dot{\mathcal{T}}} c(\pi(t)) P(Q_t)$$

where $c(m)$ is the cost of test X_m. We will always assume $c(m) \equiv 1$.

3.3 A Recursion

Let $C^\star(\mathcal{M}, D)$ be the minimum value of $C(T, \mathcal{M})$ over all trees whose maximal depth is bounded by D. Consider a tree with test m at the root and let $\mathcal{T}_0$ and $\mathcal{T}_1$ be the left and right subtrees, respectively. Then clearly

$$\begin{aligned} C(T, \mathcal{M}) &= \lambda + P(X_m = 0) \left(H(Y|T_0, X_m = 0) + \lambda Ed(T_0) \right) \\ &+ P(X_m = 1) \left(H(Y|T_1, X_m = 1) + \lambda Ed(T_1) \right). \end{aligned}$$

It follows that $C^\star(\mathcal{M}, D)$ obeys the following recursion:

Proposition 1 *For $D = 0$,*

$$C^\star(\mathcal{M}, 0) = H(p_0).$$

For $D > 0$,

$$C^\star(\mathcal{M}, D) = \min \left\{ \begin{array}{l} H(p_0); \\ \\ \lambda + \min_{m \in \{1,\ldots,M\}} \left\{ \begin{array}{l} P(X_m = 0) C^\star(\mathcal{M}(.|X_m = 0), D-1) + \\ P(X_m = 1) C^\star(\mathcal{M}(.|X_m = 1), D-1) \end{array} \right. \end{array} \right. \tag{3}$$

The minimal cost $C^\star(\mathcal{M}, D)$ is positive and decreasing in D, and hence converges as $D \to \infty$ to the minimal cost of an unbounded tree. We approximate this cost by $C^\star(\mathcal{M}, D)$ for a large enough value of D. Direct evaluation of $C^\star(\mathcal{M}, D)$ is computationally prohibitive (except for small numbers of tests and small depths). However, if the tests are conditionally independent, then exact computation becomes feasible in non-trivial cases, as we shall see shortly.

4 Constrained Twenty Questions

Perhaps the simplest model is the one underlying the familiar parlor game of "twenty questions": The class is determined by the tests (in particular $M > \log_2 |\mathcal{Y}|$) and the tests are determined by the class (i.e., there is no randomness once Y is known). The model $\mathcal{M}$ is then determined by p_0 and the binary string of M test results for each class. Since doing all the tests determines Y, the natural problem is to find the testing strategy which asks the fewest number of questions on average in order to determine Y, i.e., the tree T_{glo} which minimizes $Ed(T)$ subject to $H(Y|T) = 0$.

Since Y determines $\mathbf{X}$, we have $H_t(Y|X_m) = H_t(Y, X_m) - H_t(X_m) = H_t(Y) - H_t(X_m)$. Hence (1) reduces to

$$\pi(t) = \arg \max_{m=1,\ldots,M} H_t(X_m),$$

which amounts to choosing the test at node t which divides the classes into two groups whose masses (measured by p_t) are as equal as possible.

If there is a test for *every* subset of classes ("complete tests"), then the best global strategy is the Huffman code for p_0 and

$$H(p_0) \leq Ed(T_{glo}) \leq Ed(T_{loc}) \leq H(p_0) + 1.$$

(We omit the proof of the last inequality.) However, the general problem of computing T_{glo} is NP complete [22]. Dynamic programming leads to an algorithm [17] which is exponential in either M or $|\mathcal{Y}|$, and is feasible for "small" values of these parameters. Garey and Graham [18] consider the case in which p_0 is uniform and compare the performance of greedy and optimal strategies over all possible families of tests, showing that the former can perform very poorly depending on this family.

5 Conditionally Independent, Repeatable Tests

In contrast to constrained twenty questions, suppose the tests are (conditionally) non-degenerate, an obviously more realistic case. However, in order to achieve computational feasibility, at least for "small" problems, we add the assumption of "repeatability" This provides a richer framework than constrained twenty questions in which to display the disparity in efficiency between local and global strategies.

Specifically, we suppose from here on that the tests are *conditionally independent* given Y:

$$g(\mathbf{x}|y) = \prod_{m=1}^{M} g_m(x_m|y),$$

where $g_m(x|y) = P(X_m = x|Y = y), x \in \{0,1\}, y \in \mathcal{Y}$. Suppose further that the tests are *infinitely repeatable* in the sense that there are many independent copies of each type of test. We shall continue to write $\mathbf{X} = \{X_1, ..., X_M\}$ for a *generic* set of distinctly-distributed tests. The full family of available tests is then $\{\mathbf{X}_1, \mathbf{X}_2, ...\}$, where $\mathbf{X}_j, j = 1, 2, ...$, are independent, identically distributed copies of $\mathbf{X}$. The model is then determined by $\{p_0, g_m\}$.

Remarks on Repeatability: i) This differs from constrained twenty questions in that the same test (in distribution) may now appear several times along the same branch of $\mathcal{T}$.

ii) This setting (conditional independence and repeatability) is precisely the one in [15]. More generally, it is at the intersection of sequential statistics [12], game theory [6] and adaptive control processes [5]. In these domains, optimal strategies can, in principle, be computed using dynamic programming; still, cases in which they can be expressed in simple analytic terms are uncommon and the emphasis is on asymptotic results (e.g., $Ed(T) \to \infty$) for greedy procedures. See also [13], [20] and [11], in which printed characters are classified with trees based on the assumption the image values are class-conditionally independent.

iii) This paper was motivated by experiments in pattern recognition (see the Note in §7). In most such applications, repeatability is not a realistic assumption, and nor is conditional independence for that matter, at least in strict terms. However, when the original feature vector is varied and high-dimensional (as in image processing), and the number of classes is small, it may often be the case that certain subsets of tests have *nearly* the same conditional distribution and are *nearly* conditionally independent.

5.1 Sufficiency of the Posterior

The key observation is that the evolution of the distribution of $(\mathbf{X}, Y)$ as tests are performed depends only on the evolution of the posterior distribution of Y. More specifically, if Q denotes a history of tests, then the posterior is $p(y|Q) = P(Y = y|Q)$ and

$$\begin{aligned} P(\mathbf{X} = \mathbf{x}, Y = y|Q) &= p(y|Q)P(\mathbf{X} = \mathbf{x}|Q, Y = y) \\ &= p(y|Q)g(\mathbf{x}|y) \end{aligned}$$

Here $\mathbf{X}$ represents a "fresh copy" of tests conditionally independent of those appearing in Q. It follows that, for the independent model, we can just as well index the minimal cost $C^\star$ by the posterior p_t as $\mathcal{M}$.

Updating the posterior based on a new test X_m is very simple:

$$\begin{aligned} p(y|Q, X_m = x) &= \frac{P(X_m = x|Q, Y = y)P(Y = y|Q)P(Q)}{\sum_{y' \in \mathcal{Y}} P(X_m = x|Q, Y = y')P(Y = y'|Q)P(Q)} \\ &= \frac{g_m(x|y)p(y|Q)}{\sum_{y' \in \mathcal{Y}} g_m(x|y')p(y'|Q)} \end{aligned}$$

In particular, at the children t_0 and t_1 of an internal node t, we obtain $p_{t_0}(y)$ and $p_{t_1}(y)$ from $p_t(y)$ by choosing $Q = Q_t, m = \pi(t)$ and $x = 0, 1$, respectively. In a similar manner, we see that

$$H(Y|Q_t, X_m) = \sum_{x=0,1} \sum_{y \in \mathcal{Y}} g_m(x|y)p_t(y)H(Y|Q_t, X_m = x)) \tag{4}$$

where $p(y|Q_t, X_m), y \in \mathcal{Y}$ can be expressed in terms of g_m and p_t as above.

The consequence for the local strategy (1) is that *computing* $H(Y|Q_t, X_m)$ *under the model* $\{p_0, g_m\}$ *is the same as computing* $H(Y|X_m)$ *under the model* $\{p_t, g_m\}$. One implication of this was pointed out in [15]: If there is a dominating test X_{m^*} in the sense that $H(Y|X_{m^*}) \leq \min_m H(Y|X_m)$ *under any prior* p_0, then only this test would appear in both T_{loc} and T_{glo}. Needless to say, such tests never exist in practice.

Turning to global strategies, the test assignment π^* of the optimal tree now has a very simple characterization. Let $\mathcal{P}_0 = \{p_0\}$ and, for $k > 0$, let $\mathcal{P}_k$ denote the set of *all possible posterior distributions after* k *tests*, i.e., all possible distributions $p(.|Q)$ where Q is a conjunction of k test results. In particular, $p_t \in \mathcal{P}_{d(t)}$. Then depending on λ and the model $\{p_0, g_m\}$, there is a sequence of functions

$$\Psi_k : \mathcal{P}_k \to \{1, 2, ..., M\}, \ 0 \leq k \leq D - 1$$

which gives the optimal test at depth k as a function of the posterior after k tests. Here again D is the maximum allowable depth. In other words, at any internal node t of the optimal tree:

$$\pi^*(t) = \Psi_{d(t)}(p_t) \tag{5}$$

Consequently, due to conditional independence, the complexity of computing a global strategy reduces to counting posteriors, which, as we shall see in the following sections, is further simplified by the assumption of repeatability.

5.2 Computational Complexity

If the number of tests M and the maximum depth D are small enough we can compute $C^\star(p_0, D)$ and the corresponding tree $\mathcal{T}_{glo}$ very efficiently. The interest of this cost analysis is that within these constraints one can display comparisons between exact and virtually exact minimal cost trees T_{glo} and the corresponding greedy trees T_{loc} and thereby asses the performance loss as well as the feasibility of alternatives to stepwise entropy reduction.

The important computational issue is the growth of $\mathcal{P}_k$ as k increases and how finely we quantize it if we forgo an exact computation (as in Example 3 in the following section). For example, in the simplest case of just two classes $\{a, b\}$, suppose we quantize $p(a|Q) \in [0, 1]$ into L levels; obviously $p(b|Q) = 1 - p(a|Q)$. The complexity of using dynamic programming in order to compute the minimal cost tree (under the approximation resulting from this quantization) is then only $O(MLD)$. *However*, this *a priori* quantization induces errors and a better approximation to T_{glo} is discussed in §5.2.2.

We can compute the complexity of an exact recursion. In order to determine $C^\star(p_0, D)$ we need $C^\star(p(.|X_m = x), D - 1)$ for $x = 0, 1$ and $m = 1, ..., M$. In other words, we need $C^\star(p, D - 1)$ for

$$p \in \mathcal{P}_1 = \{p(.|X_i = x), 1 \leq i \leq M, x \in \{0, 1\}\}$$

which in turn requires $C^\star(p, D - 2)$ for

$$p \in \mathcal{P}_2 = \{p(.|X_i = x_i, X_j = x_j), 1 \leq i, j \leq M, x_i, x_j \in \{0, 1\}\},$$

and so forth. Hence we need to compute the size of each $\mathcal{P}_k$. This is of course also evident from a backwards induction argument based on the characterization of

Tree Depth	0	1	2	3	4	5	6	7	8	9	10	total
No. Posteriors	1	4	10	20	35	56	84	120	165	220	286	1001

Figure 2: Possible posteriors after k tests, $k = 0 \ldots 10$.

the optimal strategy given by (5); see §5.2.1 below. If k tests are performed, the posterior obviously depends only on the number of events of each "type" (m, x), where $m = 1, 2, ..., M$ and $x \in \{0, 1\}$. The order in which these events occur along the branch is irrelevant. Let η_j be the number of events of type $j = 1, 2, ..., 2M$ relative to some ordering of the $2M$ pairs (m, x). Then of course $0 \leq \eta_j \leq 2M$ and $\sum_j \eta_j = k$. We want the number of distinct sequences $(\eta_1, ..., \eta_{2M})$. But there is a 1-1 correspondence between these and sequences $\alpha_j = \eta_1 + \cdots + \eta_j + j$ for $j = 1, 2, ..., 2M - 1$. Since

$$1 \leq \alpha_1 < \alpha_2 < \cdots < \alpha_{2M-1} \leq k + 2M - 1,$$

we have

$$|\mathcal{P}_k| = \binom{k + 2M - 1}{2M - 1}.$$

Values for $M = 2$ and $k = 0, \ldots, 10$ are given in Figure 2. Notice that $|\mathcal{P}_k|$ grows slowly with k compared with $(2M)^k$, which is the number of possible situations after k tests. This simple argument allows us to compute optimal trees in reasonable time for $10 \leq D \leq 20$ and $2 \leq M \leq 4$.

The computation of T_{glo} can be organized either iteratively and "bottom-up" using standard dynamic programming or recursively and "top-down" using (3). The latter is slower in simple cases but has the advantage that it can be easily modified to yield an *approximation* of the optimal tree when the number of tests gets relatively large. This approximation is different from, and superior to, the one mentioned earlier based on *a priori* quantization of the posterior.

5.2.1 Bottom-up Computation

We start at the terminal nodes and work up:

- Step 0: For each $p \in \mathcal{P}_D$, compute and store $C^\star(p, 0) = H(p)$
- Step 1: For each $p \in \mathcal{P}_{D-1}$, compute and store $C^\star(p, 1)$ using equation (3) and the values stored in Step 0.
- Step $D-1$: For each $p \in \mathcal{P}_1$, compute and store $C^\star(p, D-1)$ using equation (3) and previously stored values.
- Step D: Compute $C^\star(p_0, D)$ using equation (3) and previously stored values.

Hence, the algorithm amounts to filling in a table with $D+1$ rows corresponding to different depths. The entries in row k are a variable length set of vectors - all the distributions in $\mathcal{P}_k$ - and the minimal costs; the first row has only p_0. The first row filled is row D; it has $|\mathcal{P}_D|$ entries. Then row $D-1$ is filled using the entries in row D; it has $|\mathcal{P}_{D-1}|$ entries, and so on. After the table is made it is a simple matter to generate the functions $\{\Psi_k\}$ and hence T_{glo} itself (equivalently, the optimal testing strategy π) by a top-down pass collecting the minimizing tests at each level.

Since at each step there is a loop over posteriors and possible tests, the total complexity as a function of D and M is proportional to

$$M \sum_{k=0}^{D} |\mathcal{P}_k| = M \sum_{k=0}^{D} \binom{k+2M-1}{k} = M \binom{D+2M}{D} \tag{6}$$

The first equality was derived in the previous section and the second one can be found for example in [23], p. 54. Consequently, the complexity in D is bounded by D^{2M}. Notice that this bound is *independent* of the number of classes.

In case of $M = 2$ and $D = 10$ the effective computing time on a 225 Mhz PC is one-tenth of a second. Figure 3 shows the value in (6), in thousands, when the number of tests is $M = 2, 3, 4$ and the maximal depth is $D = 10, 20, 30$.

	2	3	4
10	2	24	175
20	21	691	12,432
30	93	5,843	195,613

Figure 3: Value of équation (6), in thousands, for 2, 3 or 4 test types and maximum depth 10, 20 or 30.

5.2.2 Top-down Computation

The computation can also be organized recursively, but top-down. The algorithm still involves completing the table mentioned above, but the computations are performed in a different order corresponding to a depth-first examination of the M-ary tree associated with (3). Thus the core of the program is a recursive procedure that computes $C^\star(p,k)$. Start with $k = D$ and $p = p_0$; if this value is in the table return it. If not, go to (3) and look for $C^\star(p,k)$ for $k = D-1$ and $p = p(.|X_m = x)$ for $x = 0$ and $m = 1$; p is computed from $\{p_0, g_m\}$ as indicated above. If this entry is not in the table, call the same procedure again for $k = D-2$, each time computing the new posterior and checking to see if it is in the table. At the beginning the procedure is called D times until we simply compute $C^\star(p,0) = H(p)$ for posterior corresponding to the event $Q = \{X_{11} = 0, X_{12} = 0, ..., X_{1D} = 0\}$ where $X_{1j}, 1 \leq j \leq D$ are independent copies of X_1. The main program is a call to this procedure with the parameters $p = p_0$ and $k = D$.

Although this implementation is more demanding than dynamic programming, the amount of computation is much less than it appears. Very quickly most, and then all, of the entries needed to compute $C^\star(p,k)$ are found in the table. Moreover, the recursive method can be easily modified to *approximate* an optimal tree as follows: Instead of looking for an exact match for the posterior, check if the optimal cost has been already computed at the given depth for a distribution *sufficiently close* to the

desired posterior. This provides a much better approximation to the optimal tree than *a priori* quantization of the posterior, which is problematic as the number of classes increases.

6 Bounds on the Information Gain

In this section we consider maximum possible gains and minimum possible costs due to a set of tests. Let $t \in \dot{\mathcal{T}}$. The information gain at node t is $H_t(Y) - H_t(Y|X_{\pi(t)})$, and the information gain due to T is $H(Y) - H(Y|T)$. Note that since the tree $\mathcal{T}$ already provides a binary coding of the values of T, and since the mean code length of a random variable is always larger than its entropy, one always has

$$H(Y) - H(Y|T) \leq H(Y,T) - H(Y|T) = H(T) \leq Ed(T).$$

Proposition 3 provides a better bound in the case of conditionally independent, repeatable tests. It is based on the following identity, the proof of which follows easily by induction on the number of leaves.

Proposition 2

$$H(Y) - H(Y|T) = \sum_{t \in \dot{\mathcal{T}}} P(Q_t)(H_t(Y) - H_t(Y|X_{\pi(t)})) \tag{7}$$

In the case of conditionally independent tests, there is a simple, tractable bound on the information gain of any T. For each $m = 1, ..., M$, define the "channel capacity" [14]

$$c(X_m, Y) = \max_{p_0}[H(Y) - H(Y|X_m)].$$

The maximum is over all possible distributions for Y. Let t be an internal node of $\mathcal{T}$; the information gain $H_t(Y) - H_t(Y|X_{\pi(t)})$ is determined by p_t and $\{g_{\pi(t)}\}$. Hence the information gain at t is bounded by $c(X_{\pi(t)}, Y)$. Substituting this bound into (7)

and using the characterization in §3 of expected depth as a sum over internal nodes, we arrive at the following bound on the total information gain, which may also be interpreted as a coupled constraint on $H(Y|T)$ and $Ed(T)$:

Proposition 3 *For any tree T and any model $\{p_0, g_m\}$,*

$$H(Y) - H(Y|T) \leq Ed(T) \max_{m \in \{1,\ldots,M\}} c(X_m, Y) \tag{8}$$

Since $c(X_m, Y) \leq 1$, this bound is better than the general one given earlier.

7 Experiments

We now give several examples to illustrate the difference in performance between T_{loc} using the recursion (1) and T_{glo} using the cost functional (2). The behavior we exhibit remains the same if Shannon entropy is replaced by another "purity measure"; indeed, changing the splitting criterion does not seem to have a great effect on performance in general ([8],[10]). Moreover, although the examples are based on the independent model of Section 5, we believe the disparity observed might be even greater with a non-trivial, conditional dependency structure among the tests. However, constructing globally optimal trees for general models is not practical.

Example 1: The performance of classification trees made using (1) may degrade considerably if $\max_{y \in \mathcal{Y}} p_0(y)$ is near one. Here is a toy example in which the greedy strategy selects the "wrong" tests at small depths, resulting in an expected depth 1.6 times larger than T_{glo} in order to achieve the same error rate or final entropy.

There are two classes with $p_0(a) = 10^{-4}$ and $p_0(b) = 1 - 10^{-4}$ and two tests with

$$g_1(1|a) = 1 \text{ and } g_1(1|b) = 0.5$$

$$g_2(1|a) = 0.5 \text{ and } g_2(1|b) = 0.$$

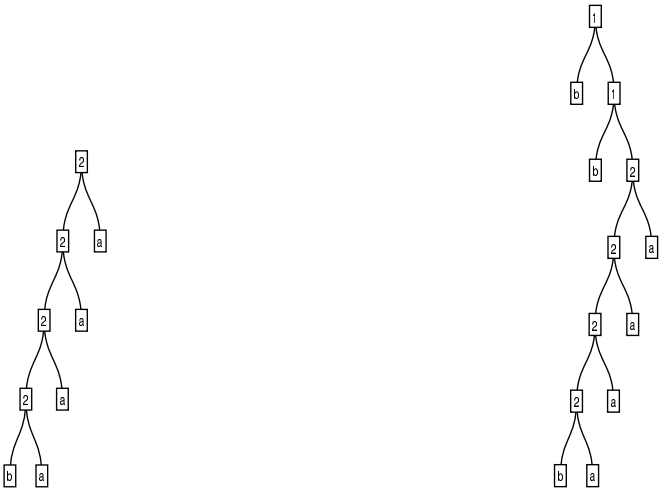


Figure 4: Left: Locally optimal tree. Right: Globally optimal tree. The error rates are the same but the *mean* depth of the global tree is smaller.

In other words, X_1 always answers "yes" on the rare class and answers randomly on the common class, and vice-versa for X_2.

Note: This example was motivated by experiments with learning algorithms for visual selection [3]; the rare class corresponds to an "object" being present at a fixed location in a large scene and the common class to "background." The first test has false negative error zero (i.e., "loses" no objects) but has false positive error 0.5, and vice-versa for X_2. Given such tests are available (and of equal cost) and given a dynamic testing strategy, how does one minimize computation subject to an error constraint?

The cost function for the globally optimal tree is (2) with maximum depth $D = 6$ and $\lambda = 10^{-4}$. The tree which minimizes cost is displayed in Figure 4; it was computed using the exact top-down recursion discussed in Section 5.2.2. The terminal nodes are labeled according to the mode of the posterior distribution. The error rate is 0 when

$Y = b$ and is $\frac{1}{16}$ when $Y = a$, concentrated in the deep node labeled b which is reached with probability approximately $\frac{1}{4}$. The mean depth is small because the probability of reaching the depth one (resp. depth two) terminal is nearly $\frac{1}{2}$ (resp. $\frac{1}{4}$), resulting in $Ed(T_{glo}) \approx \frac{5}{2}$. (The righthand side of (8) is $2.5 \times .32$ which is much larger than the actual information gain because the starting entropy is small: $H(Y) = 0.0015$.)

The locally optimal strategy always prefers test X_2 because

$$H(Y|X_1) \approx H(Y) \text{ and } H(Y|X_2) \approx \frac{1}{2}H(Y).$$

(In contrast, the global strategy puts X_1 at the top even though it provides much less average information about Y.) The depth is determined by matching the error rate of T_{glo} and the resulting tree is shown in Figure 4. The probability of exiting at the deepest terminal nodes is nearly one, which makes $Ed(T_{loc}) \approx 4$.

Example 2: Consider now a less extreme example, still with two classes and two tests. The prior is $p_0(a) = p_0(b) = 0.5$ and

$$g_1(1|a) = 0.9 \text{ and } g_1(1|b) = 0.4$$

$$g_2(1|a) = 0.6 \text{ and } g_2(1|b) = 0.1.$$

The maximum depth for T_{glo} is $D = 30$ and the tree is constructed the same way as in Example 1.

The performance of T_{loc} and T_{glo} in several cases is given in Figure 5. We adjusted the parameter λ to make either $H(Y|T_{glo}) \approx H(Y|T_{loc})$ or $Ed(T_{glo}) \approx Ed(T_{loc})$. Recall, we estimate Y by the most likely class at the leaves, denoted $\hat{Y}_T$. In this case $H(Y) = 1$ and $\max\{c(X_1, Y), c(X_2, Y)\} = 0.21$, which leads to the constraint

$$1 \leq H(Y|T) + 0.21Ed(T).$$

This is consistent with the values in Figure 5.

Again, there is generally a significant difference in performance between T_{loc} and T_{glo}, as well as in the shape of the trees; for instance, T_{glo} is very unbalanced relative

	$P(\hat{Y}_T \neq Y)$	$H(Y\|T)$	$Ed(T)$
T_{loc}	0.014	0.083	10.2
T_{glo}	0.010	0.080	6.6
T_{glo}	0.001	0.012	10.2
T_{loc}	0.051	0.237	5.6
T_{glo}	0.021	0.147	5.4
T_{glo}	0.038	0.228	4.5

Figure 5: Comparing performance of local and global strategies for the model in Example 2 with maximal depth 30.

Class y	a	b	c	d	e	f
$p_0(y)$	0.5	0.1	0.1	0.1	0.1	0.1
$g_1(1\|y)$	0.9	0.1	0.9	0.1	0.1	0.1
$g_2(1\|y)$	0.9	0.1	0.1	0.9	0.1	0.1
$g_3(1\|y)$	0.1	0.9	0.1	0.1	0.9	0.1
$g_4(1\|y)$	0.1	0.9	0.1	0.1	0.1	0.9

Figure 6: The model in Example 3. There are 6 classes and 4 types of tests.

to T_{loc}. It seems that the expected depth with balanced priors needs to be larger than with skewed priors in order to see a very sharp difference. For example, see Figure 5 in the case $Ed(T_{loc}) = Ed(T_{glo}) = 10.2$.

Example 3: Examples with more classes and more tests show the same qualitative behavior. We use the approximation procedure outlined in Section 5.5.2. in order to compute T_{glo} with six classes and four tests; the model is presented in Figure 6 and the results in Figure 7.

	$P(\hat{Y}_T \neq Y)$	$H(Y\|T)$	$Ed(T)$
T_{loc}	0.037	0.190	6.6
T_{glo}	0.023	0.180	5.4
T_{glo}	0.010	0.087	6.5

Figure 7: Comparing performance of local and global strategies for the model in Example 3.

8 Discussion

Classification trees are a popular method for addressing problems arising in non-parametric estimation, especially in domains such as pattern recognition ([4],[19], [21], [28]) in which the data are often high dimensional. Artificial neural networks are more popular, but tree-structured decision-making is easier to interpret; another advantage is the natural way in which "feature selection" is performed during tree construction [9]. As a result, there is a continuing interest in improving methods for constructing tree classifiers, especially in the data-driven case in which trees are "induced" from samples in a training set $\mathcal{L}$, i.e., test statistics and conditional entropies are estimated from $\mathcal{L}$. For example, from time to time new purity measures, splitting rules and pruning recipes are proposed and existing ones are compared ([8],[10], [24],[27]). And recently the dramatic gains from using *multiple trees* have been documented and analyzed from the point of view of randomization, negative correlation and the bias/variance decomposition ([1],[2],[7],[16], [29]).

We have analyzed the limitations of the basic induction method itself, at least in cases in which the greedy designs are likely to lead to very inefficient trees when measured by global criteria such as mean path length. Such cases arise when some classes are very rare and when the training set $\mathcal{L}$ is small; the benefits of choosing good tests are then accentuated since the amount of data available at a node for estimating information gains and class likelihoods is rapidly decreasing with depth of

the node. Indeed, we would argue that the interesting limit in pattern recognition and other applications is $|\mathcal{L}| \to 0$ rather than $|\mathcal{L}| \to \infty$, and that the most effective way to introduce problem-specific knowledge into the design of the classifier is by "hard-wiring" global constraints.

Finally, how might global optimization be relevant for inducing trees either from data or from more complex models, especially in applications to pattern recognition and machine learning where assumptions such as independence and repeatability are usually violated? The natural path would appear to be $\mathcal{L} \to \mathcal{M} \to T_{glo}$: First estimate a model from the data and then calculate an efficient tree from the model. But unless $\mathcal{M}$ is severely restricted *a priori*, it will not be sufficiently elementary to deduce T_{glo}. Yet it is precisely the rich dependency structure in the feature vector which makes the underlying classification problem interesting and challenging. Perhaps globally optimal strategies which are derived from simplified, approximate models (for instance assuming conditional independence but using the actual marginal test statistics) might serve as "blueprints" for recursive tree construction. Given the disparities we have illustrated, the rewards could be significant.

References

[1] Y. Amit, G. Blanchard, and K. Wilder. Multiple randomized classifiers: Mrcl. Technical Report 496, Department of Statistics, University of Chicago, 1999.

[2] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

[3] Y. Amit, D. Geman, and B. Jedynak. Efficient focusing and face detection. In H. Wechsler and J. Phillips, editors, *Face Recognition: From Theory to Applications, NATO ASI Series F*. Springer-Verlag, Berlin, 1998.

[4] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Trans. PAMI*, 19(11), 1997.

[5] R. Bellman. *Adaptive Conrol Process: A Guided Tour.* Princeton University Press, 1961.

[6] D. Blackwell and M. A. Girschick. *Theory of Games and Statistical Decisions.* John Wiley, 1954.

[7] L. Breiman. Arcing classifiers. *Annals of Statistics*, 26:801–878, 1998.

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, CA., 1984.

[9] D. Brown, V. Corruble, and C. L. Pittard. A comparison of decision tree classifiers with backpropagation neural networks for multimodal classification problems. *Pattern Recognition*, 26:953–961, 1993.

[10] W. Buntine and T. Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8:75–85, 1992.

[11] R. G. Casey and G. Nagy. Decision tree design using a probabilistic model. *IEEE Trans. Information Theory*, 30:93–99, 1984.

[12] H. Chernoff. *Sequential Analysis and Optimal Design.* SIAM, 1972.

[13] Y. S. Chow, H. Robbins, and D. Siegmund. *Great Expectations: The Theory of Optimal Stopping.* Houghton Mifflin, 1971.

[14] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley, New York, 1991.

[15] M. H. DeGroot. *Optimal Statistical Decisions.* McGraw-Hill, New York, 1970.

[16] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *J. Artificial Intell. Res.*, 2:263–286, 1995.

[17] M. R. Garey. Optimal binary identification procedures. *SIAM J. Appl. Math.*, 23:173–186, 1972.

[18] M. R. Garey and R. L. Graham. Performance bounds on the splitting algorithm for binary testing. *Acta Informatica*, 3:1974, 1974.

[19] S. B. Gelfand and E. J. Delp. On tree structured classifiers. In I. K. Sethi and A. K. Jain, editors, *Artificial Neural Networks and Statistical Pattern Recognition*, pages 51–70. North Holland, Amsterdam, 1991.

[20] J. C. Gittins. *Multi-armed Bandit Allocation Indices.* John Wiley, 1989.

[21] J. Huang, S. Gutta, and H. Wechsler. Detection of human faces using decision trees. In *Proceedings, Second International Conference on Automatic Face and Gesture Recognition*, pages 248–252. IEEE Computer Society Press, 1996.

[22] L. Hyafil and R. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5:15–17, 1976.

[23] D. E. Knuth. *Fundamental Algorithms.* Addison-Wesley, 1973.

[24] M. Miyakawa. Criteria for selecting a variable in the construction of efficient decision trees. *IEEE Trans. Computers*, 38:130–141, 1989.

[25] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[26] J. R. Quinlan and R. L. Rivest. Inferring decision trees using minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[27] I. K. Sethi. Decision tree performance enhancement using an artificial neural network implementation. In I. K. Sethi and A. K. Jain, editors, *Artificial Neural Networks and Statistical Pattern Recognition.* North Holland, Amsterdam, 1991.

[28] L. Spirkovska. Three-dimensional object recognition using similar triangles and decision trees. *Pattern Recognition*, 26:727–732, 1993.

[29] K. Wilder. *Decision tree algorithms for handwritten digit recognition*. PhD thesis, University of Massachusetts, Amherst, Massachusetts, 1998.

Chapter 8

Maximum likelihood set for estimating a point mass function

LETTER Communicated by Liam Paninski

Maximum Likelihood Set for Estimating a Probability Mass Function

Bruno M. Jedynak
bruno.jedynak@jhu.edu
Département de Mathématiques,
Université des Sciences et Technologies de Lille, France,
and Department of Applied Mathematics,
and Center for Imaging Science,
Johns Hopkins University, Baltimore, MD 21218, U.S.A.

Sanjeev Khudanpur
khudanpur@jhu.edu
Department of Electrical and Computer Engineering,
Johns Hopkins University, Baltimore, MD 21218, U.S.A.

We propose a new method for estimating the probability mass function (pmf) of a discrete and finite random variable from a small sample. We focus on the observed counts—the number of times each value appears in the sample—and define the maximum likelihood set (MLS) as the set of pmfs that put more mass on the observed counts than on any other set of counts possible for the same sample size. We characterize the MLS in detail in this article. We show that the MLS is a diamond-shaped subset of the probability simplex $[0, 1]^k$ bounded by at most $k \times (k-1)$ hyperplanes, where k is the number of possible values of the random variable. The MLS always contains the empirical distribution, as well as a family of Bayesian estimators based on a Dirichlet prior, particularly the well-known Laplace estimator. We propose to select from the MLS the pmf that is closest to a fixed pmf that encodes prior knowledge. When using Kullback-Leibler distance for this selection, the optimization problem comprises finding the minimum of a convex function over a domain defined by linear inequalities, for which standard numerical procedures are available. We apply this estimate to language modeling using Zipf's law to encode prior knowledge and show that this method permits obtaining state-of-the-art results while being conceptually simpler than most competing methods.

1 Introduction

Let p be a probability mass function (pmf) over a set $\{1, \ldots, k\}$ of finite cardinality. This may represent a set of numerical values for a quantitative

variable or a set of indices for a qualitative variable. The latter situation is often qualified as nonmetric, as will be the case in section 4, where the indices will refer to words in the English vocabulary.

Suppose that we observe n samples $x_1, \ldots, x_n$, that are independent and identically distributed (i.i.d.) with common pmf p, which is unknown and needs to be estimated from the observed samples. Prior information may be available about p and, in particular, a specific estimate, or an estimate of a certain form, may be preferred when $n = 0$.

For the case when $n \gg k$, a very satisfactory answer is the empirical distribution or type $\hat{p}$, namely:

$$\hat{p}(X = i) = \hat{p}_i = \frac{1}{n} \sum_{t=1}^{n} \mathbf{1}(x_t = i) \equiv \frac{n_i}{n}, \qquad i \in \{1, \ldots, k\}, \tag{1.1}$$

where $\mathbf{1}(\cdot)$ is an indicator function and, hence, n_i is the number of times the value i is observed in the sample.

When n is small, the pioneering work of Laplace (for $k = 2$) has led to the well-known Bayesian estimates as alternatives to the type. During World War II, while working on cracking German cryptographic systems, Jack Good and Alan Turing invented a method for regularizing the type (Good, 1953; Orlitsky, Santhanam, & Zhang, 2003). In their case, $k = 26$ was the number of letters in the Latin alphabet, and $n \approx 100 - 1000$. In section 4, we consider a case where k is the number of words in the English vocabulary, which is set to about 10^5, and the training sample is $n \approx 10^6$ words. Many smoothing techniques, most being variations on the Good-Turing idea, have been compared for such a case by Chen and Goodman (1996) and Chen and Rosenfeld (1999). Excellent empirical performance is obtained by using Good-Turing–like estimators. With the exception of the Bayesian estimates, however, there is often only a heuristic justification and no principled derivation of the estimation formulae.

There have, of course, been numerous studies of the pmf estimation problem since Laplace, and it is not our intention to present a comprehensive survey of the literature here, which begins at least as far back as Lidstone (1920) and continues to be an active area of investigation (Ristad, 1995; Poschel, Ebeling, Froemmel, & Ramirez, 2003).

We propose the following new method for estimating p. We consider the counts—the number of times each value appears—and define the maximum likelihood set (MLS) as the set of probability mass functions that put more mass on the observed counts than on any other set of counts possible for the given n. In a second step, an element is chosen from this set. It can be the one with maximum entropy or another based on available prior information. This view of the problem, we believe, is very natural—indeed, so much so that when we first arrived at this view, we expected that someone had already investigated it. We have not found any evidence of this in the literature.

1.1 The Empirical Distribution. The empirical distribution, or type, of a sample $x_1, \ldots, x_n$, as briefly mentioned earlier, is

$$\hat{p} = \left(\frac{n_1}{n}, \ldots, \frac{n_k}{n}\right), \text{ with } n = \sum_{i=1}^{k} n_i \,, \tag{1.2}$$

where $n_i, 1 \leq i \leq k$, are the counts, that is, the number of times the value i appeared in the sample. We write $\mathcal{P}^k$ the set of pmfs over a set of cardinality k and $\mathcal{P}_n^k$ the set of types with denominator n over a set of cardinality k. The probability, under $p \in \mathcal{P}^k$, of observing $x_1, \ldots, x_n$ is

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{k} p_i^{n_i} \,, \tag{1.3}$$

where n_i are the counts as above. The right-hand side of equation 1.3, viewed as a function of the pmf p, is called the likelihood function and may be rewritten as

$$\prod_{i=1}^{k} p_i^{n_i} = 2^{-n(D(\hat{p},p)+H(\hat{p}))} \,, \tag{1.4}$$

where

$$D(p,q) = \sum_{i=1}^{k} p_i \log_2 \frac{p_i}{q_i}, \tag{1.5}$$

with $0 \log_2 \frac{0}{q} = 0$ and $p \log_2 \frac{p}{0} = \infty$ for $p > 0$, is the Kullback-Leibler distance of p from q, and

$$H(p) = -\sum_{i=1}^{k} p_i \log_2 p_i, \tag{1.6}$$

with $0 \log_2 0 = 0$, is the Shannon entropy of p.

It is clear from equation 1.4 that the type $\hat{p}$ is a sufficient statistic for estimating p. Also note that $\hat{p}$ is the maximum likelihood estimate (MLE) of p, that is, the choice of p for which the likelihood equation 1.3 of $x_1, \ldots, x_n$, is maximum. Indeed, $D(\hat{p}, p) \geq 0$, with equality iff $p = \hat{p}$ (cf, e.g., Cover & Thomas, 1991).

For k fixed and $n \to \infty$, the type is a strongly consistent and efficient estimate of the pmf. However, the type may not be the best possible estimate for finite n. For example, one may have prior information about the true distribution that is captured in the type only for very large n. There is also

a more structural objection: when k is large, there might be many values $1 \le i \le k$, for which $p_i \ll \frac{1}{n}$. In this case, with high probability, we will observe $n_i = 0$. Hence, low-probability events tend to be underestimated and high-probability events overestimated by $\hat{p}$. One manifestation of this effect is that the expected entropy of the type underestimates the entropy of the original pmf. Indeed,

$$E\left[H(\hat{p})\right] = -E\left[\sum_{i=1}^{k} \hat{p}_i \log \frac{\hat{p}_i}{p_i} p_i\right] = -E\left[D(\hat{p}, p)\right] + H(p) \le H(p).$$

In section 2, we therefore construct a set of pmfs that contains the type as well as other pmfs that are close to it. In particular, it contains pmfs with larger entropy than the type. We will then choose an estimate from this set based on available prior knowledge.

1.2 Bayesian Estimates. Bayesian analysis offers an alternative to MLE. The Dirichlet family, indexed by a parameter β, is a family of prior distributions over pmfs given by

$$\pi_\beta(p) = \frac{1}{Z(\beta)} \prod_{i=1}^{k} p_i^{\beta-1}, \qquad p \in \mathcal{P}^k, \quad \beta \in \mathbb{R}, \tag{1.7}$$

where $Z(\beta)$ is a normalizing constant. Note that for $\beta = 1$, equation 1.7 reduces to the uniform distribution over $\mathcal{P}^k$. Now, if the Bayesian cost function is quadratic, that is,

$$L(p, q) = \sum_{i=1}^{k} (p_i - q_i)^2, \tag{1.8}$$

then the Bayesian estimate corresponding to the Dirichlet prior is the posterior expectation of p given $x_1, \ldots, x_n$, which can be shown to be

$$\hat{p}_\beta(i) = \frac{n_i + \beta}{n + \beta k}, \qquad \forall\, 1 \le i \le k. \tag{1.9}$$

This is often referred to as an add-β rule. The special case of $\beta \to 0$ yields the MLE $\hat{p}$, and $\beta = 1$—the so-called Laplace rule (cf. e.g. Lidstone, 1920). Estimators with $\beta = 0.5$ and $\beta = \frac{1}{k}$ have also been considered (see Nemenman, Shafee, & Bialek, 2002). Note that all such estimators with $\beta > 0$ assign a strictly positive mass to every value in $\{1, \ldots, k\}$, and they all converge to the type as $n \to \infty$.

We will see that the set from which we will choose our estimate contains all add-β rules in equation 1.9 for $0 \le \beta \le 1$.

1.3 Minimax Estimates. An alternative to Bayesian analysis is minimax analysis where one seeks an estimate that would be optimal in the worst case over the underlying model and in average over the observations. More precisely, if p is the underlying model and q an estimate of p, one builds the functional

$$R(q) = \sup_{p=(p_1,\ldots,p_k)} \sum_{n_1,\ldots,n_k;\sum_{i=1}^k n_i=n} \frac{n!}{n_1!\ldots n_k!} p_1^{n_1} \ldots p_k^{n_k} L(p,q). \tag{1.10}$$

For the quadratic cost, equation 1.8, as well as for the standardized quadratic cost,

$$L(p,q) = \sum_{i=1}^{k} \frac{(p_i - q_i)^2}{p_i}, \tag{1.11}$$

the minimum of $R(q)$ is achieved by an add-β rule, with $\beta = k^{-1}\sqrt{n}$ (Steinhaus, 1957) and $\beta = 0$ (Olkin & Sobel, 1979) respectively.

1.4 Maximum Entropy Estimates. Maximum entropy estimation is another standard solution to data sparseness. Instead of estimating $\hat{p}$, the maximum entropy method first estimates $\hat{p}(A_j) = \hat{a}_j$ for select sets $A_j \subset \{1,\ldots,k\}$, for which we have sufficient evidence in the n samples. Fixing the probability of some subsets of $\{1,\ldots,k\}$ in this manner typically underspecifies the pmf of interest, leading to a set $\mathcal{M}$ of admissible pmfs,

$$\mathcal{M} = \left\{ p \in \mathcal{P}^k \;:\; p(A_j) = \hat{a}_j,\;\; j = 1,\ldots,J \right\}, \tag{1.12}$$

in which the estimate $\hat{p}$ is but one member. From this admissible set, the pmf with the highest Shannon entropy is then chosen as the estimate of p. It is well known (see, Berger, Della Pietra, & Della Pietra, 1996) that the pmf with the maximum entropy has an exponential form:

$$\hat{p}_{\mathrm{ME}}(i) = \frac{1}{Z(\Lambda)} \exp\left\{ \sum_{j=1}^{J} \lambda_j \mathbf{1}(i \in A_j) \right\}, \qquad \forall\, 1 \le i \le k, \tag{1.13}$$

where the parameters $\Lambda = (\lambda_1,\ldots,\lambda_J)$ are chosen to satisfy the constraints of equation 1.12.

It can be shown that for every i, as long as at least one $p \in \mathcal{M}$ satisfies $p_i > 0$, it follows that $\hat{p}_{\mathrm{ME}}(i) > 0$. Thus, the maximum entropy estimate is inherently smooth.

There are several heuristics but few principles for selecting the sets A_j or even J. In language modeling, some A_j's are typically singleton, specifying, for instance, the probability of words that have been seen sufficiently

often in the sample; some A_j's may contain all words that can take on a certain grammatical part of speech (e.g., adjectives), and some A_j's may overlap with others, for example. Therefore, while maximum entropy estimation eliminates the need for some of the ad hoc assumptions made by other techniques, it leaves open the problem of selecting the sets used to define $\mathcal{M}$.

Another weakness of the classical maximum entropy method, as others have pointed out, is that the specification of $\mathcal{M}$ via equality constraints leads to an ad hoc choice for any candidate A_j: one must either constrain its probability to be exactly $\hat{a}_j$ or leave it completely unconstrained. This is unsatisfactory. For instance, if one were considering as candidate sets A_j all singleton sets, then the naive act of including all of them in the definition of $\mathcal{M}$ leads to $\mathcal{M} = \{\hat{p}\}$. On the other hand, leaving out all i for which, say, $n_i = 1$ from the definition of $\mathcal{M}$ may result in an estimate under which $n_i > 0$ and $n_{i'} = 0$, but $\hat{p}_{\mathrm{ME}}(i) = \hat{p}_{\mathrm{ME}}(i')$. Maximum entropy estimation has therefore been proposed with inequality constraints (cf. Khudanpur, 1995; Kazama & Tsujii, 2003):

$$\mathcal{M} = \left\{ p \,:\, a_j \leq p(A_j) \leq b_j,\ \ j = 1, \ldots, J \right\}. \tag{1.14}$$

To the best of our knowledge, there has not been much discussion in the literature of a principled way to make the choice of a_j and b_j, particularly of a way that depends on only the observed sample, and not on other ad hoc assumptions about p.

Yet another variation on maximum entropy consists of minimizing a functional of the form

$$\sum_{j=1}^{J} \mu_j d\left(p(A_j), \hat{a}_j\right) - H(p), \tag{1.15}$$

where $d(.,.)$ is some metric of deviation from the constraints of equation 1.12 and the parameters $\mu = (\mu_1, \ldots, \mu_J)$ are estimated, usually, from held-out data. Yet another way to relax the constraints in equation 1.12 is to note, using convex duality (Berger et al., 1996), that the parameters Λ that satisfy the constraints are exactly the parameters for which the model of equation 1.13 assigns maximum likelihood to the observed sample. One may then choose a penalized likelihood approach with a regularizing function of Λ. Still, several parameters need to be estimated from held-out data in either case. Several such methods are compared in Chen and Goodman (1996) for the estimation of bigram and trigram language models.

In section 2, we will seek to provide a principled way of relaxing the linear equality constraints in maximum entropy estimation.

1.5 Good-Turing and Other Held-Out Methods. In Jelinek (1998, p. 258), the author asks, "How much larger a probability should be assigned to an event observed once than to one not observed at all, or, in general, whether the ratio of probabilities of events observed n and m times, respectively, should really be n/m?"

Considering pmfs that put more mass on the observed counts than on any others, which we do in section 2, will lead to one answer to this question: equation 2.7. The Good-Turing and other held-out methods answer the question in a different way.

The basic idea is to divide the data into two parts. The first part, called the development set, is used for the collection of counts $\{n_i\}$. The second part, called the held-out set, is used to estimate additional parameters. A typical structure is as follows:

$$\tilde{p}_i = \begin{cases} \alpha \times \frac{n_i}{n} & \text{if } n_i > M, \\ q_i & \text{if } n_i \leq M, \end{cases} \tag{1.16}$$

where the (usually small) threshold M, and smoothed probability estimates q_i, $i = 0, \ldots, M$, are the additional parameters.

The Good-Turing estimate (Good, 1953; Orlitsky et al., 2003; McAllester & Schapire, 2000) is obtained by setting

$$q_i = \frac{r_{n_i+1}}{r_{n_i}} \frac{n_i + 1}{n}, \qquad i \in \{1, \ldots, k\}, \tag{1.17}$$

where r_c is the number of symbols $j \in \{1, \ldots, k\}$ whose count $n_j = c$. Thus, q_i for a symbol i depends not just on its count n_i and n, but on the counts of all other symbols.

Note that if $n_i > n_j$, it is not necessarily true that $q_i \geq q_j$, though this frequently holds in practice for symbols with very small counts. In other words, q_i may not respect the rank ordering implied by the empirical counts $\{n_i\}$, particularly for symbols with large counts. For this reason, the threshold M is often chosen to be small enough so as not to have this undesirable effect. In language modeling, for example, M is typically chosen to be 10 or less, depending on n. The parameter α is then computed so that $\tilde{p}_i$ sums to unity.

The Good-Turing estimate performs remarkably well for pmf on words. However, its derivation is somewhat ad hoc and unsatisfactory.

2 The Maximum Likelihood Set

One of the simplest and driving ideas in statistics is as follows: what we observe has to be fairly likely; otherwise we would not have observed it. One way to quantify this is to say that what we observe has to be more likely under the true pmf than any other comparable event. Let's define the

MLS as the set of pmfs that put more mass on the observed type than on any other type given n. Let $p = (p_1, \ldots, p_k)$ be a pmf over $\{1, \ldots, k\}$. The p-probability of observing the type $\hat{p} = (\frac{n_1}{n}, \ldots, \frac{n_k}{n})$ is

$$f(p, \hat{p}) = \frac{n!}{n_1! \ldots n_k!} \prod_{i=1}^{k} p_i^{n_i} . \tag{2.1}$$

The MLS, with these notations, is defined as

$$\mathcal{M}(\hat{p}) = \{p \in \mathcal{P}^k \ : \ \forall \hat{q} \in \mathcal{P}_n^k, \ f(p, \hat{p}) \geq f(p, \hat{q})\} . \tag{2.2}$$

We will see in section 2.3 that this set always contains the type $\hat{p}$, which is the MLE for p, and that it shrinks down to it as $n \to \infty$. For finite n, it contains pmfs that might reflect prior information such as smoothness or other desirable properties in a better way than the type, but still remain close to the observed counts. Moreover, this set is a close convex subset of $\mathcal{P}^k$, opening the way to numerical optimization.

Using Stirling formulas, as well as equation 1.4, one can check that

$$f(p, \hat{p}) \doteq 2^{-D(\hat{p}, p)}, \text{ where } u_n \doteq v_n \Leftrightarrow \lim_{n \to \infty} \frac{1}{n} \log \frac{u_n}{v_n} = 0. \tag{2.3}$$

Hence, for n sufficiently large, the MLS associated with a type $\hat{p}$ is roughly

$$\left\{p \in \mathcal{P}^k \ : \ D(\hat{p}, p) \leq D(\hat{q}, p), \quad \forall \hat{q} \in \mathcal{P}_n^k\right\}, \tag{2.4}$$

leading to the loose description that the MLS is the set of pmfs that are "closer" to the observed type than to any other.

2.1 Characterization of the Maximum Likelihood Set. The MLS admits a simpler though still implicit representation. Given the observed counts $(n_1, \ldots, n_k)$, define a neighborhood relationship on the set of types with denominator n: the neighbors of $(n_1, \ldots, n_k)$ are the types obtained by changing a single sample from one value to another one. That is, assume that for a pair of indexes $1 \leq i, j \leq k$, we have $n_j > 0$ and $n_i < n$; then $(n'_1, \ldots, n'_k)$, defined by

$$n'_i = n_i + 1, \quad n'_j = n_j - 1, \quad \text{and} \quad n'_l = n_l \quad l \neq i \text{ or } j, \tag{2.5}$$

is a neighbor of $(n_1, \ldots, n_k)$.

If a pmf is in the MLS, then it has to put more mass on the observed type than on any of its neighbors. It turns out that the converse is also true, which leads to the following result:

Proposition 1. *A pmf $p = (p_1, \ldots, p_k)$ on the set $\{1, \ldots, k\}$ belongs to the MLS $\mathcal{M}(\hat{p})$ associated with the counts $(n_1, \ldots, n_k)$ if and only if*

$$n_j p_i \leq (n_i + 1) p_j, \qquad \forall 1 \leq i \neq j \leq k, \tag{2.6}$$

or equivalently,

$$\frac{\hat{p}_i}{\hat{p}_j + \frac{1}{n}} \leq \frac{p_i}{p_j} \leq \frac{\hat{p}_i + \frac{1}{n}}{\hat{p}_j}, \qquad \forall 1 \leq i \neq j \leq k, \tag{2.7}$$

where, by convention, $\frac{a}{0} = +\infty$ whenever $a > 0$.

The proof uses elementary algebra and is relegated to the appendix.

2.2 Motivating Examples. For $k = 2$, the MLS is

$$\begin{aligned}\mathcal{M}(\hat{p}) &= \mathcal{M}\left(\left(\frac{n_1}{n}, 1 - \frac{n_1}{n}\right)\right) \\ &= \left\{p = (p_1, 1 - p_1);\ \frac{n_1}{n+1} \leq p_1 \leq \frac{n_1 + 1}{n+1}\right\}.\end{aligned}$$

Note that this set contains the type and shrinks down to it as the number of samples goes to infinity. Beside the connection with Dirichlet priors mentioned in section 1, the MLS in this case can be obtained through Bayesian estimation of a proportion with quadratic cost function and a beta(α, β) prior distribution. It is the set of estimators corresponding to the prior parameters (α, β) satisfying $\alpha + \beta = 1$ (see Hogg & Craig, 1995, p. 368).

The MLSs for $k = 3$ are illustrated in Figure 1 for two different values of n. The MLSs are convex cells with linear boundaries. They have at most $k \times (k - 1)$ boundaries, one corresponding to each neighboring type.

In order to select an estimate from the MLS, one could choose the pmf with maximum Shannon entropy. This choice will be motivated further in section 3. We use it here to illustrate properties of the MLS. For example, if the counts $(n_1, \ldots, n_k)$ are made of 0s and 1s only, then the pmf selected is the uniform distribution over $\{1, \ldots, k\}$, since it is of maximum entropy over all pmfs over $\{1, \ldots, k\}$ and it is included in the MLS, as one can check from equation 2.6. In contrast, if there is one value, say the first one, that gets all the counts, then the selected estimate is, for $n > 0$,

$$p_1^* = \frac{n}{n + k - 1}, \quad \text{and} \quad p_l^* = \frac{1}{n + k - 1}, \quad \forall 1 < l \leq k. \tag{2.8}$$

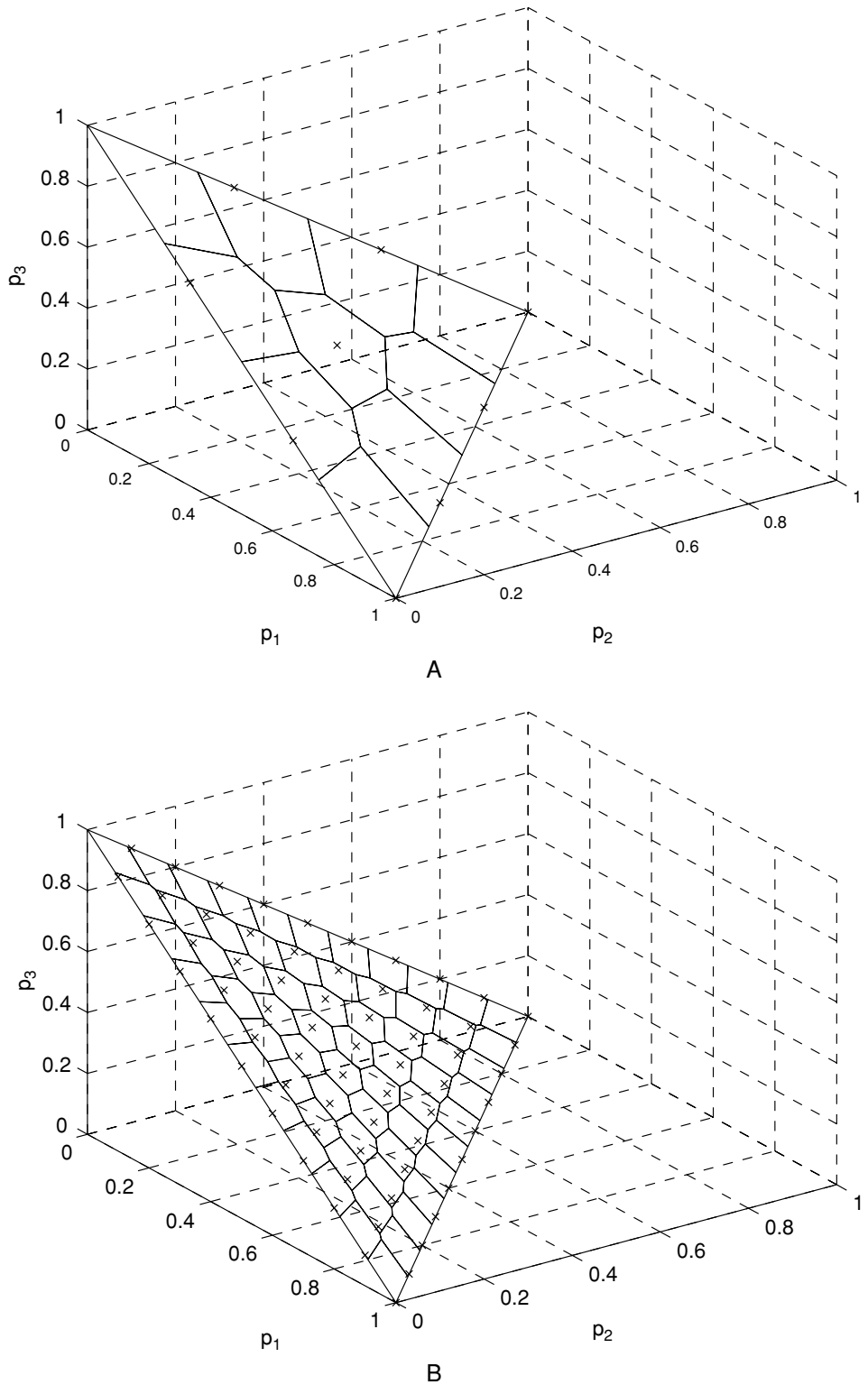


Figure 1: Illustration of the maximum likelihood sets for all the possible types for alphabet size $k = 3$. (A) $n = 3$ samples. (B) $n = 10$ samples. Each "cell" is an MLS containing exactly one type marked with a cross.

If $n < k$, then note that $p_1^* \leq 0.5$, which stands in sharp contrast with the estimate $\hat{p}_1 = 1$ given by the type. Equation 2.8 is a direct consequence of the property 3.4.

2.3 Properties of the Maximum Likelihood Set. We now present some insightful and useful properties of the MLS.

Proposition 2. *Let $\hat{p} = (\frac{n_1}{n}, \ldots, \frac{n_k}{n})$ be a type. The elements $p = (p_1, \ldots, p_k)$ of the MLS $\mathcal{M}(\hat{p})$ defined by $\hat{p}$ satisfy the following:*

$$\hat{p} \ll p \quad i.e. \quad n_i > 0 \Rightarrow p_i > 0, \quad \forall 1 \leq i \leq k, \tag{2.9}$$

$$n_i < n_j \Rightarrow p_i \leq p_j \quad \forall 1 \leq i, j \leq k, \tag{2.10}$$

$$\frac{n}{n+k}\,\hat{p}_i \;\leq p_i \leq\; \hat{p}_i + \frac{1}{n} \quad \forall 1 \leq i \leq k, \tag{2.11}$$

$$\|p - \hat{p}\|_1 \;=\; \sum_{i=1}^{k} |p_i - \hat{p}_i| \leq \frac{2(k-1)}{n}, \tag{2.12}$$

$$\hat{p} \in \mathcal{M}(\hat{p}), \tag{2.13}$$

but no other type with denominator n is an element of $\mathcal{M}(\hat{p})$. If $x_1, \ldots, x_n$ are independent samples with common pmf $q \in \mathcal{P}^k$, then the MLS defined by their type $\hat{p}$ is such that

$$\sup_{p \in \mathcal{M}(\hat{p})} \|p - q\|_1 \to 0 \quad as \quad n \to \infty \quad with\ probability\ 1. \tag{2.14}$$

Proposition 2 is essentially a corollary of proposition 1. Details of the proof are in the appendix. Properties 2.9 and 2.10 are desirable for any estimate of the pmf generating $x_1, \ldots, x_n$. Properties 2.11 and 2.12 show how the elements of the MLS may deviate from the underlying type. Property 2.14 shows that for a fixed k, as n gets large, all the elements in the MLS get closer to the pmf generating the samples.

It is easy to see, by comparing equation 2.11 and 1.9, that the MLS contains the Bayesian estimates for $0 \leq \beta \leq 1$.

3 Selecting an Element from the Maximum Likelihood Set

Every pmf in the MLS satisfies a number of properties, as outlined above, that one would consider desirable in an estimate of the pmf generating the samples $x_1, \ldots, x_n$, and we advocate $\mathcal{M}(\hat{p})$ as an admissible set from which a particular pmf may be selected using secondary criteria. One such criterion is outlined next.

Proposition 3. *Let $\hat{p} = (\frac{n_1}{n}, \ldots, \frac{n_k}{n})$ be a type and $\mathcal{M}(\hat{p})$ its associated MLS. Let $q = (q_1, \ldots, q_k)$ be a pmf such that $\hat{p} << q$. Then there exists a unique element $p^* \in \mathcal{M}(\hat{p})$ such that*

$$D(p^*, q) = \min_{p \in \mathcal{M}(\hat{p})} D(p, q)\,. \tag{3.1}$$

Note from equation 2.6 that $\mathcal{M}(\hat{p})$ is convex and closed in the Euclidean topology on $\mathcal{P}^k$. The existence of p^* therefore follows from theorem 2.1 in Csiszar (1975), and the uniqueness follows from the convexity of $p \mapsto D(p, q)$.

The pmf q may be viewed as a means of incorporating a prior estimate in the estimation process. In the case when $n \gg k$, the MLS has a very small radius, and the choice of q has a negligible effect on the choice of p^*. In the limit as $n \to 0$, $p^* \to q$ by continuity. Therefore, in the small sample situation, the choice of q will greatly influence p^*.

One may choose for q the uniform pmf over $\{1, \ldots, k\}$. p^* is then the element of $\mathcal{M}(\hat{p})$ with maximum Shannon entropy. It has been argued by Nemenman et al. (2002) that entropy might be the nonmetric (categorical data) analog of smoothness. Other compelling arguments for this choice have been made by Jaynes (1994).

In a situation where one needs to estimate a conditional pmf $p(\cdot|y)$ and the marginal pmf $p(\cdot)$ is known, a viable prior estimate is $q(\cdot) = p(\cdot)$. See Jelinek (1998) for related smoothing methods in language modeling.

If one chooses a measure such as the Kullback-Leibler (K-L) distance to select a pmf from the MLS, an additional satisfactory property of the selected pmf emerges.

Proposition 4. *Let $\mathcal{M}(\hat{p})$ be the MLS defined by the counts $(n_1, \ldots, n_k)$. For any pmf $q \gg \hat{p}$, the pmf*

$$p^* = \arg \min_{p \in \mathcal{M}(\hat{p})} D(p, q) \tag{3.2}$$

has the "monotonicity" property:

$$n_i = n_j \quad \textit{and} \quad q_i \geq q_j \quad \Rightarrow \quad p_i^* \geq p_j^* \quad \forall\, 1 \leq i \neq j \leq k. \tag{3.3}$$

Furthermore,

$$n_i = n_j \quad \textit{and} \quad q_i = q_j \quad \Rightarrow \quad p_i^* = p_j^* \quad \forall\, 1 \leq i \neq j \leq k. \tag{3.4}$$

The proof is again relegated to the appendix.

Every pmf $p \in \mathcal{M}(\hat{p})$ has been shown, via equation 2.10, to be faithful to the evidence. The monotonicity property, equation 3.3, characterizes the

selection rule of proposition 3: if i is a priori more likely than j, then, in the absence of evidence to the contrary, it continues to be more likely under the selected p^*. The special case, equation 3.4, has significant implications for the numerical computation of p^*, as will be discussed in the following section.

Note that the K-L divergence of equation 3.1 is not the only "distance" one may use to select a pmf from the MLS. Any other function $D(\cdot, \cdot)$ with a projection theorem that guarantees the existence and uniqueness of p^* in equation 3.1, together with an algorithm that computes the projection, may be used. An obvious choice is the Euclidean distance, which leads to a standard quadratic programming problem.

3.1 Numerical Optimization Issues. The optimization problem, equation 3.1, cannot in general be solved in closed form and in practice requires a numerical procedure. The setting is known in numerical optimization literature as general linearly constrained optimization (cf., Fletcher, 1981, and Bazaraa, Sherali, & Shetty, 1993). Stated briefly, one needs to minimize a convex function over a domain defined by linear inequalities such as equation 2.6. We minimize the K-L distance of equation 3.1 subject to p satisfying equation 2.6 using the numerical optimization package CFSQP developed by Lawrence, Zhou, and Tits (1997).

The number of constraints specifying the MLS is $k(k-1)$. A typical language modeling situation requires a vocabulary of $k \approx 10^5$ words. Checking just once that a pmf is inside the domain therefore may in general require about 10^{10} operations. Fortunately, choosing q to be piecewise constant considerably reduces the dimensionality. To see this, consider the extreme situation where q is the uniform pmf. Two indexes $1 \le i,\ j \le k$ may be considered equivalent if $n_i = n_j$, and the optimization may be performed over the set of pmfs on $\{1, \ldots, k\}$ modulo this equivalence relation, thanks to equation 3.4. What is the number of indexes in this set? With n samples, it contains no more than $\sqrt{2n}$ indexes. This is therefore the "effective" k when q is uniform. For other pmfs q, the corresponding equivalence relation is $n_i = n_j$ together with $q_i = q_j$.

4 Language Modeling

Statistical language models are a key component in applications such as automatic speech recognition, machine translation, spelling correction, and document retrieval. Language modeling entails estimating a probability distribution over word sequences, and this is typically done by modeling the sequence of words in a sentence by a finite memory Markov chain. An n-gram model is a set of conditional pmfs $P(w_n|w_1, \ldots, w_{n-1})$, one for every conditioning event. In applications such as document retrieval, where word order is not of paramount importance and a bag-of-words representation is

adequate, i.i.d. models, called unigram models, are used. In all cases, there is a need to estimate a pmf, marginal or conditional, on the vocabulary. In this section, we present experimental results for the estimation of unigram models.

If obtaining smooth estimates is the primary goal, one would naturally use the uniform distribution in the role of q in equation 3.1. We obtain empirical results for this (maximum entropy estimation) case as a first step. It should be clear to the reader, however, that all words are not equally likely even a priori, and it is known from several studies that the count n_i and the rank of a word i, when the vocabulary is sorted in order of decreasing counts, has a roughly inverse relationship. The relationship, sometimes called Zipf's law (cf. Li, 1999), makes for a natural prior estimate q for estimating the unigram pmf via equation 3.1. Specifically, we consider

$$q_{\text{Zipf}}(i) = \frac{\alpha(k)}{\text{rank}(i)}, \tag{4.1}$$

where $\alpha(k)$ is a normalizing constant. Empirical studies (Ha, Sicilia, Ming, & Smith, 2002) show that this is a good initial estimate for unigrams. Note that α need not be computed, since it plays no role in the minimization of equation 3.1. The resulting estimate p^* in the MLS may then be interpreted as the pmf supported by the evidence $x_1, \ldots, x_n$, which is closest to Zipf's law in the sense of K-L divergence. This seems a plausible choice for language modeling.

A problem, however, remains: for a given vocabulary, there is no a priori way of determining the rank ordering of words. One could possibly use word length to perform such ordering. We take a simpler approach and use the rank ordering empirically observed in $x_1, \ldots, x_n$ to determine q. We make a further modification to break ties: all words that have the same count in $x_1, \ldots, x_n$ get a rank, namely, the mean of the ranks spanned by those equal-count words. This modification results in an important numerical simplification. By assuming words with the same observed counts to have the same q-probability, we are assured that they will have the same p^* probability, reducing the number of free variables in the numerical optimization of equation 3.1 and indeed the specification of p^*. Without this modification, p^* would have up to $k - 1$ free parameters, and in case of most language models, this is impractical.

We have conducted experiments on English text from the *Wall Street Journal* corpus, which contains articles from the general news and financial domain. A particular subset of this corpus, the UPenn Treebank corpus (http://www.cis.upenn.edu/~treebank/home.html), has been widely used by many researchers in language modeling, and we use this for our experiments as well. The corpus is divided into sections, numbered 00 through 24. We use sections 00 to 20 as our training corpus; it contains 900,000 word tokens. Sections 21 and 22, containing 100,000 tokens, are used variably as a

training or a held-out corpus as needed, and sections 23 and 24, containing 100,000 tokens make up our test corpus. For the purpose of studying the variability of the estimates, we divided sentences in sections 00 to 22 into 10 roughly equal parts, and results will be presented on these smaller corpora in the following.

We made a list of all seen words from sections 00 to 22 and augmented this vocabulary with a set of "unseen" words. The decision on how many unseen words to include is ad hoc. We use a leave-one-out estimate of the number of unseen words by asking, for each x_t in $x_1, \ldots, x_n$, whether it would be an unseen word if the vocabulary were to be extracted from $\{x_1, \ldots, x_{t-1}, x_{t+1}, \ldots, x_n\}$, $t = 1, \ldots, n$. It is easy to see that this procedure yields $n_0 = n_1$; the number of unseen words is exactly equal to the number of words seen only once in the corpus. This procedure, while not theoretically satisfactory, is performed out of necessity.

We remark that the MLS of equation 2.2 is well defined even for an infinite vocabulary, and with a suitable prior estimate q, it may be possible to let the vocabulary size be unbounded for the estimate of equation 3.1 as well.

4.1 Empirical Results. The box at the top of Figure 2 illustrates, using crosses, the empirical pmf $\hat{p}$ obtained from sections 00 to 22, where the words have been (re)ordered along the abscissa in decreasing order of $\hat{p}_i$. Specifically, for $i = 1, \ldots, k_0$, the ordinate shows the logarithm (to the base 2) of

$$\frac{n_{\sigma(1)}}{n}, \ldots, \frac{n_{\sigma(k_0)}}{n}, \tag{4.2}$$

with $n_{\sigma(1)} \geq \ldots \geq n_{\sigma(k_0)}$. $k_0 = 37{,}001$ is the number of distinct words seen in sections 00 to 22. The Zipf prior of equation 4.1 is shown in the same box using dots: it is a straight line with slope -1. A uniform prior would be a horizontal line on this plot. Finally, in the same box, the lower and upper bounds on each p_i in the MLS, per equation 2.11, are also illustrated using a solid and a dashed line, respectively:

$$\left\{ \left(\log i, \log \frac{n_{\sigma(i)}}{n+k} \right) \text{ and } \left(\log i, \log \frac{n_{\sigma(i)}+1}{n} \right), \ 1 \leq i \leq k_0 \right\}, \tag{4.3}$$

where the number of words in the vocabulary $k = 52{,}743$ is estimated using the procedure described above. Note that the envelope of the MLS has a trumpet-like shape. For large counts, the upper bound of the MLS is essentially indistinguishable from the type. The estimated pmf p^* may decrease the mass for these outcomes but cannot increase it significantly. However, for small counts, the envelope of the MLS has a flared bell shape showing the statistical variability of the corresponding probabilities and that the type tends to underestimate rare events. Any pmf chosen from

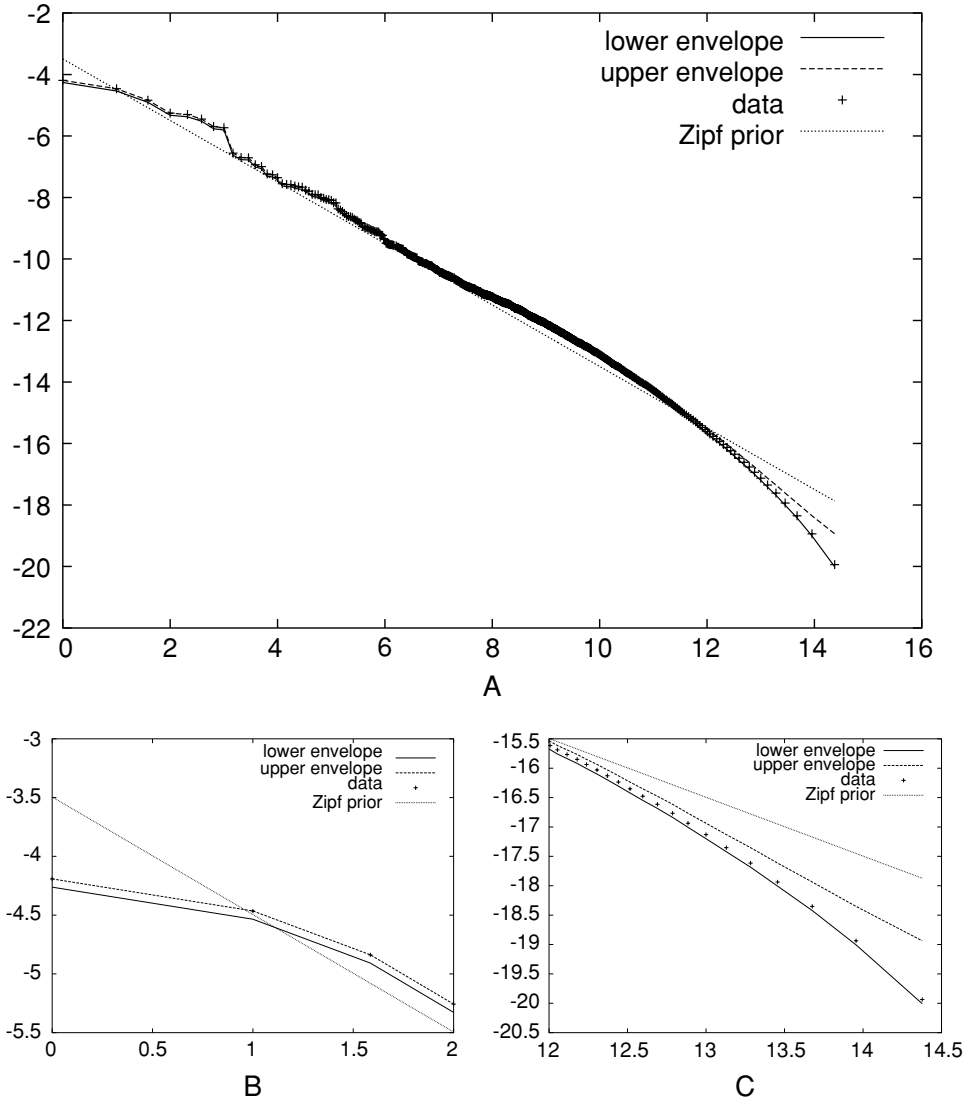


Figure 2: Plot of the empirical pmf from data, the Zipf prior, and the lower and upper envelopes of the MLS on a log-log scale. (A) Full range of observed counts. (B) Zoom top left (≡ high counts). (C) Zoom bottom right (≡ low counts).

the MLS corresponds to a curve that lies between the upper and lower envelopes.

To measure the efficacy of an estimate $\tilde{p}$ of p, we compute the average code word length (in bits) that the estimate $\tilde{p}$ achieves on the type $\hat{p}_T$ of the test set, that is,

$$\ell(\tilde{p}) \;=\; \frac{1}{n_T}\sum_{t=1}^{n_T}\log\frac{1}{\tilde{p}(x_t)} \;=\; D(\hat{p}_T,\,\tilde{p}) + H(\hat{p}_T)\,, \tag{4.4}$$

Table 1: Code Word Length in Bits for pmf Estimates.

	$\hat{p}_\beta\ \beta = 1$	$\hat{p}_\beta \beta = \frac{1}{2}$	$\hat{p}_\beta\ \beta = \frac{1}{k}$	$\hat{p}_{\mathrm{GT}}$
$\ell(\cdot)$	10.21	10.21	10.52	10.19
	$p^* : q = \text{unif}$	$p^* : q = \text{Zipf}$	$p^* : q = \hat{p}_{\mathrm{GT}}$	
$\ell(\cdot)$	10.21	10.20	10.19	

	$\hat{p}_\beta\ \beta = 1$	$\hat{p}_\beta\ \beta = \frac{1}{2}$	$\hat{p}_\beta\ \beta = \frac{1}{k}$	$\hat{p}_{\mathrm{GT}}$
Average $\ell(\cdot)$	10.58	10.42	11.31	10.37
SD	0.017	0.017	0.036	0.016
	$p^* : q = \text{unif}$	$p^* : q = \text{Zipf}$	$p^* : q = \hat{p}_{\mathrm{GT}}$	
Average $\ell(\cdot)$	10.58	10.40	10.37	
SD	0.015	0.017	0.018	

Notes: Upper table: $n = 10^6$ words. Lower table: average and standard deviation over 10 training sets with $n = 10^5$ words. $\hat{p}_\beta$ is the add-β rule of equation 1.9. $\hat{p}_{\mathrm{GT}}$ is the Good-Turing estimate of equations 1.16 and 1.17. p^* is the MLS estimate of equation 3.1 with the prior q as indicated.

where n_T is the size of the test set, the x_ts are the words of the test set, and $H(\cdot)$ is the Shannon entropy.

Experimental results, for the *Wall Street Journal* data, along with standard deviations, when available, are shown in Table 1.

Looking at the average code word lengths in Table 1, the reader unfamiliar with language modeling might be surprised to see how well the Good-Turing (G-T) estimate (fifth column) performs compared to the add-β rules. Three MLS-derived estimates are presented. In the first of these, we have used the uniform pmf as a prior. The estimate thus obtained has comparable performance with the add-1 rule but not as good as the add-$\frac{1}{2}$ rule for the smaller training set. Next, using a Zipf prior, we increase the performance to outperform all add-β rules considered so far and come closer to the GT estimate. Third, we use the GT estimate itself as a prior. We then get an average code word length that is indistinguishable from the GT estimate. In our experiments, the GT estimate has never been inside the MLS. We have thus shown empirically that there exist pmfs that are "closer" to the empirical pmf than to any other type whose code word lengths are undistinguishable from those of the GT estimate. Furthermore, unlike the GT estimate, these pmfs are guaranteed not to contradict the observed counts in the data.

Note as an aside that the effective-k for numerical optimization is about 600 for $n = 10^6$ and about 180 when $n = 10^5$ for all priors used.

5 Conclusion

We have proposed a new method for estimating a probability mass function from a sample: we consider the observed counts; the maximum likelihood

set is defined as the set of pmfs that put more mass on the observed counts than on any other set of counts; the closest element from the MLS to a prior estimate in the Kullback-Leibler sense is then selected.

The MLS is an admissible set for estimating a pmf that has the following properties: it is built from first principles, and it is strongly consistent (see equation 2.14) and faithful to the evidence (see equations 2.9 and 2.10).

The way we select a pmf from the MLS permits encoding domain-specific information in a very natural way, as demonstrated with the Zipf law for language modeling. Moreover, it is practical, as it entails minimizing a convex function over a domain defined by linear inequalities. This is a classic problem in numerical analysis, with known solutions. This way of incorporating domain information is a novel alternative to Bayesian or minimax methods.

Experiments with pmfs on English words show that the proposed method is competitive with state-of-the-art methods.

Appendix: Proofs of Propositions 1, 2, and 4

Proof of Proposition 1. First, we establish that if $p \in \mathcal{M}(\hat{p})$, then p satisfies equation 2.6. Toward this end, for any i and any $j \neq i$ such that $n_j > 0$, let

$$\hat{q} = \left(\frac{n_1}{n}, \ldots, \frac{n_i + 1}{n}, \ldots, \frac{n_j - 1}{n}, \ldots, \frac{n_k}{n} \right). \tag{A.1}$$

By definition, $f(p, \hat{p}) \geq f(p, \hat{q})$, and hence

$$\begin{aligned} &\frac{n!}{n_1! \cdots n_i! \cdots n_j! \cdots n_k!} \prod_l p_l^{n_l} \\ &\quad \geq \frac{n!}{n_1! \cdots (n_i + 1)! \cdots (n_j - 1)! \cdots n_k!} p_i^{n_i + 1} p_j^{n_j - 1} \prod_{l \neq i,\, j} p_l^{n_l} \\ &\quad \frac{1}{n_j} p_j \geq \frac{1}{n_i + 1} p_i. \end{aligned}$$

Property 2.6 follows. If $n_j = 0$, then equation 2.6 follows trivially.

Next, we establish that if p satisfies equation 2.6, then $p \in \mathcal{M}(\hat{p})$. Toward this end, again, let

$$\hat{q} = \left(\frac{\tilde{n}_1}{n}, \ldots, \frac{\tilde{n}_k}{n} \right) \tag{A.2}$$

be an empirical pmf associated with any other set of counts $(\tilde{n}_1, \ldots, \tilde{n}_k)$ for an n-length sample. We construct a sequence of pmfs $\hat{q}^{(0)}, \ldots, \hat{q}^{(n)}$ such that

$$\hat{q}^{(0)} = \hat{q}, \quad f\left(p, \hat{q}^{(0)}\right) \leq f\left(p, \hat{q}^{(1)}\right) \leq \ldots \leq f\left(p, \hat{q}^{(n)}\right) \quad \text{and} \quad \hat{q}^{(n)} = \hat{p}. \tag{A.3}$$

In particular, we begin with $\hat{q}^{(0)}$ defined by the counts

$$\left(n_1^{(0)}, \ldots, n_k^{(0)}\right) = (\tilde{n}_1, \ldots, \tilde{n}_k), \tag{A.4}$$

and, for $m = 1, \ldots, n$,

- If $\hat{q}^{(m-1)} = \hat{p}$, then we set $\hat{q}^{(m)} = \hat{q}^{(m-1)}$.
- Otherwise, choose i and j such that $n_i^{(m-1)} > n_i$ and $n_j^{(m-1)} < n_j$, and define $\hat{q}^{(m)}$ by the counts

$$n_i^{(m)} = n_i^{(m-1)} - 1, \quad n_j^{(m)} = n_j^{(m-1)} + 1, \quad \text{and}$$
$$n_l^{(m)} = n_l^{(m-1)} \text{ for all other } l. \tag{A.5}$$

Note that a suitable pair i, j is guaranteed to exist whenever $\hat{q}^{(m-1)} \neq \hat{p}$. It is clear that for $m = 1, \ldots, n$, if $\hat{q}^{(m-1)} \neq \hat{p}$, then by construction,

$$\begin{aligned}\left\|\hat{q}^{(m)} - \hat{p}\right\|_1 &= \left\|\hat{q}^{(m-1)} - \hat{p}\right\|_1 - \frac{2}{n} \\ &= \cdots = \left\|\hat{q}^{(0)} - \hat{p}\right\|_1 - \frac{2m}{n}.\end{aligned}$$

Since $\|\hat{q} - \hat{p}\|_1 \leq 2$, it follows that $\hat{q}^{(n)} = \hat{p}$.

Finally, note that for $m = 1, \ldots, n$, if $\hat{q}^{(m-1)} \neq \hat{p}$,

$$\begin{aligned}\frac{f(p, \hat{q}^{(m)})}{f(p, \hat{q}^{(m-1)})} &= \frac{n!}{n_1^{(m)}! \cdots n_k^{(m)}!} \frac{n_1^{(m-1)}! \cdots n_k^{(m-1)}!}{n!} \prod_{l=1}^{k} p_l^{n_l^{(m)} - n_l^{(m-1)}} \\ &= \frac{1}{n_j^{(m-1)} + 1} \frac{n_i^{(m-1)}}{1} \frac{p_j}{p_i} \\ &\geq \frac{n_i + 1}{n_j} \frac{p_j}{p_i} \\ &\geq 1,\end{aligned}$$

where the first inequality holds by construction, since $n_i^{(m-1)} > n_i$ and $n_j^{(m-1)} < n_j$, and the second inequality holds due to equation 2.6.

Proof of Proposition 2. Let us suppose that there is an index $1 \leq j \leq k$ such that $n_j > 0$ and $p_j = 0$. Replacing in equation 2.6, it implies that $\forall 1 \leq i \leq k, i \neq j$, $p_i = 0$ which is impossible since $p_j = 0$. This proves equation 2.9. Equation 2.10 is also a consequence of equation 2.6, as the reader can check.

We remark that equation 2.6 still holds for indexes $i = j$. Then, summing out, we obtain, for any subset $A \subset \{1, \ldots, k\}$,

$$\sum_{i \in A} \sum_{j=1}^{k} \hat{p}_j \, p_i \leq \sum_{i \in A} \sum_{j=1}^{k} \left(\hat{p}_i + \frac{1}{n} \right) p_j \text{ and} \tag{A.6}$$

$$\sum_{j \in A} \sum_{i=1}^{k} \hat{p}_j \, p_i \leq \sum_{j \in A} \sum_{i=1}^{k} \left(\hat{p}_i + \frac{1}{n} \right) p_j , \tag{A.7}$$

from which we obtain

$$\forall A \subset \{1, \ldots, k\}, \; \hat{p}(A) \frac{n}{n+k} \leq p(A) \leq \hat{p}(A) + \frac{\#A}{n} , \tag{A.8}$$

where $\#A$ is the number of elements in A. Setting $A = \{i\}$ gives equation 2.11. Now, from Cover and Thomas (1991, p. 300),

$$\| p - \hat{p} \|_1 = 2(p(A) - \hat{p}(A)); \; A = \{1 \leq i \leq k; \; p_i > \hat{p}_i\} . \tag{A.9}$$

Using equation A.8, we obtain

$$\| p - \hat{p} \|_1 \leq 2 \frac{\#A}{n} \leq \frac{2(k-1)}{n} . \tag{A.10}$$

Using equation 2.6, one can directly check that $\hat{p} \in \mathcal{M}(\hat{p})$. If another type in $\mathcal{P}_n^k$ is also an element of $\mathcal{M}(\hat{p})$, then $\hat{p}$ has a neighbor that is an element of $\mathcal{M}(\hat{p})$, following the argument in the part ($\Leftarrow$) of the proof of proposition 1. Let's call $\hat{q}$ this neighbor. It is such that $\hat{q}_i = \frac{n_i + 1}{n}$ and $\hat{q}_j = \frac{n_j - 1}{n}$ for some indexes $1 \leq i, \, j \leq k$ such that $n_i < n$ and $n_j > 0$. Now, as an element of $\mathcal{M}(\hat{p})$, it satisfies

$$(n_i + 1) \hat{q}_j \geq n_j \hat{q}_i . \tag{A.11}$$

But this is equivalent to saying that $n_i \leq -1$, which is impossible.

Finally,

$$\sup_{p \in \mathcal{M}(\hat{p})} \| p - q \|_1 \leq \frac{2(k-1)}{n} + \| \hat{p} - q \|_1 , \tag{A.12}$$

using the triangular inequality as well as the bound, equation 2.12. Equation 2.14 follows from the fact that the type converges to the true distribution in $\| . \|_1$.

Proof of Proposition 4. Assume, to the contrary, that $p_i^* < p_j^*$ for some $i \neq j$ with $n_i = n_j$ and $q_i \geq q_j$. Define a pmf p^{**} by

$$p_l^{**} = \begin{cases} p_j^* & \text{for } l = i, \\ p_i^* & \text{for } l = j, \\ p_l^* & \text{for } l \neq i \text{ or } j. \end{cases} \tag{A.13}$$

In other words, construct p^{**} by "switching" the ith and the jth entries of p^*. Since $p^* \in \mathcal{M}(\hat{p})$, p^* satisfies equation 2.6. But $n_i = n_j$ then implies that, by construction, p^{**} also satisfies equation 2.6. Thus, $p^{**} \in \mathcal{M}(\hat{p})$. Next, note that

$$\begin{aligned} D(p^*\|q) - D(p^{**}\|q) &= \sum_{l=1}^{k} p_l^* \log \frac{p_l^*}{q_l} - \sum_{l=1}^{k} p_l^{**} \log \frac{p_l^{**}}{q_l} \\ &= p_i^* \log \frac{p_i^*}{q_i} + p_j^* \log \frac{p_j^*}{q_j} - p_j^* \log \frac{p_j^*}{q_i} - p_i^* \log \frac{p_i^*}{q_j} \\ &= p_i^* \log \frac{q_j}{q_i} - p_j^* \log \frac{q_j}{q_i} \\ &= \left(p_i^* - p_j^*\right) \log \frac{q_j}{q_i} \geq 0 \end{aligned}$$

which contradicts proposition 3, since p^* is the unique minimizer of $D(p\|q)$ in $\mathcal{M}(\hat{p})$.

Acknowledgments

We thank Ali Yazgan for his valuable assistance in the use of the CFSQP package and in conducting most of the empirical studies in section 4.1. This research was partially supported by the National Science Foundation via grants ITR-0225656 and IIS-9982329, ARO DAAD19/-02-1-0337, and general funds from the Center for Imaging Science at the Johns Hopkins University. Finally, we are grateful to the anonymous referees, who gave several insightful suggestions toward improving this article.

References

Bazaraa, M. S., Sherali, H. D., & Shetty, C. (1993). *Nonlinear programming*. New York: Wiley.

Berger, A. L., Della Pietra, S. A., & Della Pietra, V. J. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, *22*, 39–71.

Chen, S. F., & Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics* (pp. 310–318). Santa Cruz, CA: Association for Computational Linguistics.

Chen, S. F., & Rosenfeld, R. (1999). *A gaussian prior for smoothing maximum entropy models* (Tech. Rep.). Pittsburgh, PA: Carnegie Mellon University.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Csiszar, I. (1975). I-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, *3*, 146–158.

Fletcher, R. (1981). *Practical methods of optimization* (Vol. 2). New York: Wiley.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, *40*, 237–264.

Ha, L. Q., Sicilia, E., Ming, J., & Smith, F. J. (2002). Extension of Zipf's law to words and phrases. In *International Conference on Computational Linguistics (COLING'2002)* (pp. 315–320). Taipei, Taiwan.

Hogg, R. V., & Craig, A. T. (1995). *Introduction to mathematical statistics*. Upper Saddle River, NJ: Prentice Hall.

Jaynes, E. T. (1994). *Probability theory: The logic of science*. Cambridge: Cambridge University Press.

Jelinek, F. (1998). *Statistical methods for speech recognition*. Cambridge, MA: MIT Press.

Kazama, J., & Tsujii, J. (2003). Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing* (pp. 137–144). Sapporo, Japan.

Khudanpur, S. (1995). A method of ME estimation with relaxed constraints. In *Proceedings of the Johns Hopkins University Language Modeling Workshop* (pp. 1–17). Baltimore: Center for Language and Speech Processing, Johns Hopkins University.

Lawrence, C. T., Zhou, J. L., & Tits, A. L. (1997). *User's guide for CFSQP version 2.5: A C code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints* (Tech. Rep. No. TR-94-16r1). College Park: Institute for Systems Research, University of Maryland.

Li, W. (1999). References on Zipf's law. Available online: http://linkage.rockefeller.edu/wli/zipf/.

Lidstone, G. (1920). Note on the general case of the Bayes-Laplace formula for inductive or posterior probabilities. *Trans Fac. Actuaries*, *8*, 182–192.

McAllester, D., & Schapire, R. E. (2000). On the convergence rate of Good-Turing estimators. In *Proc. 13th Annual Conference on Computational Learning Theory*, (pp. 1–6). San Francisco: Morgan Kaufmann.

Nemenman, I., Shafee, F., & Bialek, W. (2002). Entropy and inference, revisited. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 14*. Cambridge, MA: MIT Press.

Olkin, I., & Sobel, M. (1979). Admissible and minimax estimation for the multinomial distribution and k independent binomial distributions. *Annals of Mathematical Statistics*, *7*, 284–290.

Orlitsky, A., Santhanam, N. P., & Zhang, J. (2003). Always good Turing: Asymptotically optimal probability estimation. *Science*, *302*, 427–431.

Poschel, Ebeling, W., Froemmel, C., & Ramirez, R. T., (2003). Correction algorithm for finite sample statistics. *Eur. Physics*, *12*, 531–541.

Ristad, E. S. (1995). *A natural law of succession* (Tech. Rep. No. CS-TR-495-95). Princeton, NJ: Department of Computer Science, Princeton University.

Steinhaus, H. (1957). The problem of estimation. *Annals of Mathematical Statistics*, *28*, 633–648.

Received February 17, 2004; accepted January 5, 2005.

Chapter 9

Finding a needle in a haystack: conditions for reliable detection in the presence of clutter

Finding a Needle in a Haystack: Conditions for Reliable Detection in the Presence of Clutter*

Bruno Jedynak‡and Damianos Karakos§

October 23, 2006

Abstract

We study conditions for the detection of an N-length iid sequence with unknown pmf p_1, among M N-length iid sequences with unknown pmf p_0. We show how the quantity $M2^{-N\,D(p_1||p_0)}$ determines the asymptotic probability of error.

Keywords: reliable detection, probability of error, Sanov's theorem, Kulback-Leibler distance, phase transition.

1 Introduction

Our motivation for this paper has its origins in Geman et. al. (1996), where an algorithm for tracking roads in satellite images was experimentally studied. Below a certain clutter level, the algorithm could track a road acurately, and suddenly, with increased clutter level, tracking would become impossible. This phenomenon was studied theoretically in Yuille et. al.(2000 and 2001) . Using a simplified statistical model, the authors show that, in an

*This research was partially supported by ARO DAAD19/-02-1-0337 and general funds from the Center for Imaging Science at The Johns Hopkins University.

†USTL and Department of Applied Mathematics, Johns Hopkins University

‡Mail should be adressed to: Bruno Jedynak, Clark 302b, Johns Hopkins University, Baltimore, MD, 21286-2686

§Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, 21286-2686

apropriate asymptotic setting, the number of false detections is subject to a phase transition. Our objective in this paper is to generalize these results. First, we demonstrate, in the same setting, that the phase transition phenomenon occurs for the error rate of the maximum likelihood estimator. Second, we consider the situation where the underlying statistical model is unknown; i.e., there is a special object among many others but it is not known *how* it is special (it is an outlier, in some sense). We show that the same phase transition phenomenon occurs in this case as well. Moreover, we propose a target detector that has the same asymptotic performance as the maximum likelihood estmator, had the model been known. Simulations illustrate these results.

Let

$$X = \begin{pmatrix} X_1^1 & X_1^2 & \dots & X_1^N \\ X_2^1 & X_2^2 & \dots & X_2^N \\ \vdots & \vdots & & \vdots \\ X_{M+1}^1 & X_{M+1}^2 & \dots & X_{M+1}^N \end{pmatrix} \tag{1}$$

be a $(M+1) \times N$ matrix made of independent random variables (rvs) taking values in a finite set. We denote by $X_m = (X_m^1, \dots, X_m^N) \in \mathcal{X}^N$ the rvs in line m and by $X_{(m)}$ the ones that are *not* in line m. There is a special line, the *target*, with index t. All the other lines will be called *distractors*. The rvs X_t are identically distributed with point mass function (pmf) p_1. The other ones, $X_{(t)}$, are identically distributed with pmf $p_0 \neq p_1$. The goal is to estimate t, the target, from a single realization of X. If p_0 and p_1 are "close", the target does not differ much from the distractors, a situation akin to "finding a needle in a haystack".

2 Known distributions

Let x be a realization of X. Then, the log-likelihood[1] of x is

$$l(x) = \sum_{n=1}^{N} \log p_1(x_t^n) + \sum_{m=1, m\neq t}^{M+1} \sum_{n=1}^{N} \log p_0(x_m^n) \quad (2)$$

$$= \sum_{n=1}^{N} \log \frac{p_1(x_t^n)}{p_0(x_t^n)} + \sum_{m=1}^{M+1} \sum_{n=1}^{N} \log p_0(x_m^n) \quad (3)$$

The maximum likelihood estimator (mle) for t is then

$$\hat{t}(x) = \arg \max_{1\leq m\leq M+1} \sum_{n=1}^{N} \log \frac{p_1(x_m^n)}{p_0(x_m^n)} \quad (4)$$

We call the *reward* of line m the quantity

$$\frac{1}{N} \sum_{n=1}^{N} \log \frac{p_1(x_m^n)}{p_0(x_m^n)} \quad (5)$$

The mle entails choosing the line with the largest reward. The quantity of interest is the probability that the mle differs from the target:

$$e(M, N) = I\!P(\hat{t}(X) \neq t) \quad (6)$$

which is the probability that a distractor gets a reward which is greater than the reward of the target. If M is fixed , letting $N \to \infty$, and using the law of large numbers, we obtain

$$\frac{1}{N} \sum_{n=1}^{N} \log \frac{p_1(x_t^n)}{p_0(x_t^n)} \to D(p_1, p_0) \text{ and} \quad (7)$$

$$\frac{1}{N} \sum_{n=1}^{N} \log \frac{p_1(x_m^n)}{p_0(x_m^n)} \to -D(p_0, p_1) \text{ for every } m \neq t \quad (8)$$

[1] Logarithms are base 2 throughout the paper.

almost surely, where

$$D(p,q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{9}$$

is the Kulback-Leibler distance between p and q. Hence, as long $p \neq q$, $D(p,q) > 0$, and the reward of the target converges to a positive value while the reward of each distractor converges to a negative value which allows us to show that the error of the mle goes to zero. One can even bound $e(M,N)$ from above for any fixed M and N as follows

Theorem 1

$$e(M,N) \leq M \left(\sum_x \sqrt{p_0(x)p_1(x)} \right)^{2N}. \tag{10}$$

Note that

$$0 \leq \sum_x \sqrt{p_0(x)p_1(x)} = 1 - Hellinger(p_0,p_1) \leq 1, \tag{11}$$

where $Hellinger(p_0,p_1)$ is the Hellinger distance between p_0 and p_1. The proof, using classical large deviations techniques, is in Section 6. Note that if the right-hand side of (10) goes to 0 as $M \to \infty$ and $N \to \infty$, the probability that the mle differs from the target goes to 0. This condition, however, is not necessary. As we show below, there is a maximum rate at which M can go to infinity in order for the probability of error to go to zero (if M increases faster, then the probability of error goes to one). A similar result, i.e., that the number of distractors for which the reward is larger than the reward of the target follows a phase transition, was also shown by Yuille et. al.(2000). We present below the same analysis for the convergence of the mle. The phase transition, or in other words, the dependence of the probability of error on the rate at which M goes to infinity, is expressed in the following theorem:

Theorem 2

$$\textit{If } \exists \varepsilon > 0, \textit{ such that } \lim_{M,N\to\infty} M 2^{-N(D(p_1,p_0)-\varepsilon)} = 0 \quad \textit{then } \lim_{M,N\to\infty} e(M,N) = 0, \tag{12}$$

and

$$\textit{If } \exists\, \varepsilon > 0, \textit{ such that } \lim_{M,N\to\infty} M2^{-N(D(p_1,p_0)+\varepsilon)} = +\infty \quad \textit{then } \lim_{M,N\to\infty} e(M,N) = 1. \quad (13)$$

The intuition is as follows. First, as N goes to infinity, if M remains fixed, the probability of error goes to zero (exponentially fast, following a large deviation phenomenon) since the reward of the target line converges to a positive value, while the reward of the distractors converges to a negative value (as was mentioned earlier). On the other hand, as the number M of distractors increases, when N remains fixed, the probability that there exists a distractor with a reward larger than the reward of the target increases as well. These are two competing phenomena, whose interaction gives rise to the "critical rate" $D(p_1, p_0)$. The detailed proof appears in Section 6.

Note: In order for the limits of functions of M, N to be well-defined as $M, N \to \infty$, we assume that M is, in general, a function of N. Hence, all limits $\lim_{M,N\to\infty}$ should be interpreted as $\lim_{N\to\infty}$, with the proviso that M is increasing according to some function of N. We kept the notation $\lim_{M,N\to\infty}$ for simplicity.

3 Unknown Distributions

We now look at the case where p_0 and p_1 are unknown. It is clear that the error rate of any estimator in this context cannot be lower than the error rate of the mle (with known p_0 and p_1). Hence, (13) holds even when $e(M, N)$ is the error rate of any estimator. Can one build an estimator of the target for which the error rate will satisfy (12) ? The answer is yes as we shall see now.

A simple way of building an estimator of the target when p_0 and p_1 are unknown is to plug-in estimators of p_0 and p_1 in the previous (mle) estimator (4). Hence, let us define

$$\tilde{t}(x) = \arg \max_{1 \le m \le M+1} \sum_{n=1}^{N} \log \frac{\hat{p}_m(x_m^n)}{\hat{p}_{(m)}(x_m^n)} \tag{14}$$

where $\hat{p}_m$ and $\hat{p}_{(m)}$ are the empirical distributions of the rvs in line m and in all the other lines, respectively. I.e.,

$$\hat{p}_m(x) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\{X_m^n = x\}, \tag{15}$$

and

$$\hat{p}_{(m)}(x) = \frac{1}{MN} \sum_{n=1}^{N} \sum_{j=1, j \ne m}^{M+1} \mathbf{1}\{X_j^n = x\}. \tag{16}$$

Note that

$$\tilde{t}(x) = \arg \max_{1 \le m \le M+1} D(\hat{p}_m, \hat{p}_{(m)}). \tag{17}$$

Hence, $\tilde{t}$ is the line that differs the most (in the Kulback-Leibler sense) from the average distribution of the *other* lines. (The reader may be more familiar with the variant

$$\dot{t}(x) = \arg \max_{1 \le m \le M+1} D(\hat{p}_m, \hat{p}), \tag{18}$$

where $\hat{p}$ is the empirical distribution over all rvs, including line m; both $\dot{t}$ and $\tilde{t}$ are similar in the sense that they pick the sequence which differs the most from the rest.)

It turns out that the error rate of $\tilde{t}$, that is

$$\tilde{e}(M, N) = I\!P(\tilde{t}(X) \ne t), \tag{19}$$

where, as before, t denotes the target, has the same asymptotic behavior as the mle (4) in the case of known distributions.

Theorem 3

$$\textit{If } \exists \varepsilon > 0, \textit{ such that } \lim_{M,N \to \infty} M 2^{-N(D(p_1, p_0) - \varepsilon)} = 0 \quad \textit{then } \lim_{M,N \to \infty} \tilde{e}(M, N) = 0 \tag{20}$$

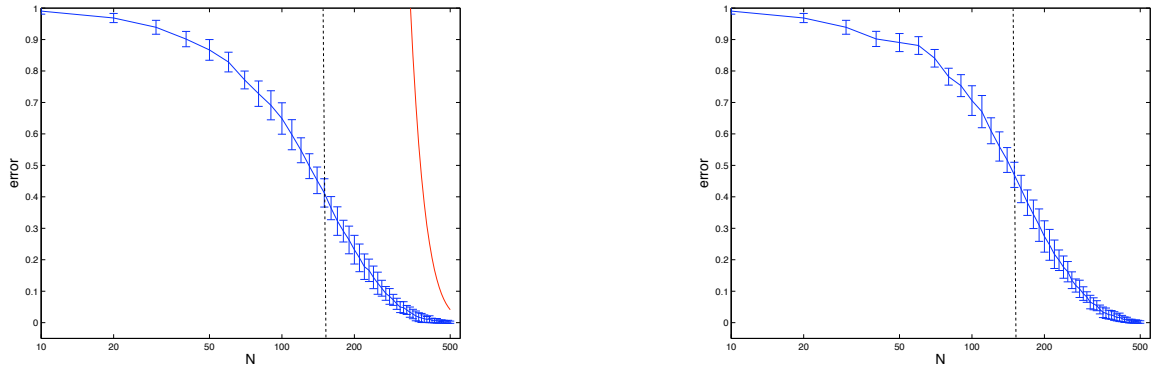


Figure 1: Estimates of the probability of error for various N, for the case $p_0 = (0.9, 0.1)$, $p_1 = (0.8, 0.2)$, $M = 1000$. The two plots correspond to the cases of known and unknown distributions, respectively. The red line represents the upper bound as established by Theorem 1.

and

$$\textit{If } \exists\, \varepsilon > 0, \textit{ such that } \lim_{M,N\to\infty} M2^{-N(D(p_1,p_0)+\varepsilon)} = +\infty \quad \textit{then } \lim_{M,N\to\infty} \tilde{e}(M,N) = 1. \quad (21)$$

The proof uses the same large deviations techniques as the proof of Theorem 2 but is slightly more complex due to the fact that the rewards are not independent anymore. The proof appears in Section 6.

4 Simulations

We now provide simulations that show Theorems 1, 2 and 3 in action.

We generated $M = 1000$ binary sequences with probabilities $p_0 = (0.9, 0.1)$ and $p_1 = (0.8, 0.2)$ for the background and the target, respectively. We varied the number N from 10 to 500, and we observed the probability of error decreasing to zero. We performed the random experiment 100 times for each value of N. The procedure was replicated 20 times in

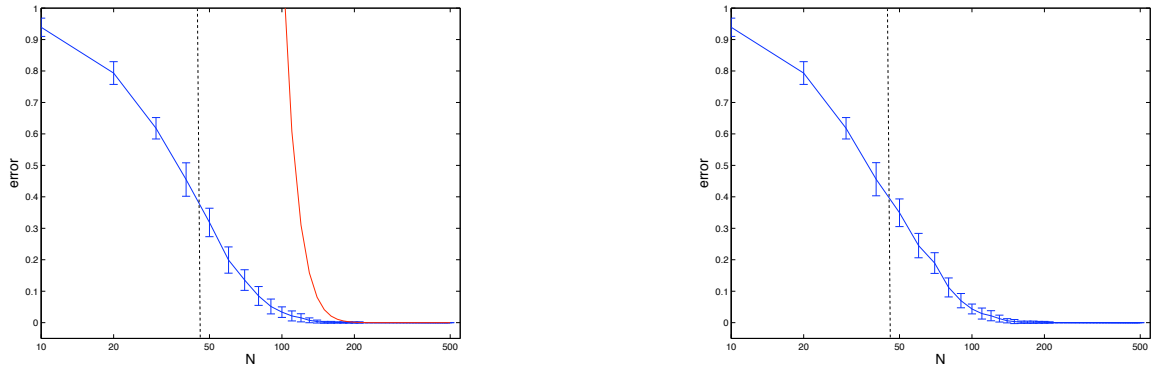


Figure 2: Estimates of the probability of error for various N, for the case $p_0 = (0.9, 0.1)$, $p_1 = (0.7, 0.3)$, $M = 1000$. The two plots correspond to the cases of known and unknown distributions, respectively. The red line represents the upper bound as established by Theorem 1.

order to compute error bars. The plots in Figure 1 show the (estimated) probability of error versus N, for the two maximum likelihood detectors (known and unknown distributions, respectively), along with 1 standard deviation error bars. As expected, the error for the case of unknown distributions is somewhat higher, as there is an additional error due to the inaccuracy in estimating the two distributions. The KL divergence is $D(p_1, p_0) \simeq 0.064$. The dashed line shows the phase transition "boundary", i.e., the value of N such that $M = 2^{ND(p_1,p_0)}$. For $M = 1000$, this value is equal to 155.5. For the known distributions plot, the red line corresponds to the upper bound established by Theorem 1, and it is equal to $1000(0.98)^N$. Similar plots for the case $p_0 = (0.9, 0.1)$ and $p_1 = (0.7, 0.3)$ are shown in Figure 2. As expected, the error curves of Figure 1 are higher than the ones in Figure 2, since the former detection case is "harder" than the latter. The phase transition boundary is depicted in Figure 2 with the dashed line at value $N = 44.9$. The upper bound of Theorem 1 is given by $1000(0.9349)^N$.

5 Conclusions

We have considered a statistical model with $M+1$ sequences of independent random variables, each of length N. All random variables have the same point mass function p_0 except for one sequence, the target, for which the common point mass function is p_1. The error of the maximum likelihood estimator for the target converges to 0 if there exists an $\varepsilon > 0$ such that $M2^{-N(D(p_1,p_0)-\varepsilon)} \to 0$, and it converges to 1 if there exists an $\varepsilon > 0$ such that $M2^{-N(D(p_1,p_0)+\varepsilon)} \to +\infty$. Moreover, when p_0 and p_1 are unknown, we are able to build an estimator of the target with the same performace; this allows us to study the important practical problem of outlier detection. We conjecture that these results can be generalized to the case of ergodic Markov chains, and we plan to report the more general results in a subsequent publication.

6 Proofs

Without loss of generality, we assume that the target line is line number 1.

Proof of Theorem 1:

$$
\begin{aligned}
e(M,N) &= \mathbb{P}\left(\max_{2\le m\le M+1}\sum_{n=1}^{N}\log\frac{p_1(X_m^n)}{p_0(X_m^n)} > \sum_{n=1}^{N}\log\frac{p_1(X_1^n)}{p_0(X_1^n)}\right) && (22)\\
&\le M\mathbb{P}\left(\sum_{n=1}^{N}\log\frac{p_1(X_2^n)}{p_0(X_2^n)} > \sum_{n=1}^{N}\log\frac{p_1(X_1^n)}{p_0(X_1^n)}\right) && (23)\\
&= M\mathbb{P}\left(\prod_{n=1}^{N}\left(\frac{p_1(X_2^n)p_0(X_1^n)}{p_0(X_2^n)p_1(X_1^n)}\right)^s > 1\right), \text{ for all } s>0 && (24)\\
&\le ME\left[\prod_{n=1}^{N}\left(\frac{p_1(X_2^n)p_0(X_1^n)}{p_0(X_2^n)p_1(X_1^n)}\right)^s\right] && (25)\\
&= M\left[E\left(\frac{p_1(X_2^1)p_0(X_1^1)}{p_0(X_2^1)p_1(X_1^1)}\right)^s\right]^N && (26)
\end{aligned}
$$

where (25) is due to the Markov inequality.

Let us define

$$f(s) \triangleq E\left[\left(\frac{p_1(X_2^1)p_0(X_1^1)}{p_0(X_2^1)p_1(X_1^1)}\right)^s\right] \quad \text{and} \quad g(s) \triangleq \ln f(s) \tag{27}$$

One can check that $f'(1/2) = g'(1/2) = 0$. Moreover, using Hölder's inequality, Grimmett et. al. (1992), it is easy to show that, for any $s, t > 0$ and $0 \leq \alpha \leq 1$,

$$E\left[\left(\frac{p_1(X_2^1)p_0(X_1^1)}{p_0(X_2^1)p_1(X_1^1)}\right)^{\alpha s+(1-\alpha)t}\right] \leq \left(E\left[\left(\frac{p_1(X_2^1)p_0(X_1^1)}{p_0(X_2^1)p_1(X_1^1)}\right)^s\right]\right)^{\alpha} \left(E\left[\left(\frac{p_1(X_2^1)p_0(X_1^1)}{p_0(X_2^1)p_1(X_1^1)}\right)^t\right]\right)^{1-\alpha}. \tag{28}$$

By taking the log on both sides, we deduce that g is a convex function of s. Hence, it achieves its minimum value at $s = 1/2$ (therefore, f achieves its minimum value at $s = 1/2$). This leads to the tightest upper bound in (26), i.e.,

$$e(M, N) \leq M f^N(\frac{1}{2}) = M\left(\sum_x \sqrt{p_0(x)p_1(x)}\right)^{2N}. \tag{29}$$

■

In order to prove Theorems 2 and 3, we start with two technical lemmas that will be useful later on.

Lemma 1 *Let U and R be two rvs, and $y \in \mathbb{R}$. Then, for any $\varepsilon > 0$,*

$$\mathbb{P}(U > y + \varepsilon) - \mathbb{P}(R < -\varepsilon) \leq \mathbb{P}(U + R > y) \leq \mathbb{P}(U > y - \varepsilon) + \mathbb{P}(R > \varepsilon) \tag{30}$$

Proof of Lemma 1:

$$\begin{aligned} \mathbb{P}(U + R > y) &= \mathbb{P}(U + R > y, R \leq \varepsilon) + \mathbb{P}(U + R > y, R > \varepsilon) && (31) \\ &\leq \mathbb{P}(U > y - \varepsilon) + \mathbb{P}(R > \varepsilon) && (32) \end{aligned}$$

and

$$\begin{aligned} I\!P(U+R\leq y) &= I\!P(U+R\leq y, R<-\varepsilon)+I\!P(U+R\leq y, R\geq -\varepsilon) \quad (33)\\ &\leq I\!P(U\leq y+\varepsilon)+I\!P(R<-\varepsilon) \quad (34)\end{aligned}$$

which allows us to obtain the lower bound by computing the complementary event. ■

Lemma 2 *Let $(V_1^N,\ldots,V_M^N)$ be a sequence of M independent, identically distributed, discrete random variables. Moreover, assume that the following large deviation property holds for some $z\in I\!R$,*

$$I\!P(V_1^N>z)\doteq 2^{-NI(z)},\ \textit{where } I(z)>0 \textit{ and } a_N\doteq b_N \Leftrightarrow \lim_{N\to+\infty}\frac{1}{N}\log\frac{a_N}{b_N}=0. \quad (35)$$

Then, if

$$\exists\,\varepsilon>0 \quad s.t. \quad \lim_{M,N\to+\infty} M2^{-N(I(z)-\varepsilon)}=0, \quad then \quad \lim_{M,N\to+\infty} I\!P(\max_{1\leq m\leq M} V_m^N>z)=0. \quad (36)$$

Also, if

$$\exists\,\varepsilon>0 \quad s.t. \quad \lim_{M,N\to+\infty} M2^{-N(I(z)+\varepsilon)}=+\infty, \quad then \quad \lim_{M,N\to+\infty} I\!P(\max_{1\leq m\leq M} V_m^N>z)=1. \quad (37)$$

Proof of Lemma 2: Let $\varepsilon>0$ be arbitrarily small. Then, there exists $N_0(\varepsilon)>0$ such that

$$\forall\, N>N_0, \quad \left|\frac{1}{N}\log\left(\frac{I\!P(V_1^N>z)}{2^{-NI(z)}}\right)\right|<\varepsilon. \quad (38)$$

To prove the first part, we start with the following claim:

$$(\exists\,\varepsilon'>0)\ (\forall\, N'>0)\ (\exists\, N>N'):\quad I\!P(\max_{1\leq m\leq M(N)} V_m^N>z)>\varepsilon'.$$

Then, using the union bound, we obtain

$$(\exists\,\varepsilon'>0)\ (\forall\,N'>0)\ (\exists\,N>N'):\quad \sum_{m=1}^{M(N)} I\!P(V_m^N>z)=M(N)I\!P(V_1^N>z)>\varepsilon'. \tag{39}$$

By picking $N'>N_0(\varepsilon)$, (39) becomes

$$(\exists\,\varepsilon'>0)\ (\forall\,N'>N_0(\varepsilon))\ (\exists\,N>N'):\quad M2^{-N(I(z)-\varepsilon)}>\varepsilon'.$$

Hence, $M2^{-N(I(z)-\varepsilon)}$ does not converge to zero for any $\varepsilon>0$, as required.

To prove the second part, we first assume that $N>N_0(\varepsilon)$, as above. Then,

$$\begin{aligned}
I\!P(\max_{1\le m\le M} V_m^N>z) &= 1-I\!P(\max_{1\le m\le M} V_m^N\le z) && (40)\\
&= 1-I\!P^M(V_1^N\le z) && (41)\\
&= 1-2^{M\log(1-I\!P(V_1^N>z))} && (42)\\
&\ge 1-2^{-MI\!P(V_1^N>z)} && (43)\\
&\ge 1-2^{-M2^{-N(I(z)+\varepsilon)}}, && (44)
\end{aligned}$$

where the first inequality is a consequence of the inequality $\log(1-x)\le -x$, and the second inequality arises from (38). Note that (44) is true for any arbitrary $\varepsilon>0$. Hence, if there exists $\varepsilon>0$ such that $M2^{-N(I(z)+\varepsilon)}\to+\infty$, then necessarily $I\!P(\max_{1\le m\le M} V_m^N>z)\to 1$. This concludes the proof of the second part, and the proof of the lemma. ■

We are now ready for

Proof of Theorem 2:

$$\begin{aligned}
e(M,N) &= I\!P\left(\max_{2\le m\le M+1}\frac{1}{N}\sum_{n=1}^{N}\log\frac{p_1(X_m^n)}{p_0(X_m^n)}>\frac{1}{N}\sum_{n=1}^{N}\log\frac{p_1(X_1^n)}{p_0(X_1^n)}\right) && (45)\\
&= I\!P(U_M^N+R_N>D(p_1,p_0)),\quad \text{where} && (46)
\end{aligned}$$

$$U_M^N = \max_{2\le m\le M+1} \frac{1}{N}\sum_{n=1}^{N} \log \frac{p_1(X_m^n)}{p_0(X_m^n)} \quad \text{and} \tag{47}$$

$$R_N = D(p_1,p_0) - \frac{1}{N}\sum_{n=1}^{N} \log \frac{p_1(X_1^n)}{p_0(X_1^n)} \tag{48}$$

From the law of large numbers, $R_N \to 0$ in probability. Hence, for all $\eta > 0$ and $\alpha > 0$, and for N sufficiently large, using Lemma 1,

$$I\!P(U_M^N > D(p_1,p_0)+\eta) - \alpha \le e(M,N) \le I\!P(U_M^N > D(p_1,p_0)-\eta) + \alpha \tag{49}$$

Let us define

$$V_m^N \triangleq \frac{1}{N}\sum_{n=1}^{N} \log \frac{p_1(X_m^n)}{p_0(X_m^n)} \tag{50}$$

Now, using Sanov's theorem, Dembo at. al.(1998),

$$I\!P(V_2^N \ge D(p_1,p_0)) \doteq 2^{-ND(p_1,p_0)} \tag{51}$$

Indeed,

$$I\!P(V_2^N \ge D(p_1,p_0)) \doteq 2^{-ND(p^*,p_0)} \tag{52}$$

where

$$D(p^*,p_0) = \inf_{p\in\mathcal{C}} D(p,p_0), \text{ with } \mathcal{C} = \{p; E_p \log \frac{p_1}{p_0} \ge D(p_1,p_0)\}, \tag{53}$$

and for $p \in \mathcal{C}$,

$$D(p,p_0) = E_p \log \frac{p}{p_0} = D(p,p_1) + E_p \log \frac{p_1}{p_0} \tag{54}$$

$$\ge D(p,p_1) + D(p_1,p_0) \ge D(p_1,p_0). \tag{55}$$

Now, by continuity of the rate function, there exists $\varepsilon > 0$ such that

$$I\!P(V_2^N > D(p_1,p_0)-\eta) \doteq 2^{-N(D(p_1,p_0)-\varepsilon)} \tag{56}$$

and there exists $\varepsilon' > 0$ such that

$$I\!P(V_2^N > D(p_1, p_0) + \eta) \doteq 2^{-N(D(p_1,p_0)+\varepsilon')} \tag{57}$$

Finally, since

$$U_M^N = \max_{2 \le m \le M+1} V_m^N \tag{58}$$

and the rvs $V_2^N, \ldots, V_{M+1}^N$ are iid, we obtain the required result from Lemma 2. ■

Proof of Theorem 3: We proceed along the same lines as for the proof of Theorem 2.

$$\begin{aligned}
\tilde{e}(M,N) &= I\!P\left(\max_{2\le m\le M+1} \frac{1}{N}\sum_{n=1}^{N} \log\frac{\hat{p}_m(X_m^n)}{\hat{p}_{(m)}(X_m^n)} > \frac{1}{N}\sum_{n=1}^{N}\log\frac{\hat{p}_1(X_1^n)}{\hat{p}_{(1)}(X_1^n)}\right) && (59)\\
&\le I\!P(U_M^N + R_M^N > D(p_1,p_0)), \quad \text{with} && (60)\\
U_M^N &= \max_{2\le m\le M+1} \frac{1}{N}\sum_{n=1}^{N}\log\frac{\hat{p}_m(X_m^n)}{p_0(X_m^n)} && (61)\\
R_M^N &= A_M^N + B^N + C^N && (62)\\
A_M^N &= \max_{2\le m\le M+1} \frac{1}{N}\sum_{n=1}^{N}\log\frac{p_0(X_m^n)}{\hat{p}_{(m)}(X_m^n)} && (63)\\
B^N &= \frac{1}{N}\sum_{n=1}^{N}\log\frac{\hat{p}_{(1)}(X_1^n)}{p_0(X_1^n)} && (64)\\
C^N &= D(p_1,p_0) - \frac{1}{N}\sum_{n=1}^{N}\log\frac{\hat{p}_1(X_1^n)}{p_0(X_1^n)} && (65)
\end{aligned}$$

For all $\eta > 0$, from Lemma 1,

$$\tilde{e}(M,N) \le I\!P(U_M^N > D(p_1,p_0) - \eta) + I\!P(R_M^N > \eta). \tag{66}$$

Let

$$V_m^N = \frac{1}{N}\sum_{n=1}^{N}\log\frac{\hat{p}_m(X_m^n)}{p_0(X_m^n)}, \quad 2 \le m \le M+1. \tag{67}$$

Using Sanov's theorem,

$$I\!P(V_2^N \geq D(p_1, p_0)) \doteq 2^{-ND(p_1,p_0)}. \tag{68}$$

Indeed,

$$I\!P(V_2^N \geq D(p_1, p_0)) \doteq 2^{-ND(p^*,p_0)}, \tag{69}$$

where

$$D(p^*, p_0) = \inf_{p \in \mathcal{C}} D(p, p_0), \text{ with } \mathcal{C} = \{p; E_p \log \frac{p}{p_0} \geq D(p_1, p_0)\}. \tag{70}$$

And for $p \in \mathcal{C}$,

$$D(p, p_0) = E_p \log \frac{p}{p_0} \geq D(p_1, p_0). \tag{71}$$

Now, by continuity of the rate function, there exists $\varepsilon > 0$ such that

$$I\!P(V_2^N > D(p_1, p_0) - \eta) \doteq 2^{-N(D(p_1,p_0)-\varepsilon)}. \tag{72}$$

To show that (66) approaches zero as $M, N \to \infty$ with $M2^{N(D(p_1,p_0)-\varepsilon)} \to 0$, it suffices to prove that $R_M^N \to 0$, since the term $I\!P(U_M^N > D(p_1, p_0) - \eta)$ of (66) goes to zero by virtue of (72), Lemma 2, and the fact that

$$U_M^N = \max_{2 \leq m \leq M+1} V_m^N, \tag{73}$$

and the rvs $V_2^N, \ldots, V_{M+1}^N$ are iid.

Using the law of large numbers, $C^N \to 0$ in probability. Also,

$$\begin{aligned} I\!P(A_M^N > \eta) &\leq M I\!P\left(\frac{1}{N}\sum_{n=1}^{N} \log \frac{p_0(X_2^n)}{\hat{p}_{(2)}(X_2^n)} > \eta\right) &(74)\\ &\leq M I\!P\left(\max_{1 \leq n \leq N} \log \frac{p_0(X_2^n)}{\hat{p}_{(2)}(X_2^n)} > \eta\right) &(75)\\ &\leq MN I\!P\left(\log \frac{p_0(X_2^1)}{\hat{p}_{(2)}(X_2^1)} > \eta\right) &(76) \end{aligned}$$

$$\leq \quad MN \max_x \mathbb{P}\left(\log \frac{p_0(x)}{\hat{p}_{(2)}(x)} > \eta\right) \tag{77}$$

$$= \quad MN \max_x \mathbb{P}(\hat{p}_{(2)}(x) < 2^{-\eta} p_0(x)) \tag{78}$$

$$\leq \quad MN \max_x 2^{-N(M-1)I(x,\eta)} = MN 2^{-N(M-1)J(\eta)} \tag{79}$$

where $I(x,\eta) > 0$ is a rate function, and $J(\eta) = \min_x I(x,\eta)$. The last inequality comes from the fact that $\hat{p}_{(2)}(x) \to p_0(x)$ in probability. A similar argument shows that $B^N \to 0$ in probability as well.

Note that the result would still hold if we replaced $\hat{p}_{(m)}$ with $\hat{p}$, i.e., with the empirical distribution over the full data. ■

References

[1] Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*. Springer Verlag, 1998.

[2] D. Geman and B. Jedynak. An active testing model for tracking roads from satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14, January 1996.

[3] G.R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford Science Publications, 1992.

[4] Alan L. Yuille and James M. Coughlan. Fundamental limits of bayesian inference: Order parameters and phase transitions for road tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):160–173, February 2000.

[5] Alan L. Yuille, James M. Coughlan, Yingnian Wu, and Song Chun Zhu. Order parameters for detecting target curves in images: When does high level knowledge help? *International Journal of Computer Vision*, 41:9–33, 2001.

Chapter 10

Learning to match: deriving optimal template-Matching algorithms from probabilistic image models

Int J Comput Vis (2010) 88: 189–213
DOI 10.1007/s11263-009-0258-5

Learning to Match: Deriving Optimal Template-Matching Algorithms from Probabilistic Image Models

Camille Vidal · Bruno Jedynak

Received: 26 July 2008 / Accepted: 26 May 2009 / Published online: 19 June 2009

Abstract Finding correspondences between images by template matching is a common problem in image understanding. Although a variety of solutions have been proposed, most of them rely on the arbitrary choice of a template and a matching function. Often, different cost functions lead to different results, and the choice of a good cost for a specific application remains an art. Statistical models on the other hand, allow us to derive optimal learning and matching algorithms from modeling assumptions using likelihood maximization principles. The key contribution of this paper is the development of a statistical framework for learning what function to optimize from training examples. We present a family of statistical models for grayscale images, which allow us to derive optimal template-matching algorithms. The intensity at each pixel is described by a random variable whose distribution is encoded by a deformable template. Firstly, we assume the intensity distribution to be Gaussian and derive an intensity-matching algorithm, which is a generalization of the classical sum-of-squared differences. Then, we introduce a hidden segmentation variable in the probabilistic model and derive a segmentation-matching algorithm that can handle photometric variations. Both models are exemplified on the automatic detection of anatomical landmarks in brain Magnetic Resonance Images.

Keywords Statistical learning · Deformable template · Image registration · Anatomical landmark detection

C. Vidal (✉) · B. Jedynak
Johns Hopkins University, 3400 N Charles Street, Baltimore, MD 21218, USA
e-mail: camille.vidal@jhu.edu

1 Introduction

Image registration and matching refer to the problems of finding a transformation f that puts, respectively, two images or two sets of points into correspondence. These problems are central to numerous applications in several areas of pattern analysis, such as computer vision and medical imaging. For instance, an early application of image registration is image stitching, which refers to the problem of building a panorama of a natural scene from a collection of images of the scene (Szeliski 2006). More recently, feature matching has been one of the key technologies behind advances in object recognition based on extracting and matching scale-space invariant features from a collection of images, e.g., Lowe (2003), Dalal and Triggs (2005).

In medical imaging, one of the objectives is to build computational models of anatomical structures from a collection of images of different individuals (Grenander and Miller 1998). Image registration is central to the estimation of these models, firstly because the images are often acquired under different conditions, which means that the images need to be aligned before analysis. In addition, with the recent advancements in computational anatomy, the amount of deformation between a template image and an instance image is used as a way to build metrics and statistical models on a collection of images, e.g., Qiu et al. (2007).

Several registration and matching algorithms have been proposed and tested on different image analysis problems achieving great performance. Most of these algorithms find an optimal transformation by minimizing an energy function. However, we will argue below that different energy functions lead to different results, and the choice of a "good" energy function often depends on the application. There is a need of developing a unifying framework for image regis-

tration and a generic method to derive matching and registration algorithms.

1.1 Registration by Energy Minimization

Most of the proposed methods for image registration rely on an energy minimization formulation. The template or image source, denoted by x_0 is deformed by f, so that it looks alike with the target image x. The energy function used for image matching or image registration,

$$\mathcal{J}(x, x_0, f, \gamma) = \mathcal{A}(x, x_0, f) + \gamma \mathcal{R}(f), \tag{1}$$

is usually composed of two terms related by a weighting factor by a $\gamma \in \mathbb{R}$. The data term $\mathcal{A}$ measures the similarity between the deformed template $x_0 \circ f^{-1}$ and the target image x. The regularization term $\mathcal{R}$ is used to reduce the set of possible deformations and ensure uniqueness of the solution by, for instance, penalizing non-smooth or large deformations.

The matching result intrinsically depends on the choice of the energy function $\mathcal{J}$. The solution of this optimization problem minimizes the trade-off between matching the deformed template and the target image and satisfying the regularization constraint. Changing the data attachment term or the regularization term generally modifies the solution of the problem. Most of the time these choices are made arbitrarily. Although numerous possibilities have been explored, e.g., Zitová and Flusser (2003), Goshtasby et al. (2003), Szeliski (2006), it is not known in general how to choose the appropriate cost-function. We summarize below the most commonly used data attachment and regularization terms.

1.1.1 The Data Attachment Term

Similarity measures are typically classified into two categories: feature-based and image-based.

The first group is based on sparse feature matching, where matching generally starts with extracting the adequate features from the source and target images. Ideally these features should be invariant to scaling and other usual transformations. The solution to the registration problem is the deformation that minimizes the distance between the position of the features in the deformed image and their position in the target image, while fulfilling the chosen regularization constraint. The main advantage of this method is its low computational load due to the sparseness of the information, which allows its usage in real-time applications. On the other hand, precisely because the information to perform the matching is sparse, in regions with low level of information the matching will probably be less accurate. Nevertheless, this type of similarity function performs well in the presence of numerous matching features and for relatively simple deformation models.

The second category of similarity functions, so-called image-based measures, compares the intensity, in the simplest case, of the deformed template $x_0 \circ f^{-1}$ to the intensity of the target image x. As opposed to the feature-based measures, this type of cost function relies on a dense comparison between the deformed template and the image. Although the computational load is higher, this type of matching cost is more appropriate to local non-rigid deformations. Classical similarity functions are the absolute intensity difference (Barnea and Silverman 1972), the sum of squared intensity difference (SSD) (e.g., Friston et al. 1995; Ashburner and Friston 1999) or the correlation coefficient (Pratt 1974). Additional cost functions are based on other functions of the image such as local Fourier coefficients (Glasbey and Mardia 2001), edge distribution (Li et al. 1995), to cite only a few of them. Finally other image-matching functions are based on information theoretic criteria, such as comparing the intensity distribution of the source and the target using joint entropy (Studholme et al. 1995; Collignon et al. 1995) or mutual information (Collignon et al. 1995; Viola 1995; Wells et al. 1996; Maes et al. 1997).

1.1.2 The Regularization Term

The choice of the regularization term is usually motivated by the type of deformations that need to be considered in the problem at hand. If a global alignment is sufficient, rigid or affine transformations will be favored as it is defined by a small number of parameters. On the other hand, these transformations are generally not "flexible" enough to model subtle deformations, such as the ones observed in medical imaging.

Non-rigid deformation models are often preferred to model subtle changes in these images. There exist numerous representations for non-rigid (and non-affine) deformations. Low-dimensional representations such as free-form deformations, or more generally spline-based deformations, are parameterized by the displacement of control points (Bookstein 1992; Joshi and Miller 2000; Rohr et al. 2001). The deformation is obtained by interpolating the control point displacements to the rest of the image using smooth basis functions. The choice of the basis function influences significantly the properties of the resulting deformation (Wahba 1990; Bookstein 1989; Arad et al. 1994; Rohr 2001).

Alternative approaches model the image as a physical continuum, whose deformation follows a mechanic model such as an elastic or a fluid deformation. In that case, the deformation field (or the velocity field) is the solution of a Partial Differential Equation (PDE). Examples of image registration using these models can be found in e.g., Bajcsy and Kovačič (1989), Davatzikos (1997), Bro-Nielsen and Gramkow (1996), Lester et al. (1999).

Finally the weight parameter γ in (1) is most of the time manually tuned. Sometimes γ is modified as the optimization proceeds in order to favor first rigid deformations and then allow for non-rigid deformations that provide a more accurate matching result. It is generally believed that such techniques prevent the optimization algorithm from getting trapped in local minima.

1.2 Statistical Models for Image Registration

Although many registration algorithms have been proposed, the design of registration algorithms for a new task or modality remains an art. In general, it is not clear what cost function should be used. The choice is frequently based on intuition or trial and error, depending on the specific task at hand.

Viola (1995), Roche et al. (2000), Glasbey and Mardia (2001) studied the case of intensity images with limited changes of illumination from a statistical point of view. Assuming that the noise between the template image and the target image is Gaussian, they showed that the maximum likelihood estimator of the deformation corresponds exactly to the deformation minimizing the sum of squared differences.

Recently, there have been several works on developing generative statistical models for different tasks such as image classification (Allassonnière et al. 2007) or image segmentation (Levin and Weiss 2006). They learn the model parameters from learning samples and estimate by likelihood maximization the variable of interest, respectively the class of the image or the segmentation. Our work follows similar principles and applies them to the case of image matching, which means that the variable of interest is the deformation that maps the template onto a new image.

1.3 Paper Contributions

We present different examples of model for normalized gray-level images and for gray-level images with intensity variations (i.e. coming from different acquisition protocols). Using maximum likelihood principles, we derive simple algorithms for image matching based on the modeling assumptions and provide the corresponding optimal matching function. Because the matching function is derived from the generative model following maximum likelihood principles, it is possible to understand how the modeling assumptions relate to the final cost function. In all cases the derived matching functions are very intuitive and correspond in some cases to well-known energy functions such as the sum-of-squared differences.

We illustrate the different models on the specific problem of landmark detection in brain MRI. The landmark detection task consists of localizing a set of anatomical landmarks defined by an expert and manually located on training images. Using the technique proposed in this paper, we have been able to derive generic adaptive algorithms for the simultaneous detection of one or more landmarks. As opposed to other existing methods for landmark detection (Thirion 1996; Frantz et al. 2000; Wörz and Rohr 2006), the proposed algorithm adapts automatically to all types of landmarks for which a training set can be obtained.

2 Anatomical Landmark Detection

An anatomical landmark is a point in the image that corresponds to a specific part of the anatomy (Bookstein 1992; Thirion 1996; Frantz et al. 2000). They are defined by an expert and commonly used to set correspondences between images. We denote by $y \in \mathbb{R}^{dK}$ a vector containing the position of K landmarks in an image. The position of the landmarks in the template is fixed and denoted by $\bar{y} \in \mathbb{R}^{dK}$.

2.1 Landmark Detection as a Local Registration Problem

We model a landmarked image as the result of a bijective deformation acting on a template x_0, such that the landmark locations in the template $\bar{y}$ are mapped onto y in the target image, i.e. $f(\bar{y}) = y$. To simplify the problem, we assume that the deformation f is fully characterized by the correspondences of the landmarks in the template and in the image. Therefore, when $\bar{y}$ is fixed, it is equivalent to estimate the location y or to estimate the deformation that maps the template onto the target image. We formulate the landmark detection as an image matching problem:

$$\hat{f} = \arg\max_{f \in \mathcal{F}} \mathcal{A}(x, x_0, f) + \gamma \mathcal{R}(f) \quad \text{and} \quad \hat{y} = \hat{f}(\bar{y}). \tag{2}$$

The deformation $f : \mathbb{R}^d \to \mathbb{R}^d$ is parametrized by the landmark displacements from the reference location $\bar{y}$ to the image location y. Using spline interpolation, the displacements of the landmarks is interpolated to the rest of the image support. The resulting deformation depends on the choice of the interpolation function used. Therefore we reduce the set of possible deformations by fixing κ the interpolation function of the spline-based deformation. It can be shown that there exists a unique deformation that satisfies the landmark matching constraint $f(\bar{y}) = y$ and that can be written:

$$\forall t, \quad f(t) = t + \sum_{k=1}^{K} \kappa(t, \bar{y}_k)\beta_k, \quad \text{with } \beta_k \in \mathbb{R}^d. \tag{3}$$

According to Mercer's theorem, it is equivalent to fix the basis function κ or a regularization term of the form $\|f - Id\|_{\mathcal{F}}$, with $\mathcal{F}$ a Hilbert space of smooth functions of $\mathbb{R}^d$. For simplicity, we fix arbitrarily the deformation

model. It would be interesting though in future work to include the deformation model as a parameter of the statistical model to be learnt from the training set.

We choose for our application to landmark detection to work with a Gaussian kernel of variance σ^2:

$$\forall t, \quad \kappa(t, \bar{y}_k) = \exp\left(-\frac{\|t - \bar{y}_k\|^2}{2\sigma^2}\right). \tag{4}$$

The main advantage of this kernel over the commonly used Thin-Plate Spline approach (Bookstein 1989) is that the deformation has a local support, controlled by the variance of the kernel. Other locally defined spline models may be used such as B-spline or Clamped Plate Spline (Wahba 1990; Twining et al. 2002).

2.2 Landmark Detection

We propose to take advantage of a training set of annotated images, in which the landmarks have been manually positioned. The proposed method consists of learning the model parameters from a training set. Then, the estimated model is used to detect landmarks in new images.

We denote by $\theta \in \Theta$ the model parameters, $x_1^N \in \mathbb{R}^{SN}$ the training set of N images, $y_1^N \in \mathcal{Y} \subset \mathbb{R}^{dKN}$ the location of the landmarks in the training images and $x \in \mathbb{R}^S$ a new image. The model parameters are estimated by likelihood maximization

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \ell(x_1^N, y_1^N; \theta). \tag{5}$$

As for the landmark detection, it is carried out by maximizing the likelihood of a new image with respect to the landmark locations, while using the previously learnt model parameters:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \ell(x, y; \hat{\theta}). \tag{6}$$

3 Deformable Intensity Model

3.1 The Gaussian Image Model

Roche et al. (2000), Glasbey and Mardia (2001) propose to build a simple statistical model for registering two images. The target image x is modeled as the result of the action of a random bijective deformation f applied to the template image x_0, corrupted by an additive Gaussian noise. Denoting by Λ the support of the target image and s a pixel (or voxel) included in Λ,

$$\forall s \in \Lambda, \quad x(s) = x_0(f^{-1}(s)) + \epsilon(s), \tag{7}$$

with $\epsilon(s) \sim \mathcal{N}(0, \tau^2)$, the centered Gaussian distribution of variance τ^2, and $x(s)$ the real random variable representing the image intensity at pixel s, and $x_0(f^{-1}(s))$ the intensity in the template at pixel $t = f^{-1}(s)$. In terms of probability distribution, it means that the intensity at pixel s, given the registering deformation f, follows a Gaussian distribution, whose mean is given by the intensity at the corresponding location of the template. Assuming the intensity at each pixel is independent given the deformation f, the whole image likelihood is:

$$p(x|f) \propto \exp\left(-\frac{\sum_{s \in \Lambda} |x(s) - x_0(f^{-1}(s))|^2}{2\tau^2}\right). \tag{8}$$

In this formulation, the deformation f and the image x are random variables, while x_0 the template image and the noise variance τ^2 belong to the parameters of the model. Therefore, given two images, a source image x_0 and a target image x, the registration of x_0 onto x consists of finding the deformation f that maximizes the conditional likelihood of the observation x. The best deformation, in terms of likelihood, is given by:

$$\hat{f} = \arg\max_{f \in \mathcal{F}} \ln p(x|f), \tag{9}$$

$$= \arg\min_{f \in \mathcal{F}} \sum_{s \in \Lambda} |x(s) - x_0(f^{-1}(s))|^2. \tag{10}$$

The maximum likelihood estimator $\hat{f}$ corresponds to the deformation that minimizes the sum of squared intensity difference (SSD) of the two images, as originally defined in Barnea and Silverman (1972). SSD has since then been broadly used for image matching and tracking in video sequences, and is considered as a benchmark for image matching.

In what follows, we present a model which is closely related to the Gaussian image model and demonstrates with this simple example how to derive a landmark detection algorithm.

3.2 Description of the Generative Model

The generative model relies on the joint distribution of the observations and of the variable of interest. We have made the assumption in Sect. 2 that the deformation is parametrized by the landmark locations y, thus the joint probability by $p(x, y)$. The template x_0 is a parameter of the statistical model, to be estimated from the training data.

Using Bayes' formula, the joint probability of the image x and the location of the landmarks y is

$$p(x, y) = p(x|y)p(y). \tag{11}$$

As it is often the case in generative models for images, we assume statistical independence of the image intensities given the location of the landmarks such that the conditional

probability can be written as a product over all the pixels of the image support. Assuming that the image is defined on a finite grid $\Lambda \in \mathbb{R}^d$,

$$p(x, y) = \prod_{s \in \Lambda} p(x(s)|y)p(y). \tag{12}$$

In the above Gaussian model, the noise τ^2 is a global parameter of the model and is independent from the location in the image. Thus the template or source image is a deterministic function defined on Λ_T, a finite grid of $\mathbb{R}$. In our approach, we choose to work with probabilistic templates, because we believe that the deformations defined by few landmarks are not "flexible" enough to model the geometric variability of real images. Probabilistic templates contain more information and allow us to capture both the photometric and the geometric variations, while working with a simple deformation model. We propose to model the intensity value as a Gaussian distribution whose mean and variance depends on the pixel location:

$$\forall s, \quad x(s)|y \sim \mathcal{N}(x_0(f_y^{-1}(s)), \tau_0^2(f_y^{-1}(s))), \tag{13}$$

with f_y the deformation that maps the landmarks of the template $\bar{y}$ to y in the image.

It means that the template contains at each pixel of Λ_T an intensity value and a standard deviation. As a consequence, the likelihood of an image is similar to the expression derived from the Gaussian model (8), except that the intensity variance depends on the pixel location:

$$\ell(x, y) = -\sum_{s \in \Lambda} \log \tau_0^2(f_y^{-1}(s)) + \frac{(x(s) - x_0(f_y^{-1}(s)))^2}{2\tau_0^2(f_y^{-1}(s))} - \sum_{s \in \Lambda} \frac{1}{2} \log 2\pi + \log p(y). \tag{14}$$

The log-likelihood of an image increases when the intensity observed in the image corresponds to the one contained in the deformed template. The weight of each pixel varies depending on its position in the image. Regions with lower intensity variance in the template have more importance than the regions with larger variance.

In order to generate images using this model, one first randomly samples a grayscale image from the Gaussian distribution $\mathcal{N}(x_0(t), \tau_0^2(t))$. The landmark position is sampled from $p(y)$ and used to determine the deformation f_y. The final image is obtained by deforming the randomly sampled grayscale image by f_y. The landmarks of the template are by construction mapped to the position y in the final image.

3.3 Model Selection Using a Training Set

Model selection consists of learning the parameters θ of the deformable model from the training set of annotated images (x_1^N, y_1^N). The model has two sets of parameters: the template parameters, for all t, $x_0(t)$ and $\tau_0^2(t)$, and the landmark prior distribution $p(y)$. The training images are considered as independent samples of $p(x, y)$. Thus, the likelihood of the training set:

$$\ell(x_1^N, y_1^N; \theta) = \sum_{i=1}^{N} \ell(x^{(i)}|y^{(i)}) + \sum_{i=1}^{N} p(y^{(i)}). \tag{15}$$

The likelihood function is a sum of two independent terms, therefore the optimization with respect to the template and the estimation of the prior distribution of the landmarks can be performed independently.

3.3.1 *Direct Estimation of the Deformable Template*

The template is learned by likelihood maximization with respect to (x_0, τ_0^2):

$$\ell(x_1^N|y_1^N; x_0, \tau_0^2) = \sum_{i=1}^{N} \sum_{s \in \Lambda_i} \ln p(x^{(i)}(s)|y^{(i)}). \tag{16}$$

Using the deformable model assumption,

$$x^{(i)}(s)|y^{(i)} \sim \mathcal{N}(x_0(t), \tau_0^2(t)) \quad \text{with } t = f_{y^{(i)}}^{-1}(s). \tag{17}$$

We denote by $\pi(x, t)$ the probability density for the intensity value x at t. Thus,

$$\ell(x_1^N|y_1^N; x_0, \tau_0^2) = \sum_{i=1}^{N} \sum_{s \in \Lambda_i} \ln \pi(x^{(i)}(s), f_{y^{(i)}}^{-1}(s)). \tag{18}$$

Because the deformation $f_{y^{(i)}}^{-1}$ depends on the image, it is not possible to change the order of sums. In consequence the estimation of the template parameters is a complex joint estimation problem. We propose to approximate the likelihood function (18) by performing a change of variable. The sum over the pixels of the image is approximated by an integral over the support of the image.[1]

$$\ell(x_1^N|y_1^N; x_0, \tau_0^2) \approx \sum_{i=1}^{N} \int_{\mathbb{R}^d} \ln \pi(x^{(i)}(s), f_{y^{(i)}}^{-1}(s)) ds. \tag{19}$$

For each image i, we perform the change of variable $s = f_{y^{(i)}}(t)$, and denote by $|J_{f_{y^{(i)}}}(t)|$ the absolute value of the deformation Jacobian at t.

$$\ell(x_1^N|y_1^N, x_0, \tau_0^2) = \sum_{i=1}^{N} \int_{\mathbb{R}^d} \ln \pi(x^{(i)}(f_{y^{(i)}}(t)), t)|J_{f_{y^{(i)}}}(t)| dt. \tag{20}$$

[1]For sake of simplicity, we assume that all the images are defined on $\mathbb{R}^d$ padding them with zeros and using linear interpolation if necessary

Finally, we approximate the likelihood by exchanging the order of the sum and the integral. After discretization of the integral:

$$\ell(x_1^N|y_1^N; x_0, \tau_0^2) = \sum_{t\in\Lambda_T}\sum_{i=1}^{N} \ln \pi(x^{(i)}(f_{y^{(i)}}(t)), t)|J_{f_{y^{(i)}}}(t)|. \quad (21)$$

The above approximation of the likelihood function will appear regularly in the estimation of the model. From now on we will refer to it as the "approximated integral change of variable". This approximation allows us to transform the joint optimization with respect to all the pixel parameters in as many independent problems as pixels in the finite grid Λ_T. The likelihood optimization with respect to $(x_0(t), \tau_0^2(t))$ becomes separable. The computation of (21) requires to interpolate the grayscale image to extend the definition of $x(f_{y^{(i)}}(t))$ to all possible values of t and y. Thus, the log-likelihood of the training set is:

$$\sum_{t\in\Lambda_T}\sum_{i=1}^{N}\left[-\frac{1}{2}\ln\tau^2(t) - \frac{|x(f_{y^{(i)}}(t)) - x_0(t)|^2}{2\tau^2(t)}\right]|J_{f_{y^{(i)}}}(t)|, \quad (22)$$

and its maximization at each pixel t, with respect to $x_0(t)$ and $\tau_0^2(t)$ has a closed form solution:

$$\hat{x}_0(t) = \frac{\sum_{i=1}^{N} x(f_{y^{(i)}}(t))|J_{f_{y^{(i)}}}(t)|}{\sum_{i=1}^{N}|J_{f_{y^{(i)}}}(t)|}, \quad (23)$$

$$\hat{\tau}_0^{\,2}(t) = \frac{\sum_{i=1}^{N}[x(f_{y^{(i)}}(t)) - x_0(t)]^2|J_{f_{y^{(i)}}}(t)|}{\sum_{i=1}^{N}|J_{f_{y^{(i)}}}(t)|}. \quad (24)$$

The Maximum Likelihood Estimator (MLE) is similar to the classical MLE of a Gaussian sample, except that each sample is weighted by the Jacobian of the corresponding transformation. If the Jacobian is locally equal to 1, it is locally equivalent to averaging the observed intensities, after registration of the training images.

3.3.2 Learning the Distribution of the Landmark Locations

Classical density estimation methods can be used to estimate the prior distribution of the landmarks in the image based on the training samples. As the number of landmarks increases and the size of sample stays limited, one might need to incorporate some regularization in the density estimation. In practice, in all the experiments presented in this paper, we did not incorporate any prior information.

3.4 Local Intensity Matching for Landmark Detection

We use the model learnt in the training phase to predict the location of the landmarks in a new image. The log-likelihood of a new grayscale image is

$$\ell(x|y; \hat{x}_0, \hat{\tau}_0^2) = -\frac{1}{2}\sum_{s\in\Lambda}\left[\ln 2\pi + \ln\hat{\tau}_0^2(f_y^{-1}(s)) + \frac{|x(s) - \hat{x}_0(f_y^{-1}(s))|^2}{\hat{\tau}_0^2(f_y^{-1}(s))}\right]. \quad (25)$$

We use the MLE to predict the location of the landmarks:

$$\hat{y} = \arg\max_y \ell(x|y; \hat{x}_0, \hat{\tau}_0^2). \quad (26)$$

3.4.1 Local Intensity Matching Algorithm

When using SSD for image matching, it is implicitly assumed that the noise parameter τ is constant throughout the template. Therefore all the image pixels have the same weight. Because the variance in the Deformable Intensity Model (DIM) varies depending on the location in the template, the pixels with lower variance have greater weight in the cost function than the pixels for which the intensity variance is large. Pixels around the landmarks generally correspond to regions of low variance. In consequence, the cost function focuses on matching the intensity around the landmarks. This is well illustrated in Fig. 2.

3.4.2 Optimization by Gradient Ascent

The optimization is performed by a steepest gradient ascent. We initialize the gradient ascent with the identity deformation, or equivalently $y \leftarrow \bar{y}$:

1. Initialize the gradient ascent with $y \leftarrow \bar{y}$,
2. Iterate until convergence:
   (a) Compute $\nabla_y \ell(x, y; \hat{x}_0, \hat{\tau}_0^2)$,
   (b) Find $a \geq 0$ such that:
   $\ell(x, y + a\nabla_y \ell(x, y; \hat{x}_0, \hat{\tau}_0^2); \hat{x}_0, \hat{\tau}_0^2) \geq \ell(x, y; \hat{x}_0, \hat{\tau}_0^2)$,
   (c) $y \leftarrow y + a\nabla_y \ell(x, y; \hat{x}_0, \hat{\tau}^2, \hat{p}(y))$.

We assume that the algorithm has converged when the likelihood does not increase significantly between two iterations.

3.4.3 Computation of the Likelihood Gradient

The derivative with respect to y of the likelihood function (25) can be written analytically. The inverse transformation though, f_y^{-1}, in the case of spline-based deformation does not have a closed form expression. To overcome this issue we perform the integral change of variable: $s = f_y(t)$. It gives:

$$\ell(x|y;\hat{x}_0,\hat{\tau})$$
$$\propto -\sum_{t\in\Lambda_T}\left[\ln\hat{\tau}_0^2(t)+\frac{|x(f_y(t))-\hat{x}_0(t)|^2}{\hat{\tau}_0^2(t)}\right]|J_{f_y}(t)|. \quad (27)$$

Hence, the intensity $x(f_y(t))$ and the deformation Jacobian $|J_{f_y}(t)|$ depend on the location of the landmarks. Without entering in the details of the computation, it is possible to obtain an analytical expression of the Jacobian gradient with respect to y. As for the intensity, we model the image as a continuous function $x:\mathbb{R}^d\to\mathbb{R}$, such that its derivative can be written as the derivative of the composition $x\circ f_y$ with respect to each landmark coordinate:

$$\frac{\partial x}{\partial y_{kl}}(f_y(t))=\left\langle\frac{\partial x}{\partial c_l}(f_y(t)),\frac{\partial f_y^{(l)}}{\partial y_{kl}}(t)\right\rangle, \quad (28)$$

with $\frac{\partial x}{\partial c_l}(f_y(t))$ the derivative of x with respect to the l-th Cartesian coordinate and $\frac{\partial f_y^{(l)}}{\partial y_{kl}}(t)$ the partial derivative of the l-th coordinate of the deformation with respect to the l-th coordinate of the k-th landmark.

The complete gradient expression is:

$$\frac{\partial\ell(x|y;\hat{x}_0,\hat{\tau}_0)}{\partial y_{kl}}$$
$$=-\frac{1}{2}\sum_{t\in\Lambda_T}\left[\ln\tau_0^2(t)+\frac{(x(f_y(t))-x_0(t))^2}{\tau_0^2(t)}\right]\frac{\partial|J_{f_y}(t)|}{\partial y_{kl}}$$
$$-\sum_{t\in\Lambda_T}\frac{x(f_y(t))-x_0(t)}{\tau_0^2(t)}|J_{f_y}(t)|\frac{\partial x(f_y(t))}{\partial y_{kl}}. \quad (29)$$

When necessary, we use linear interpolation to estimate the image intensity for all values of y and t.

3.5 Detection Results

We use 47 T1-weighted Magnetic Resonance (MR) brain images acquired on a Philips-Intera 3-Tesla scanner, with an isotropic resolution of 1 mm^3. The images were first manually transformed into standardized Talairach space (Talairach and Tournoux 1988) using Analysis of Functional Neuroimages (AFNI) (Cox 1996) to provide a canonical orientation and an approximate alignment.

To manually locate the landmarks in the training set, the images were viewed in continuously synchronized sagittal, axial, and coronal planes. An expert located 2 sets of landmarks in each image. The first set of landmarks is located around the corpus callosum. The posterior extremity, denoted SCC1, is located in the 3D volume as the posterior extremity of the corpus callosum. SCC2 is defined on the same sagittal slice as SCC1, marking the lower extremity of the splenium of the corpus callosum. The second set of landmarks is located around the hippocampus. The expert marks the anterior extremity of the hippocampus, called the head (HoH). The tail of the hippocampus, denoted HT, is defined on the same sagittal slice, marking the posterior extremity of the hippocampus. In the case of the corpus callosum, there is a clear boundary around the structure of interest, but in the case of the hippocampus, it is very difficult even for a specialist to trace the boundary between the hippocampus and the surrounding amygdala, making it challenging to detect the head of the hippocampus. Figure 1 depicts the sagittal slices of an image and the position of the landmarks.

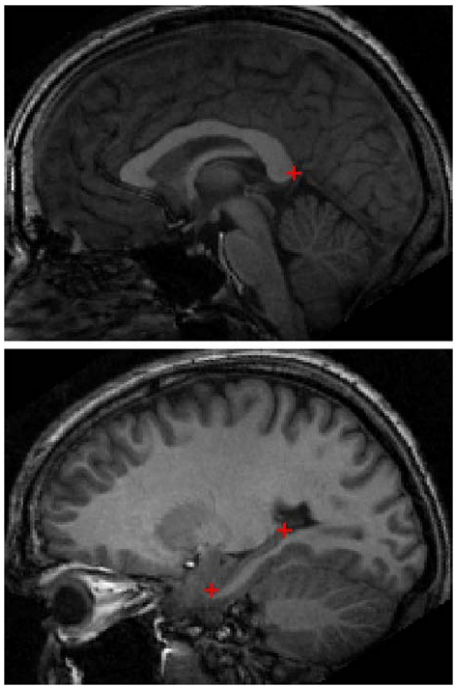
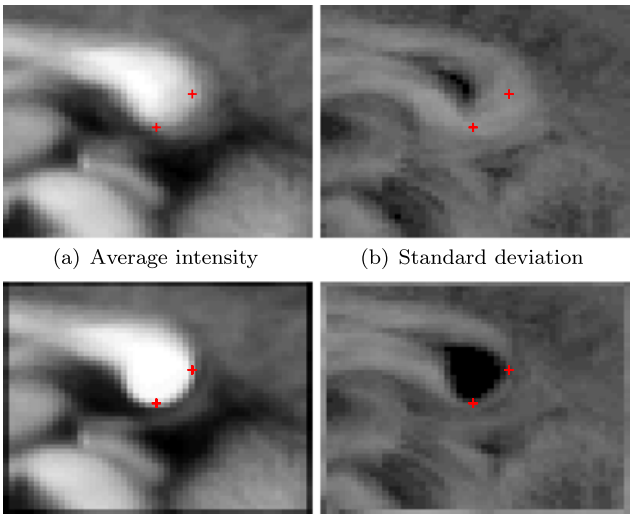


Fig. 1 (Color online) *Top*: Sagittal slice of a brain MR image. The central white structure corresponds to the corpus callosum, the *crosses* represents the position of landmark SCC1. *Bottom*: Sagittal slice at the level of the hippocampus. The *bottom left cross* represents the head of the hippocampus HoH while the *top right cross* marks the location of the tail of the hippocampus HT

The images were acquired with different contrast settings. Since the Deformable Intensity Model does not handle variations of intensity, we first normalize the image intensities. A set of 30 randomly sampled images is used for training, the learnt model is tested on the 17 remaining images.

3.5.1 *Detection in Brain Magnetic Resonance Images*

Estimated Model In the first set of experiments, we choose a Gaussian kernel with $\sigma=7$. We simultaneously detect SCC1 and SCC2 in 2D slices extracted from the 3D volume. Figures 2(a) and (b) depicts the intensity averages and variations across the stack of 30 training images before registration. For comparison, Figs. 2(c) and (d) represents the estimated intensity average and intensity variance of the template. The edges around the landmarks are sharper in the es-



(a) Average intensity (b) Standard deviation

(c) Average intensity: x_0 (d) Standard deviation: τ_0

Fig. 2 (Color online) Estimated Intensity Template ($\sigma = 7$). Intensity distribution in the training image, before (*top*) and after (*bottom*) registration. The *crosses* represent the location of the landmarks: *top-right* SCC1, *bottom-left* SCC2 after registration

timated template than in the intensity average before learning. This is due to the landmark-based registration of the training images. We chose a deformation with a small kernel variance because the point correspondences provide very sparse and local matching information. Therefore the deformation is very local and so does the sharpening of the intensity edges around the landmarks.

Landmark Detection The prediction of the landmark locations is performed on a testing set composed of 17 images. The likelihood is maximized by gradient ascent with respect to the landmark locations according to (29). We define the initial localization error of a landmark by the Euclidean distance between $\bar{y}$, the position of the corresponding landmark in the template and the location marked by the expert. The prediction error of the detection algorithm is defined as the Euclidean distance between the predicted landmark and the ground-truth given by the expert. We compare the performance of the Deformable Intensity Model (DIM) with the detection using SSD. In both cases we use the learnt template and the same deformation model to detect the location of the landmarks.

Table 1 presents the performance of the 2 methods on the detection of SCC1 and SCC2. There exists a clear improvement between the initial error and the detection results obtained by each of the 2 detection methods. The difference of performance between DIM and SSD is significant for SCC1 but not for SCC2. Recall though that SCC1 was located in the 3D volume while SCC2 is identified in the same already selected sagittal slice.

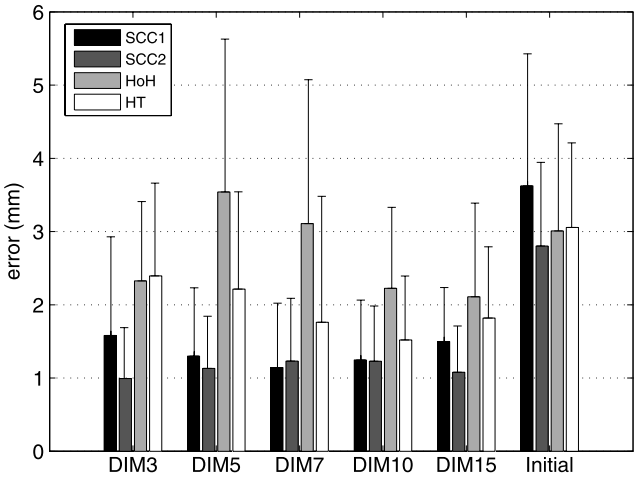


Fig. 3 Performance of the detection algorithm using DIM for different choices of kernel standard deviation: 3, 5, 7, 10 and 15. The landmarks are detected by pair: SCC1 and SCC2, HoH and HT. *Initial* corresponds to the prediction error if one uses the average location of the landmarks in the training set to predict their location in a new image

Table 1 Statistical Comparison of Detection Performance. The left side of the table contains the mean and standard deviation of the prediction error (mm) of SCC1 and SCC2 for each of the methods, on a common testing set composed of 17 images. The righthand side of the table contains the p-value of the Wilcoxon Signed Rank Test for each couple of detection methods. The p-values above the first diagonal of the table represent the test results for SCC1 and below the diagonal the p-value associated to the prediction error of SCC2. The bold figures emphasize the tests validating a difference of performance (with $\alpha = 10\%$)

	Prediction Error (mm)		Wilcoxon Test p-value		
	SCC1	SCC2	DIM	SSD	Initial
DIM	1.14 (0.88)	1.23 (0.86)	N/A	**0.0850**	**0.0002**
SSD	1.61 (0.83)	1.23 (0.74)	1.0000	N/A	**0.0014**
Initial	3.62 (1.80)	2.80 (1.14)	**0.0002**	**0.0002**	N/A

3.6 Choice of the Deformation Model

In this section we investigate how the choice of the kernel influences the performance of the algorithm. We keep a Gaussian kernel but vary its standard deviation: $\sigma = 3$, 5, 7, 10 or 15 pixels. We perform a set of experiments on both the corpus callosum and hippocampus data sets. With a large variance, the number of pixels included in the deformation support increases. Thus more pixels contribute to the likelihood variations. It can be interpreted as increasing the size of the discriminative intensity pattern used for detection.

Figure 3 represents the performance of DIM when the kernel variance varies. For most of the landmarks the best choice is $\sigma = 10$. For some landmarks the detection performance does not depend strongly on the choice of the kernel width. This is the case for SCC2. However for HoH, the width of the kernel modifies the algorithm performance.

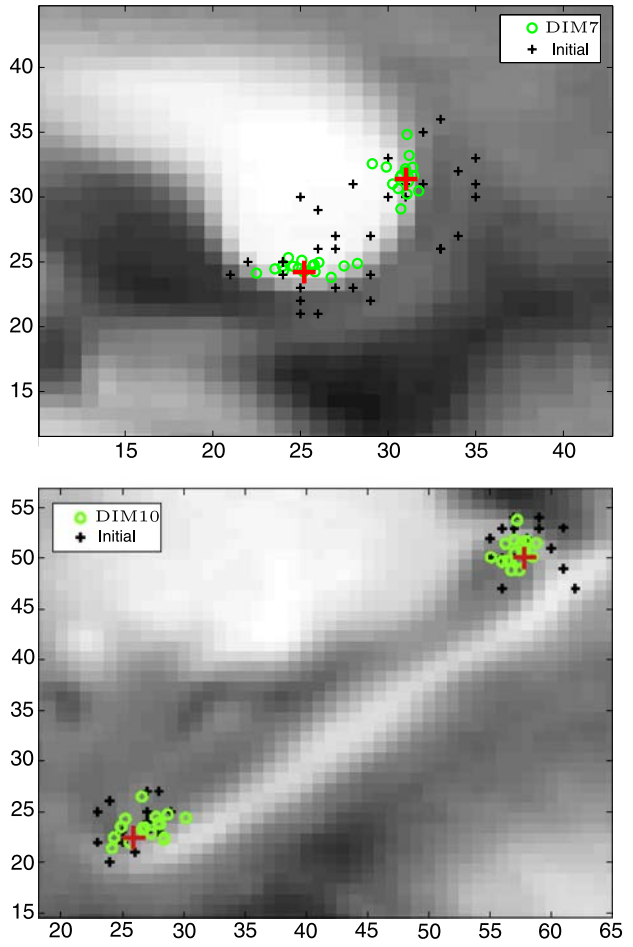


Fig. 4 Distribution of the detection errors around *top*: SCC1 and SCC2 when $\sigma = 7$, *bottom*: HoH and HT when $\sigma = 10$. The *large crosses* represent the location of the landmarks, the *smaller crosses* represent the error before detection and the *circles* represent the error distribution after detection. Notice how they are aligned along the edges of the intensity image

This can be explained by the fact that the intensity pattern around HoH is rather homogeneous, and has a low discriminative power. By increasing the size of the kernel width, we increase the size of the discriminative pattern and with it the specificity of the detection.

Figure 4 represents the spatial distribution of the detection error of DIM around the real location of the landmarks. The error is greatly diminished compared to the localization error before detection. We also notice that the residual error is oriented along the local intensity edge. It is particularly visible in the case of SCC1 and SCC2, but we have observed it in the case of the hippocampus detection as well. This oriented error diminishes when the size of the discriminative pattern increases.

3.7 Discussion

The Deformable Intensity Model is a very simple intensity matching model. Yet, it illustrates well how, by building a statistical generative model of an image, we can derive learning and matching algorithms to estimate the model parameters from training data and detect landmarks by template matching in new images. The derived algorithms are very simple: the learning step consists of a weighted average of the training set after registration, while the testing algorithm is based on gradient ascent. As the proposed models become more complex, both the learning and the testing phases become more challenging, but as the result the matching algorithms inherit of interesting properties.

4 Tissue-Based Deformable Intensity Model

The proposed Deformable Intensity Model (DIM), as any model based on intensity comparison, is not robust to intensity variations. Nevertheless it is often the case that the intensity distribution varies significantly between images, depending on the acquisition protocol. Instead of introducing a normalization step in the preprocessing of the image, we propose to build a statistical model that can deal with the intensity variability and derive the appropriate algorithms.

We propose to build the Tissue-based Deformable Intensity Model (T-DIM), using the same statistical framework and modeling principles. We introduce a non-observed image segmentation in the generative model and derive both the learning algorithm and the template-matching algorithm. The main underlying modeling assumption is that while the intensity distribution of an anatomical tissue varies depending upon the image, the spatial arrangement of the tissues is common to all the images up to some deformation, parametrized by the displacement of the control points or landmarks. Therefore we propose to build a probabilistic deformable model on the tissue-types rather than working directly on the intensity values.

4.1 Description of the Generative Model

We denote by x and y the random real vectors representing respectively the intensity vector of an image and the vector of the K landmark locations. x takes values in $\mathbb{R}^S$ and y takes values in $\mathbb{R}^{dK}$. Let z be a discrete random vector of the same dimension as the image that represents the image segmentation. $z(s)$ is the tissue type at pixel s and takes values in $\{1, \ldots, J\}$, with J the number of tissues. Since the segmentation of the image is unknown, z is a hidden variable. Finally, we introduce u a discrete random variable that characterizes the photometry variations. It allows us to model different acquisition settings, such as high contrast,

low contrast, darker or brighter images, or even an image modality. Since the acquisition parameters are unknown, u is a hidden variable.

The following assumptions are made to simplify the estimation problem. The intensity at a pixel s is assumed to be independent from the intensity at the other pixels, given the corresponding tissue type $z(s)$ and the photometric parameters u. We also assume that the intensity $x(s)$, given the tissue type $z(s)$ and the photometry u is independent from the location of the landmarks. Finally we assume that the tissue type $z(s)$ is independent from the tissue type at the other pixels, given the location of the landmarks y. Figure 5 illustrates with a Bayesian network the complete generative model of an image.

Remark 1 The different random variables of the generative model have different roles. The intensity variables, $x(s)$, i.e. the images, are observed. The landmark locations y are observed in the training set but need to be estimated in the testing set. The segmentation variables $z(s)$ and the photometry variable u are never observed, neither in the training images nor in the testing ones.

The training set x_1^N is composed of N images on which N landmarks y_1^N have been located. Each image of the training set is modeled as a sample of the joint distribution $p(x, y, z, u)$, in which both the segmentation z and the photometry u are missing.

Using the Bayesian network of Fig. 5, the joint distribution can be written as:

$$p(x, y, z, u) = p(u)p(y)\prod_{s\in\Lambda} p(x(s)|z(s), u)p(z(s)|y). \tag{30}$$

Therefore, the joint likelihood of the intensity value and the landmarks is:

$$\ell(x, y) = p(y)\sum_{u} p(u)\prod_{s\in\Lambda}\sum_{j=1}^{J} p(x(s)|z(s) = j, u)p(z(s) = j|y). \tag{31}$$

Hence, to compute the MLE of the landmark locations, $\hat{y} = \arg\max_y \ell(x, y)$, it is necessary to learn the model $\ell(x, y)$ by estimating the probability distributions involved in the likelihood function (31). The four terms to be estimated are:

- the **prior distribution of the landmark locations**, $p(y)$: since y is observed in the training set, it can be estimated from the training data;
- the **prior on the photometry**, $p(u)$: u is unobserved thus it needs to be estimated during training;

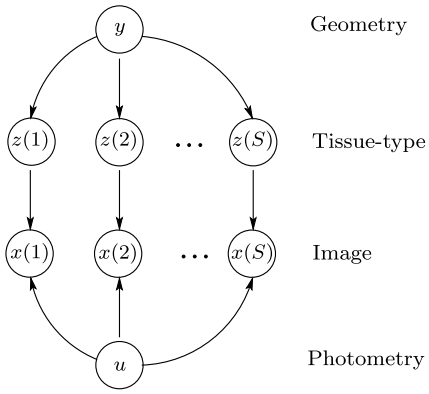


Fig. 5 Bayesian Network representing the Deformable Tissue-Based Intensity Model. y is the location of the landmarks and characterizes the geometry, $z(1), z(2), \ldots, z(S)$ represent the tissue-types at different locations in the image and $x(1), x(2), \ldots, x(S)$ the corresponding intensity values. u characterizes the photometry

- the **photometric model**, $p(x(s)|z(s), u)$: it is modeled as a Gaussian distribution $\mathcal{N}(\mu(j, u), \sigma^2(j, u))$. The parameters of the Gaussian distributions have to be learnt during training;
- the **geometric model**, $p(z(s)|y)$: We assume that the images arise from a common probabilistic deformable tissue model $\pi(j, t), \forall t \in \Lambda_T, \forall j$. At each $t \in \Lambda_T$ the tissue type probability is modeled by a point mass function, $\sum_j \pi(j, t) = 1$. Therefore the conditional distribution $p(z(s) = j|y)$ at s is given by the point mass function: $\pi(j, f_y^{-1}(s))$. The probabilistic template π contains the geometric model of the images

We first detail each of these distributions and then discuss how to estimate them from the training data.

4.1.1 Prior on the Landmark Locations

Since the landmark locations are observed in the training set, the estimation of $p(y)$ is performed independently from the estimation of the rest of the model. The same methods as in Sect. 3.3.2 can be used. Again, we will not use the prior information in the case of our application to landmark detection.

4.1.2 Prior on the Photometry

u is assumed to be a discrete variable, representing different acquisition methods. We model its distribution as a point mass function $p(u)$. Contrarily to the landmark locations, the photometry variable is not observed in the training set. Thus, its marginal distribution needs to be learnt during the training phase, simultaneously with the geometric model and the photometric parameters.

4.1.3 Deformable Tissue Template

The geometry of the image is modeled by a deformable tissue template. It means that the distribution of the tissue types in an image is given by their distribution at the corresponding location in the template, using the image-specific deformation to set the correspondences between the template and the image. The probabilistic template is a function which assigns to each node t of a finite grid $\Lambda_T \subset \mathbb{R}^d$, a point mass function $\pi(j,t), 1 \leq j \leq J$, such that $\sum_{j=1}^{J} \pi(j,t) = 1$. The template definition is extended to a bounded domain of $\mathbb{R}^d$ by linear interpolation.

The location of the landmarks is fixed in the template $\bar{y}$, such that given a family of deformations $\mathcal{F}$, there exists a unique bijective deformation $f_y \in \mathcal{F}$ which maps the template onto the image under the constraint that $f_y(\bar{y}) = y$.

In the deformable model setting, the tissue types are assumed to follow a common distribution across the registered images. Since the registering deformation is characterized by the landmark correspondences, the geometry is in practice encoded by the location of the landmarks. If there are only few landmarks, it is likely that the registration will be precise around the landmarks but potentially inaccurate at further distance. This aspect is taken care of by defining a probabilistic template, able to encode the post-registration geometry variations better than a deterministic template.

Using a deformable model consists in assuming that the spatial distribution of the tissue types given the landmark location follows the distribution given in the template at the corresponding location:

$$\forall s \in \Lambda, \quad p(z(s) = j|y) = \pi(j, f_y^{-1}(s)). \tag{32}$$

4.1.4 Photometric Model

Often in medical imaging, anatomically different tissues appear in different intensity ranges. It is the case in brain images in which 3 anatomically distinct tissues can be easily identified. The 3 tissue type intensity distributions are modeled as a mixture of Gaussian distributions as it is commonly done in brain segmentation methods. We make the same simplifying assumptions as in Wells et al. (1996): the intensity value at a pixel s depends only on the tissue type $z(s)$ and the global photometric variable u. It is assumed that the intensity distribution, given the tissue type, depends neither on the location in the image nor on the landmark location.

Given an image x and the photometric variable u, for all s and for all u:

$$p(x(s)|z(s) = j, u) = g(x(s); \mu(j,u), \sigma^2(j,u)), \tag{33}$$

with $g(x(s); \mu(j,u), \sigma^2(j,u))$ the probability of observing the intensity value $x(s)$ when the tissue model is a Gaussian distribution of parameters $\mu(j,u), \sigma^2(j,u)$. While the model is similar to the mixture model used in image segmentation, the estimation of the Gaussian distribution parameters is coupled with the estimation of the geometry as the proportions of each tissue type comes from the deformable model.

Thus, the likelihood function of an image using the Tissue-based Deformable Intensity Model is:

$$\begin{aligned}\mathcal{L}(x, y; \mu, \sigma^2, \pi) &= p(y) \\ &\quad \times \sum_u p(u) \prod_{s \in \Lambda} \sum_{j=1}^{J} g(x(s); \mu(j,u), \sigma^2(j,u)) \pi(j, f_y^{-1}(s)).\end{aligned} \tag{34}$$

4.2 Model Selection

As usual the purpose of model selection is to estimate the model parameters using the training set of landmarked images. The T-DIM, as described in Sect. 4.1, is a complete generative model of the joint distribution of image intensity x, the landmark location y, the tissue type or image segmentation z and the photometry u.

Both x and y are observed in the training set but z and u are missing. The model parameters are composed of the geometric parameters: $\pi(j,t), \forall j, \forall t$, the photometric parameters $\mu(j,u), \sigma^2(j,u), \forall j, \forall u$ and the marginal distributions of the photometric variable $p(u)$ and of the landmark locations $p(y)$. Since the model parameters, the image segmentation and the photometric parameters are unknown and need to be estimated jointly, we propose to use the Expectation-Maximization (EM) algorithm to perform the model selection. Because y is observed in the training set we work on the conditional model $x|y$.

The EM algorithm is an iterative method to maximize a likelihood function with missing variable. The algorithm iterates between the computation of the expected log-likelihood with the previous estimate of the model parameters, denoted by $\mathcal{Q}$ and maximizing that function with respect to the model parameters. In practice the first step, also called E-step consists in computing the posterior distribution of the hidden variables, in our case the segmentation z and the photometry model u.

4.2.1 Expected Log-Likelihood

The expected log-likelihood is the expectation of the joint log-likelihood with respect to the posterior distribution of the hidden variables:

$$\mathcal{Q}(\theta,\theta') = \mathbb{E}_{z,u}[\ln p_\theta(x_1^N, z_1^N, u_1^N | y_1^N) | x_1^N, y_1^N] = \sum_{i=1}^{N} \sum_s \sum_j \sum_u [A + B + C] p_{\theta'}(z^{(i)}(s) = j, u^{(i)} | x_1^N, y_1^N), \tag{35}$$

with:

$$A = \ln g(x^{(i)}(s); \mu(j, u^{(i)}), \sigma^2(j, u^{(i)})),$$

$$B = \ln \pi(j, f_{y^{(i)}}^{-1}(s)),$$

$$C = \ln p_\theta(u^{(i)}).$$

4.2.2 Details of the E-step

The E-step consists of computing the posterior distribution of the hidden variables given the data x_1^N and the landmarks y_1^N. We firstly simplify the log-expectation with the following proposition derived from the modeling assumptions.

Proposition 1

$$\forall s \in \Lambda, \forall i \in \{1, \ldots, N\},$$
$$p_{\theta'}(z^{(i)}(s) | x_1^N, y_1^N, u^{(i)}) = p_{\theta'}(z^{(i)}(s) | x^{(i)}(s), y^{(i)}, u^{(i)}).$$

Using Proposition 1 and Bayes' formula:

$$p_{\theta'}(z^{(i)}(s), u^{(i)} | x_1^N, y_1^N) = p_{\theta'}(z^{(i)}(s) | x^{(i)}(s), y^{(i)}, u^{(i)}) p_{\theta'}(u^{(i)} | x^{(i)}, y^{(i)}). \tag{36}$$
$$\forall s \in \Lambda,$$

Given the set of model parameters θ' and the distribution $p_{\theta'}(u)$ estimated at the preceding iteration, the posterior distribution is written as the product of two terms:

$$p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)}, u^{(i)}) \propto g(x^{(i)}(s); \mu'(j, u^{(i)}), \sigma'^2(j, u^{(i)})) \pi'(j, f_{y^{(i)}}^{-1}(s)), \tag{37}$$

and,

$$p_{\theta'}(u^{(i)} | x^{(i)}, y^{(i)}) \propto p_{\theta'}(u^{(i)}) \prod_{s \in \Lambda} \times \left[\sum_j g(x^{(i)}(s); \mu'(j, u^{(i)}), \sigma'^2(j, u^{(i)})) \pi'(j, f_{y^{(i)}}^{-1}(s)) \right]. \tag{38}$$

The posterior distribution is computed for each image i, each tissue type j, at each location s and for each photometric model u.

4.2.3 Details of the M-step

The maximization of the $\mathcal{Q}$-function in (35) can be decomposed in three independent maximization problems. Each of them admits a closed form solution. The solution for the photometric parameters, coming from the maximization of the first term of the $\mathcal{Q}$-function (35) are:

$$\hat{\mu}(j, u) = \frac{\sum_i \sum_s x^{(i)}(s) p_{\theta'}(z(s) = j, u | x^{(i)}, y^{(i)})}{\sum_i \sum_s p_{\theta'}(z(s) = j, u | x^{(i)}, y^{(i)})}, \tag{39}$$

$$\hat{\sigma}^2(j, u) = \frac{\sum_i \sum_s (x^{(i)}(s) - \mu'(j, u))^2 p_{\theta'}(z(s) = j, u | x^{(i)}, y^{(i)})}{\sum_i \sum_s p_{\theta'}(z(s) = j, u | x^{(i)}, y^{(i)})}. \tag{40}$$

The number of photometric intensity models U and the number of Gaussian distributions J used to describe the intensity variation is manually chosen before learning the model parameters. If $U < N$, several images may contribute to the estimation of the photometric parameters corresponding to the intensity model u. The contribution of each image to the estimation of the photometric parameters is weighted by the posterior probability of u given the specific image. The images that are unlikely to come from the intensity model u will not contribute to the estimation of its parameters $\mu(j, u), \sigma^2(j, u)$. The solution of the maximization of the third term of the $\mathcal{Q}$-function (35) is:

$$\hat{p}_\theta(u) \propto \sum_i p_{\theta'}(u | x^{(i)}, y^{(i)}). \tag{41}$$

At each iteration, the point mass function of u is updated by computing the proportion of images that are well explained by this model. A normalization term ensures that the result is a probabilistic distribution.

The template update comes from the maximization of the second term of the $\mathcal{Q}$-function (35). Since each image i comes from a specific deformation of the template, the estimation of the template is a complex joint problem. To overcome this difficulty the sum over each image is approximated using for each image the integral change of variable: $s = f_{y^{(i)}}(t)$, as detailed in Sect. 3.3.1. In consequence, the joint maximization with respect to π becomes a set of independent maximizations. The solution can be written in closed form:

$$\hat{\pi}(j, t) \propto \sum_i \sum_u p_{\theta'}(z(f_{y^{(i)}}(t)) = j, u | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) | J_{f_{y^{(i)}}}(t)|, \tag{42}$$

The update is a weighted average of the posterior probabilities of each tissue type at each location t. The contributions of the images are weighted by the local Jacobian value. Images whose grid locally contracts during the registration, i.e. ($|J| < 1$), have a smaller contribution than images whose grid expands locally, i.e. ($|J| > 1$). In regions with no grid deformation ($|J| = 1$), the update consists of computing the average proportions of the different tissue types. Notice that while the change of variable leads to an important simplification of the maximization, it becomes necessary to use some interpolation method on the image support.

4.3 Prediction of the Landmark Location

The prediction problem consists of locating y in a new image x, using the model learnt previously in the training phase. The specificity of the tissue-based model is that the tissue $z(s)$ at each location is unknown. Using the aforementioned model, the log-likelihood of a new image is given by:

$$\begin{aligned}\ell(x, y) &= \ln p(y) \\ &\quad + \sum_{s\in\Lambda} \ln \sum_u p(u) \sum_j g(x(s); \mu(j,u), \sigma^2(j,u))\pi(j, f_y^{-1}(s)).\end{aligned} \tag{43}$$

The maximum likelihood estimator is used to predict the location of the landmarks in the new image. The model parameters $\{\forall j, \forall u, \mu(j,u), \sigma^2(j,u); \forall j, \forall t, \pi(j,t)\}$ and the marginal distributions $p(u)$ and $p(y)$ were learnt during the training phase. Therefore, the likelihood function is optimized with respect to y using a gradient method.

To avoid computing the inverse of the transformation f_y, we perform the approximated integral change of variable $t = f_y^{-1}(s)$, such that the likelihood expression becomes:

$$\ell(x, y) \simeq \ln p(y) + \sum_{t\in\Lambda_T} |J_{f_y}(t)| \left[\ln \sum_u p(u) \sum_j g(x(f_y(t)); \mu(j,u), \sigma^2(j,u))\pi(j,t) \right]. \tag{44}$$

After the change of variable, the intensity values $x(f_y(t))$ and the Jacobian depend on the location of the landmarks. As we did in Sect. 3.4.3, we derive the image and the Jacobian with respect to the landmark locations to obtain the gradient expression (45). We initialize the gradient ascent with $y \leftarrow \bar{y}$.

Algorithm 1 summarizes the learning and landmark detection algorithm associated to the complete generative model.

4.4 Combining Segmentation and Registration

Two main approaches compete in brain MRI segmentation. The first approach assigns to each pixel a label depending on its intensity. This line of work, pioneered by Dempster et al. (1977), Wells et al. (1996), can be used as presented in Leemput (2001) to perform precise segmentation. The competing template-based approach aims at warping a segmented image or an atlas onto the image to be segmented. This approach allows to define regions that span different intensity ranges.

T-DIM belongs to a new set of models combining image segmentation and template-based registration. If the images are pre-registered, T-DIM boils down to a simple mixture of Gaussian distributions with the prior information given by the template. Similarly, if the image segmentation is known, the model boils down to a template-based registration problem using the segmentation as registration cue. The combined model is aimed at performing simultaneous segmentation and registration of images. In the practical example we present, the registration is computed locally only since the purpose is to detect landmarks. Recent efforts have been made to perform the registration of the image onto the atlas and the image segmentation simultaneously, using combined intensity- and template-based models, see e.g., Pohl et al. (2002, 2006) Ashburner and Friston (2005), Fischl et al. (2004), Wang et al. (2006). Notice though that the common objective of these methods is to perform segmentation while in our case, the segmentation is used as a cue for registration. In Wang et al. (2006), the template was independently learnt by averaging manually segmented images. In our work, the template is estimated from the training set which is only composed of images in which few landmarks have been located.

$$\begin{aligned}\frac{\partial \ell(x,y)}{\partial y} &= \frac{\partial p(y)}{\partial y} \cdot \frac{1}{p(y)} + \sum_{t\in\Lambda_T} |J_{f_y}(t)| \frac{\partial x(f_y(t))}{\partial y} \cdot \frac{\sum_u p(u) \sum_j g(x(f_y(t)); \mu(j,u), \sigma^2(j,u))\pi(j,t) \frac{\mu(j,u) - x(f_y(t))}{\sigma^2(j,u)}}{\sum_u p(u) \sum_j g(x(f_y(t)); \mu(j,u), \sigma^2(j,u))\pi(j,t)} \\ &\quad + \sum_{t\in\Lambda_T} \left[\ln \sum_u p(u) \sum_j g(x(f_y(t)); \mu(j,u), \sigma^2(j,u))\pi(j,t) \right] \frac{\partial |J_{f_y}(t)|}{\partial y}.\end{aligned} \tag{45}$$

Algorithm 1 Deformable Tissue-Based Intensity Model

LEARNING

Let (x_1^N, y_1^N) be a training set, $\theta = \{\pi(j,t), \forall j, \forall t;\ \mu(j,u), \sigma^2(j,u), \forall j, \forall u\}$ the set of photometric and geometric parameters, and $p_\theta(u)$ the distribution of the photometric variable.

Initialize $\forall j, \forall u,\ \mu(j,u), \sigma^2(j,u), \pi(j,t), \forall t \in \Lambda_T$, and $p_\theta(u)$.
Iterate until convergence

- **E-step:** $\forall j, \forall u, \forall i, \forall s$, compute the posterior distribution from (37) and (38):

$$p_\theta(z(s) = j, u | x^{(i)}, y^{(i)}) = p_\theta(z(s) = j | x^{(i)}(s), y^{(i)}, u) \cdot p_\theta(u | x^{(i)}, y^{(i)}).$$

- **M-step:**
  - Update the photometric parameters, $\forall j, u$,

$$\mu(j,u) = \frac{\sum_i \sum_s x^{(i)}(s) p_\theta(j,u|x^{(i)}, y^{(i)})}{\sum_i \sum_s p_\theta(j,u|x^{(i)}, y^{(i)})},$$

$$\sigma^2(j,u) = \frac{\sum_i \sum_s (x^{(i)}(s) - \mu(j,u))^2 p_\theta(j,u|x^{(i)}, y^{(i)})}{\sum_i \sum_s p_\theta(j,u|x^{(i)}, y^{(i)})},$$

  - Update the distribution of the photometric model

$$p_\theta(u) \propto \sum_i p_\theta(u|x^{(i)}, y^{(i)}),$$

  - Update the template estimate, $\forall j, t$,

$$\pi(j,t) \propto \sum_i |J_{f_{y^{(i)}}}(t)| \sum_u p_\theta(z(s) = j, u | x^{(i)}, y^{(i)}).$$

TESTING

Let x be a testing image and $\forall t, \forall j, \pi(j,t), \forall j, \forall u, \mu(j,u), \sigma^2(j,u)$, $p(u)$ the parameters and distributions learnt during training,

Initialize $y = \bar{y}$
Iterate until convergence

- **Compute** the gradient direction $\frac{\partial \ell(x,y)}{\partial y}$ using (45),
- **Determine** the step size a such that,

$$\ell\left(x, y + a\frac{\partial \ell(x,y)}{\partial y}\right) \geq \ell(x,y),$$

- **Update** the location of the landmarks,

$$y = y + a \cdot \frac{\partial \ell(x,y)}{\partial y}.$$

5 Tissue-Based Deformable Intensity Model with Image-Specific Photometric Parameters

In the complete generative model presented in what precedes, the images are modeled as samples of the joint distribution $p(x, y, z, u)$. The learning phase allows us to estimate this joint distribution and thus, if desired, to generate random images. The model relies on a fixed and finite[2] number of photometric models U, learnt during training. Because u is modeled as a hidden variable, one needs to integrate with respect to u in order to optimize the log-likelihood. This leads to a computationally involved gradient expression (45). The choice of the number of possible photometric models is balanced between reducing the computational load and capturing the training image intensity variations. Whichever the number of values of u, if the new image intensity distribution does not correspond to the intensity distribution in the training set, the detection of landmarks will be prone to errors.

5.1 Parameter Versus Hidden Variable

One way to address these concerns is to model the photometry as a nuisance parameter rather than as a hidden variable. In our case it makes sense to model it this way, because the intensity parameters may vary tremendously between images. In terms of likelihood, modeling u as a nuisance parameter means that it is enough to work with the conditional distribution:

$$\begin{aligned}\ln p(x, y|u) &= \ln p(y) + \ln \sum_z p(x, z|y, u) \\ &= \ln p(y) + \sum_{s\in\Lambda} \ln \sum_{z(s)} p(x(s)|z(s), u) p(z(s)|y). \end{aligned} \quad (46)$$

During training, the problem is reduced to estimating on the one hand the landmark distribution $p(y)$ and on the other hand the conditional joint probabilities $p(x|z,u)$ and $p(z|y)$. As for the testing algorithm, the predicted landmark location is obtained by optimizing the image and the landmark likelihood $p(x, y|u)$, with respect to y and the nuisance parameters u. We keep modeling the intensity of the image as a mixture of Gaussian distributions, except that in this model the parameters are image specific. We denote the parameters of the j-th Gaussian distribution of the i-th image by $\mu(j,i), \sigma^2(j,i)$. For the sake of simplicity in the notation we sometimes refer to the set of photometric parameters of the i-th image by $u^{(i)}$. Since $u^{(i)}$ is a set of nuisance

[2]Note that if u were a continuous variable, the E-step of the EM algorithm would not be tractable in general. In that case it is necessary to use an approximation of the EM. This problem is studied in Glasbey and Mardia (2001), Allassonniere et al. (2006).

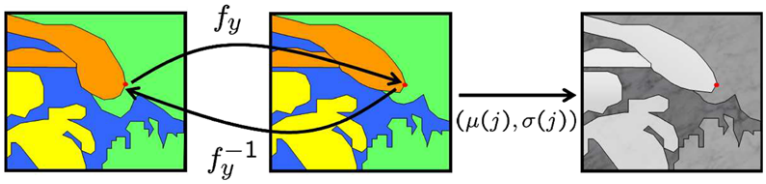


Fig. 6 Probabilistic Tissue-based Deformable Intensity Model. Left to right: a random segmentation sampled from the template distribution, the deformed segmentation, the gray scale image

parameters, it not only needs to be estimated on the training data but also on the testing images. Therefore, the optimization of the likelihood with respect to y cannot be carried out directly and we propose to use the EM algorithm to perform the joint optimization in the learning phase and in the testing algorithm. Figure 6 illustrates the deformable model.

5.2 Model Estimation by the EM Algorithm

5.2.1 Expected Log-Likelihood

Using the same reasoning as in Sect. 4.2.1, we write the expected log-likelihood of a sample of N images in which the location of the landmarks y has been identified. We denote x_1^N the set of N images and use similar notations for the set of landmark locations y_1^N, segmentations z_1^N. We denote by θ the model parameters $\pi(j,t)$ for all j and t and the nuisance parameters $\mu(j,i), \sigma^2(j,i)$ for all i and j. Finally, we denote by θ' their estimates at the preceding iteration,

$$\begin{aligned} &Q(\theta,\theta') \\ &\quad = \mathbb{E}_z[\ln p_\theta(x_1^N, z_1^N | y_1^N) | x_1^N, y_1^N, u_1^N] \\ &\quad = \sum_i \sum_s \sum_j [A+B] p_{\theta'}(z^{(i)}(s) = j | x_1^N, y_1^N, u_1^N), \end{aligned} \quad (47)$$

with:

$$A = \ln g(x^{(i)}(s); \mu(j,u), \sigma^2(j,u)),$$
$$B = \ln \pi(j, f_{y^{(i)}}^{-1}(s)).$$

The $\mathcal{Q}$-function (47) differs from the $\mathcal{Q}$-function of the complete generative model (35) in several aspects. Because the photometry is modeled as a nuisance parameter and not as a hidden variable, we do not need to estimate its distribution, which greatly simplifies the expression of the posterior distribution. On the other hand though, there are as many mixtures of Gaussian distribution to estimate as there are images in the training set.

5.2.2 Details of the E-step

Similarly to Proposition 1, one can prove that

$$\begin{aligned} &\forall s \in \Lambda, \forall i \in \{1,\dots,N\}, \\ &p_{\theta'}(z^{(i)}(s) | x_1^N, y_1^N, u_1^N) = p_{\theta'}(z^{(i)}(s) | x^{(i)}(s), y^{(i)}, u^{(i)}). \end{aligned} \quad (48)$$

The E-step consists of computing the posterior distribution of the tissue type for each image i, each tissue j, and at each location s, using the parameters learnt at the preceding iteration.

$$\begin{aligned} &p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)}, u^{(i)}) \\ &\qquad \propto g(x^{(i)}; \mu'(j,i), \sigma'^2(j,i)) \pi'(j, f_{y^{(i)}}^{-1}(s)). \end{aligned} \quad (49)$$

5.2.3 Details of the M-step

The M-step consists of maximizing each term of $\mathcal{Q}(\theta,\theta')$ with respect to $p(y)$, $\pi(j,t)$, $\mu(j,i)$, $\sigma^2(j,i)$ for all $i \in \{1,\dots,N\}$, $j \in \{1,\dots,J\}$ and for all $t \in \Lambda_T$. The first term (A) of the $\mathcal{Q}$-function (47) is maximized with respect to each image photometric parameters. For each image i and each tissue-type j:

$$\begin{aligned} &\hat{\mu}(j,i) \\ &\quad = \frac{\sum_s x^{(i)}(s) p_{\theta'}(z(s) = j | x^{(i)}(s), y^{(i)}, u^{(i)})}{\sum_s p_{\theta'}(z(s) = j | x^{(i)}(s), y^{(i)}, u^{(i)})}, \end{aligned} \quad (50)$$

$$\begin{aligned} &\hat{\sigma}^2(j,i) \\ &\quad = \frac{\sum_i \sum_s (x^{(i)}(s) - \mu'(j,i))^2 p_{\theta'}(z(s) = j | x^{(i)}(s), y^{(i)}, u^{(i)})}{\sum_s p_{\theta'}(z(s) = j | x^{(i)}(s), y^{(i)}, u^{(i)})}. \end{aligned} \quad (51)$$

Notice that contrarily to the expression in the complete generative model (39), the update is performed independently for each image.

The estimate of the template parameter is unchanged, except that there is no need to sum over all possible values of u. At each pixel t of the template, and for each tissue-type j:

$$\begin{aligned} &\pi(j,t) \\ &\quad \propto \sum_i p_{\theta'}(z(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) |J_{f_{y^{(i)}}}(t)|. \end{aligned} \quad (52)$$

5.3 Landmark Detection

We use the Maximum Likelihood Estimator to predict the location of the landmarks. Denoting $\tilde{\theta}$ the set of nuisance parameters:

$$\{\hat{\tilde{\theta}}, \hat{y}\} = \arg\max_{\tilde{\theta}, y} \ln p_{\tilde{\theta}}(x|y). \quad (53)$$

Contrarily to the MLE with the complete generative model, we need to estimate the nuisance parameters simultaneously with the variable of interest. Therefore it is necessary to employ a joint estimation technique. We propose to do so using the EM algorithm.

5.3.1 Expected Log-Likelihood and E-step

Using the same type of computation as in the training phase, we write the log-expectation to be maximized by the EM algorithm:

$$\begin{aligned} &Q(\tilde{\theta}, y; , \tilde{\theta}', y') \\ &\quad = \mathbb{E}_z[\ln p_{\tilde{\theta}}(x, z|y)|x, y] \\ &\quad = \sum_s \sum_j [A + B] p_{\tilde{\theta}'}(z(s) = j|x(s), y'), \end{aligned} \tag{54}$$

with:

$$A = \ln g(x(s); \mu(j), \sigma^2(j)),$$
$$B = \pi(j, f_y^{-1}(s)).$$

During the E-step the posterior distribution of each tissue type j is computed at each pixel s, using the template $\hat{\pi}(j, t)$ learnt during the training phase.

$$\begin{aligned} &p_{\tilde{\theta}'}(z(s) = j|x(s), y') \\ &\qquad \propto g(x(s); \mu'(j), \sigma'^2(j)) \pi(j, f_{y'}^{-1}(s)). \end{aligned} \tag{55}$$

5.3.2 Modified M-step

The classical M-step would consist of maximizing (54) with respect to y, $\mu(j)$, $\sigma^2(j)$, for all j. While the maximization with respect to the photometric parameters has a closed form solution, the optimization with respect to y is performed by gradient ascent. Unfortunately the expression of the derivative of the $\mathcal{Q}$-function with respect to y is rather complex in that case. Therefore, we propose to modify the M-step, starting with the maximization with respect to the nuisance parameters and then maximizing the log-likelihood with respect to y using the current estimates of the nuisance parameters. Algorithm 2 summarizes the modified EM.

Theorem 1 *$\forall(\tilde{\theta}', y')$, by choosing $\hat{\tilde{\theta}}, \hat{y}$ as described in Algorithm 2,*

$$\ln p_{\hat{\tilde{\theta}}}(x|\hat{y}) \geq \ln p_{\tilde{\theta}'}(x|y').$$

Proof According to the properties of the EM algorithm, choosing $\hat{\theta}$ that maximizes $\mathcal{Q}(\tilde{\theta}, y; \tilde{\theta}', y')$ leads to $\ln p_{\hat{\tilde{\theta}}}(x|y) \geq \ln p_{\tilde{\theta}'}(x|y')$. Since in addition, for all y, $\hat{y}$ is such that: $p_{\hat{\tilde{\theta}}}(x|\hat{y}) \geq p_{\hat{\tilde{\theta}}}(x|y)$, it follows that: $\ln p_{\hat{\tilde{\theta}}}(x|\hat{y}) \geq \ln p_{\tilde{\theta}'}(x|y')$. □

Therefore, the Modified EM algorithm can be used in lieu of the EM algorithm and the likelihood increases at each iteration.

In the case of the photometric parameters, the maximization of the $\mathcal{Q}$-function (54) leads to the same expressions as in the training algorithm: (50) and (51).

The optimization with respect to y is performed on the likelihood function, using the updated values of the nuisance parameters. For simplicity, we use the change of variable $s = f_y(t)$ and maximize the following expression of the likelihood with respect to y:

$$\sum_{t \in \Lambda_T} |J_{f_y}(t)| \ln \sum_{j=1}^{J} \pi(j, t) g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j)). \tag{56}$$

The gradient of the likelihood function can be written analytically (57). The gradient expression is similar to the expression of the gradient of the complete generative model (45), except that there is no need to sum over all possible values of u. In consequence the computation of the gradient expression is less demanding, but the optimization needs to be carried out by an EM algorithm.

$$\begin{aligned} \frac{\partial \ell(x, y; \hat{\tilde{\theta}})}{\partial y} &= \frac{\partial p(y)}{\partial y} \cdot \frac{1}{p(y)} + \sum_{t \in \Lambda_T} |J_{f_y}(t)| \frac{\partial x(f_y(t))}{\partial y} \sum_{j=1}^{N} \frac{\hat{\mu}(j) - x(f_y(t))}{\hat{\sigma}^2(j)} \frac{\pi(j, t) g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j))}{\sum_{j=1}^{N} \pi(j, t) g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j))} \\ &\quad + \sum_{t \in \Lambda_T} \frac{\partial |J_{f_y}(t)|}{\partial y} \ln \sum_{j=1}^{J} \pi(j, t) g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j)). \end{aligned} \tag{57}$$

Algorithm 3 summarizes the training and testing algorithms derived from the Tissue-based Deformable Intensity Model when the photometry is modeled as a nuisance parameter.

Algorithm 2 Modified EM Algorithm

Starting from some initial values of the model parameters: $\theta = \{\tilde{\theta}, y\}$, iterate until convergence:

E-step: Posterior distribution
Given the current estimates of the parameters $\theta' = \{\tilde{\theta}', y'\}$ compute the posterior distribution:

$$p_{\tilde{\theta}'}(z|x, y') \leftarrow \frac{p_{\tilde{\theta}'}(x|z, y')p_{\tilde{\theta}'}(z|y')}{\sum_z p_{\tilde{\theta}'}(x|z, y')p_{\{\tilde{\theta}', y'\}}(z)},$$

M-step: Maximization
Update the model parameters:

$$\hat{\tilde{\theta}} = \arg\max_{\tilde{\theta}} Q(\tilde{\theta}, y; \tilde{\theta}', y'), \quad \hat{y} = \arg\max_{y} \ln p_{\hat{\tilde{\theta}}}(x|y).$$

5.4 Initialization

The algorithm proposed in Algorithm 3 relies on the EM algorithm for learning the model parameter on the one hand, and for estimating the location of the landmarks on the other hand. Since the result of the EM algorithm depends on the initialization, the choice of the initialization is important to achieve stable and reliable results. We detail below the initialization of both the learning and prediction algorithms.

5.4.1 Initialization of the Learning Algorithm

As described in Algorithm 3, the learning phase alternates between estimating the photometric parameters of each image and estimating the proportions of the tissue types at each pixel. One needs to provide to the joint algorithm an initial guess of the intensity parameters as well as of the proportions. We use a Uniform distribution to initialize the tissue proportions at each pixels. As for the photometric parameters, we use a classical EM algorithm as proposed in Wells et al. (1996) to individually estimate for each image a set of photometric parameters. However, because the tissue types are estimated independently on each image, the label of the tissues do not need to match across images. Therefore, in order to recover the correspondence between tissues, we propose to build the following similarity matrix between two images i_1 and i_2, whose elements are:

$$S(j,k) = \sum_s p(z^{i_1}(s) = j|x(s)p(z^{i_2}(s) = k|x(s)). \tag{58}$$

The probability $p(z(s) = j|x(s))$ are computed from the estimated photometric parameters with the individual EM. $S(j,k)$ compares the probability of one pixel to belong to the tissue type j in image i_1 and to belong to the tissue type k in image i_2. If both probabilities are high the similarity

Algorithm 3 Tissue-Based Deformable Intensity Model (Nuisance Parameters)

LEARNING

Let (x_1^N, y_1^N) be a training set, $\theta = \{\forall j, \forall i,\ \mu(j,i),\ \sigma^2(j,i);\ \forall j, \forall t,\ \pi(j,t)\}$ the set of photometric and geometric parameters.

Initialize $\forall j, \forall i,\ \mu(j,i), \sigma^2(j,i)$, and $\forall j, \forall t \in \Lambda_T, \pi(j,t)$
Iterate until convergence

- **E-step:** compute for all j, i, and s,

$$p_\theta(z^{(i)}(s) = j|x^{(i)}(s), y^{(i)}) \propto g(x^{(i)}(s); \mu(j,i), \sigma^2(j,i))\pi(j, f_{y^{(i)}}^{-1}(s))$$

- **M-step:**
  - Update the photometric parameters, for all i and j:

$$\mu(j,i) = \frac{\sum_s x^{(i)}(s)p_\theta(z^{(i)}(s) = j|x^{(i)}(s), y^{(i)})}{\sum_s p_\theta(z^{(i)}(s) = j|x^{(i)}(s), y^{(i)})},$$

$$\sigma^2(j,i) = \frac{\sum_s (x^{(i)}(s) - \mu(j,i))^2 p_\theta(z^{(i)}(s) = j|x^{(i)}(s), y^{(i)})}{\sum_s p_\theta(z^{(i)}(s) = j|x^{(i)}(s), y^{(i)})},$$

  - Update the template estimate, for all j and t,

$$\pi(j,t) \propto \sum_i |J_{f_{y^{(i)}}}(t)| p_\theta(z^{(i)}(s) = j|x^{(i)}(s), y^{(i)}).$$

TESTING

Let x be a testing image of unknown photometric parameters $\tilde{\theta} = (\mu(j), \sigma^2(j), 1 \le j \le J)$ and π the parameters learnt during training,

Initialize $\forall j, \mu(j), \sigma^2(j)$ and $y \leftarrow \bar{y}$
Iterate until convergence

- **E-step:** for all j and s compute,

$$p_{\tilde{\theta}}(z(s) = j|x(s), y) \propto g(x(s); \mu(j), \sigma^2(j))\pi(j, f_y^{-1}(s)).$$

- **M-step:**
  - **Update** the photometric parameters for all j,

$$\mu(j) = \frac{\sum_s x(s)p_{\tilde{\theta}}(z(s) = j|x(s), y)}{\sum_s p_{\tilde{\theta}}(z(s) = j|x(s), y)},$$

$$\sigma^2(j) = \frac{\sum_s (x^{(i)}(s) - \mu(j))^2 p_{\tilde{\theta}}(z(s) = j|x(s), y)}{\sum_s p_{\tilde{\theta}}(z(s) = j|x(s), y)},$$

  - **Compute** the gradient direction $\frac{\partial \ell}{\partial y}(x, y; \tilde{\theta})$ from (57).
  - **Determine** the stepsize a such that,

$$\ell\left(x, y + a\frac{\partial \ell(x, y; \tilde{\theta})}{\partial y}; \tilde{\theta}\right) \ge \ell(x, y; \tilde{\theta}),$$

  - **Update** the location of the landmarks,

$$y = y + a \cdot \frac{\partial \ell(x, y|\tilde{\theta})}{\partial y}.$$

increases. This similarity function relies on the assumption that, in general, the pixels at the same locations belong to the same tissue type. To match corresponding tissues across images, one simply needs to search the label permutation that maximizes the sum of the diagonal term of the similarity matrix.

When all the images come from the same modality one can simply order the tissue types of each images by ranking them based on their respective Gaussian mean.

5.4.2 Initialization of the Landmark Detection Algorithm

The detection algorithm also relies on an EM algorithm, alternating between the estimation of the position of the landmarks and the photometric parameters. We use the position of the landmarks in the template as initial value for the landmark position. Indeed, it corresponds to assuming that the deformation of the template to the image is the identity. As for the photometric parameters, they are estimated by the EM algorithm on the new image, similarly to what is done during training. The labels used in the EM to identify the tissue need to be matched to the tissue type of the estimated template. To do so, we compare the probability of observing a specific tissue type in some parts of the image to the most probable tissue given by the template:

$$S^*(j,k) = \sum_s p(z(s) = k | x(s) \pi(j,s). \tag{59}$$

The best correspondences between tissues are given by the label permutation that maximizes the diagonal terms of the similarity matrix (59).

In simple cases, it is enough to reorder the tissue types based on their estimated Gaussian mean.

6 Experiments

In the following experiments we present some detection results on the database of 2D images containing the corpus callosum that we refer to as 2D-SCC. This data set contains one 2D sagittal slice of 47 3D MR images. The position of SCC1 and SCC2 is given by an expert as described in Sect. 3.5. We use 30 images for training and 17 images for testing. We also present some results on the detection of SCC1 in the whole 3D volume. Since T-DIM models the intensity distribution of each image as a nuisance parameter, there is no need to normalize the image intensities.

Figure 7 pictures few images and the corresponding histograms of 2D-SCC to illustrate the large intensity variations encountered in the database.

We keep working with a Gaussian spline deformation model, and present results for different values of σ ranging between 3 and 15 pixels. The number of tissues used to model the images is fixed before learning the probabilistic deformable tissue template. The brain is usually modeled with 3 major tissues: the Cerebro-Spinal Fluid (CSF), the Gray Matter (GM) and the White Matter (WM). In some cases it is also interesting to consider 2 additional tissue types to model the partial volume effect which generates pixel with mixed intensities. In our experiments the number of tissue types will vary between 2 [3] and 5.

6.1 Template Estimation

We use the estimation and testing algorithm described in Algorithm 3. We compare the performance obtained with this joint algorithm with that obtained with the simplified version introduced in Izard et al. (2006). The simplified model essentially decouples the estimation of the photometry and the geometry in the learning and in the testing algorithms. In terms of algorithms, it means that the intensity distribution of each image is modeled by a specific Gaussian mixture, learned independently in each image using the EM algorithm. This set of parameters is used to learn the tissue template at each pixel independently. We compare the two algorithms in terms of likelihood evolution during learning and in terms of detection performance.

Figure 8 illustrates the evolution of the likelihood of the training set composed of 30 images of 2D-SCC during learning. The template estimation is initialized by a Uniform distribution at each pixel, i.e. $\pi(j,t) = \frac{1}{J}$ for all t and j. The photometric parameters are initialized with the output of a classical EM for Gaussian mixture model estimation performed on each image independently. We compare the likelihood evolution when using the joint optimization as described in Algorithm 3 and the decoupled algorithm. In only few iterations both the joint algorithm and the decoupled optimization converge, except that the decoupled optimization is trapped in a local maximum of the likelihood. We use the parameters estimated at iteration 25 with the decoupled algorithm to initialize the joint algorithm. The likelihood gets out of the local maximum and reaches the same maximum as the joint algorithm. Figure 9 illustrates the template estimated by the decoupled and joint algorithms at iteration 25. The result of the joint optimization is sharper than the one obtained by the decoupled algorithm. For example, in the top right part of the template estimated by the decoupled algorithm, there exists a region with mixed probabilities to observe dark or bright tissue. By coupling the estimation of the template and of the photometric parameters, the latter are more precisely adjusted using the current estimate of the template as prior information. In consequence, the mixed region tends to be assigned to one type of tissue by adjusting the photometric parameters accordingly.

[3] We will use 2 tissue types only in the first experiments to simplify the representation of the learnt template and of the segmentation results.

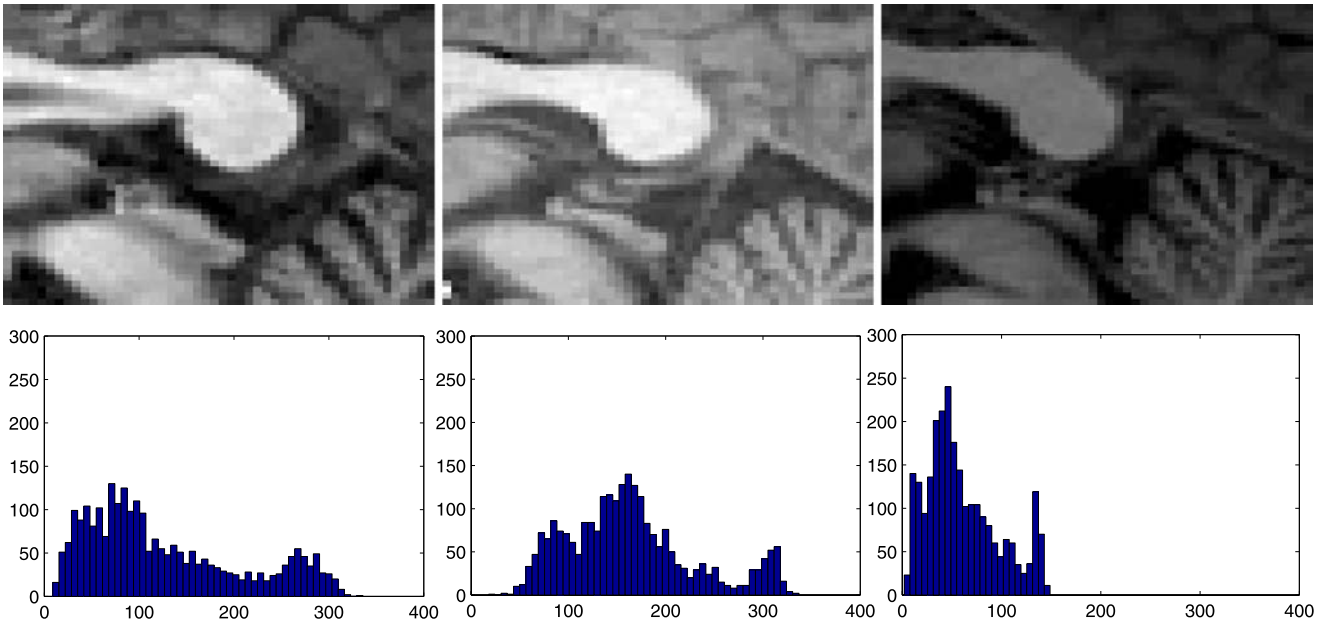


Fig. 7 *Top*: 3 sagittal slices of MR images containing the corpus callosum. *Bottom*: Intensity histograms of the corresponding grayscale images

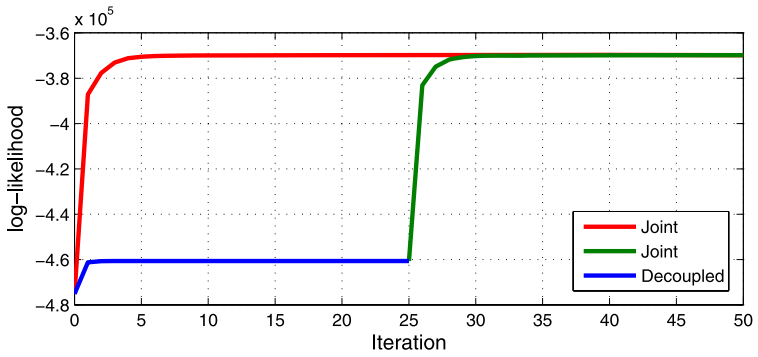


Fig. 8 (Color online) Evolution of the likelihood function during learning. The *red curve* represents the evolution of the likelihood by joint optimization. The *blue curve* represents the likelihood evolution when using the decoupled algorithm and finally the *green curve* represents the evolution of the likelihood when using the joint algorithm, initializing with the template estimate given by the decoupled algorithm presented in Izard et al. (2006). The experiment was performed around SCC1, using 30 images for training, modeling two tissue types. The deformation model is a Gaussian spline with $\sigma = 10$

6.2 Detection Performance

We present the performance of the detection algorithm on SCC1 and SCC2. To assess the advantage of the joint optimization compared to the decoupled optimization in terms of detection, we performed 4 experiments. In the first experiment, denoted by DD, we use the decoupled algorithm detailed in Izard et al. (2006) to perform the detection. In the second experiment, denoted by JD, we use the joint estimation to select the model parameters but perform the landmark detection using the decoupled algorithm. DJ refers to the opposite experiment and finally JJ refers to the complete coupled algorithm. The learning phase is initialized by estimating the intensity parameters on each image using an EM algorithm. Since the EM result depends on its initialization, it is itself initialized by a K-means algorithm and run 3 times. We keep the best set of parameters to initialize the template estimation, i.e. the set of parameters that approximate the best the observed intensity histogram. We repeated the template estimation 5 times and have obtained similar results.

Figure 10 illustrates the cumulative distribution of the prediction error for the experiments JJ, DD, JD and DJ. All 4 algorithms improve significantly the localization of the landmarks, but this is the joint method that achieves the best performance with 50% of the landmarks detected with less than 1 mm of error. Table 2 confirms these observations and shows that there exists a statistically significant differ-

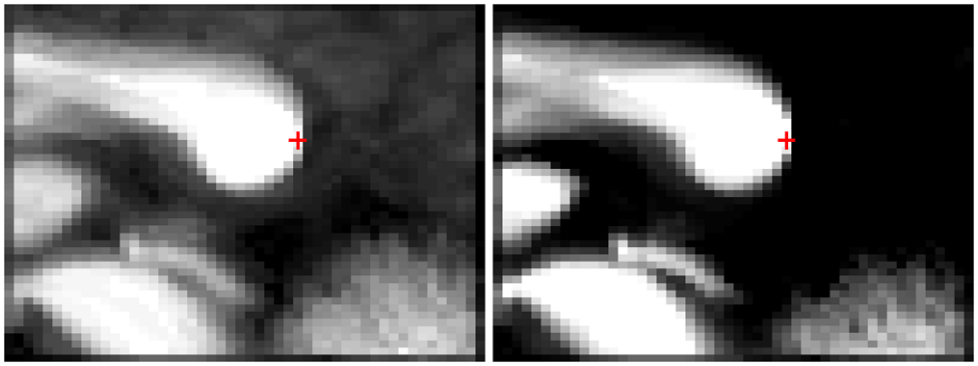


Fig. 9 Estimated Templates in the case of T2-DIM (2 tissue types). We represent the probability at each pixel to observe the brighter tissue. *White* represents a probability close or equal to 1 and *black* represents a probability close or equal to 0. The different shades of *gray* represent intermediate probabilities. The *crosses* shows the location of the landmark SCC1. *Left*: Template estimated by the decoupled algorithm. *Right*: Template estimated by the joint algorithm

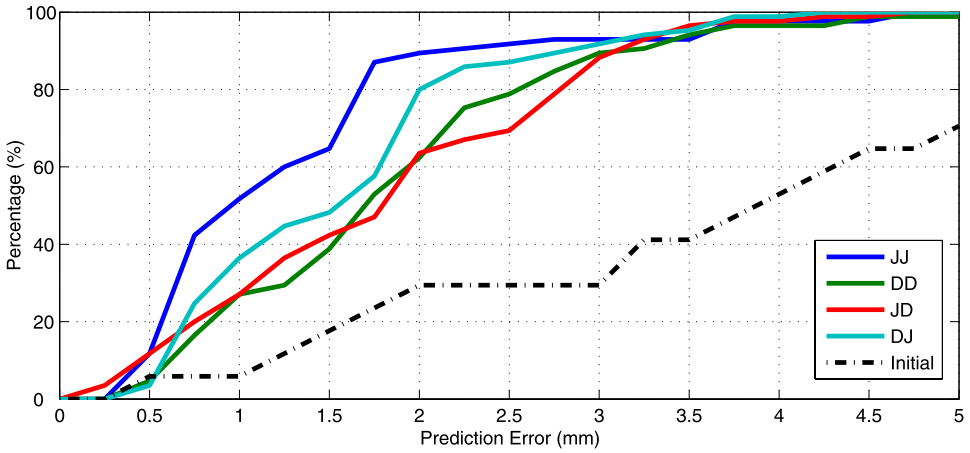


Fig. 10 (Color online) Distribution of the prediction error on the set of 17 testing images (5 estimates per images). We compare 4 algorithms composed of a learning and testing phases, joint J or decoupled D, to the initial distribution of the landmark localization error

Table 2 Prediction performance of each algorithm. p-value associated to the Wilcoxon test comparing the average of the algorithm results

Alg.	Performance (mm)	Statistical Significance			
		JJ	DD	JD	DJ
JJ	1.23 (0.91)	N/A			
DD	1.80 (0.84)	<**0.0001**	N/A		
JD	1.79 (1.06)	**0.0001**	0.9466	N/A	
DJ	1.55 (0.84)	**0.0007**	0.1225	0.1776	N/A
Initial	3.62 (1.80)	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**

ence between JJ and the other algorithms (using a Wilcoxon test). The detection results suggest that there is a significant improvement by working with a unified model rather than proceeding sequentially.

6.3 Combining Registration and Segmentation

Although the main purpose of T-DIM in our application is to locate landmarks by learning and locating characteristic patterns in the image, the algorithm also provides us indirectly with a segmentation of the image. The image segmentation is obtained by assigning each pixel to the tissue with the highest likelihood. The template serves as prior information. Locating the landmarks in a new image is equivalent to finding the best deformation from the template to the image assessing the adequacy of the image segmentation to the deformed tissue template. Figure 11 illustrates on two testing images how the segmentation serves as a cue for the estimation of the landmark location. At first, there is a mismatch between the template and the image segmentation because the template is not well registered with the image. Since the template is used as prior, it produces a poor segmentation of the tip of the corpus callosum. By deforming the template in a way that the segmentation mismatch is minimized, the landmark is brought to the appropriate location in the image.

6.4 Choice of the Parameters

The T-DIM model requires to set by hand two parameters: J the number of tissue types and σ the standard deviation of

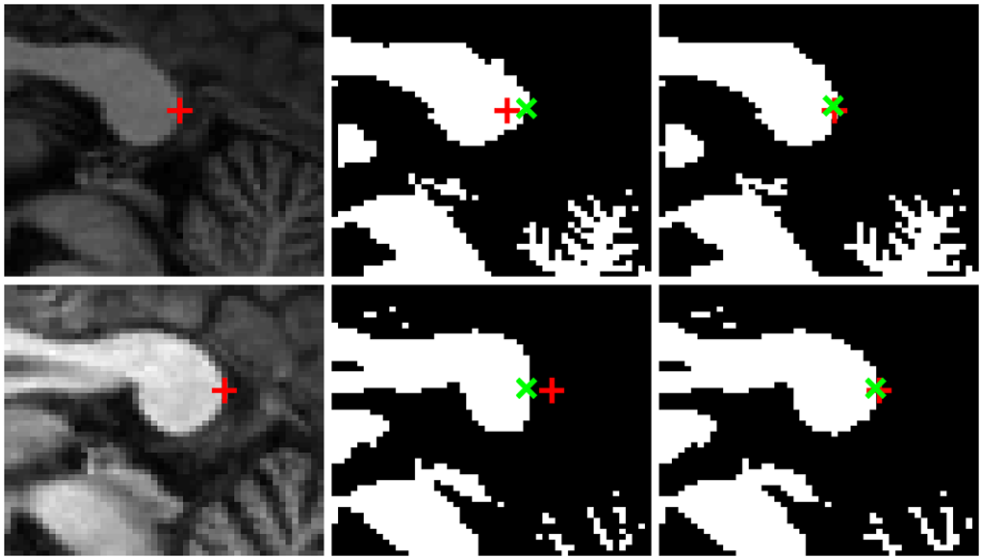


Fig. 11 Combining Registration and Segmentation. Each line represents an image of the training set. The *leftmost image* depicts the original grayscale image and the position of the landmark given by the expert. The *middle column* represents the initialization of the optimization algorithm. Notice how the segmentation does not corresponds well with the *leftmost image*. This mismatch will be corrected by deforming during the template grid during the optimization. The *cross* represents the expert location and the $\times$ the tentative location of the landmarks. In the *rightmost column*, the segmentation is obtained using the estimated deformation to register the template to the image, and using the optimized photometric parameters. The changes are mostly noticeable in the region of the landmark. The $\times$ represents the predicted location of the landmark, the *cross* shows the location marked by the expert

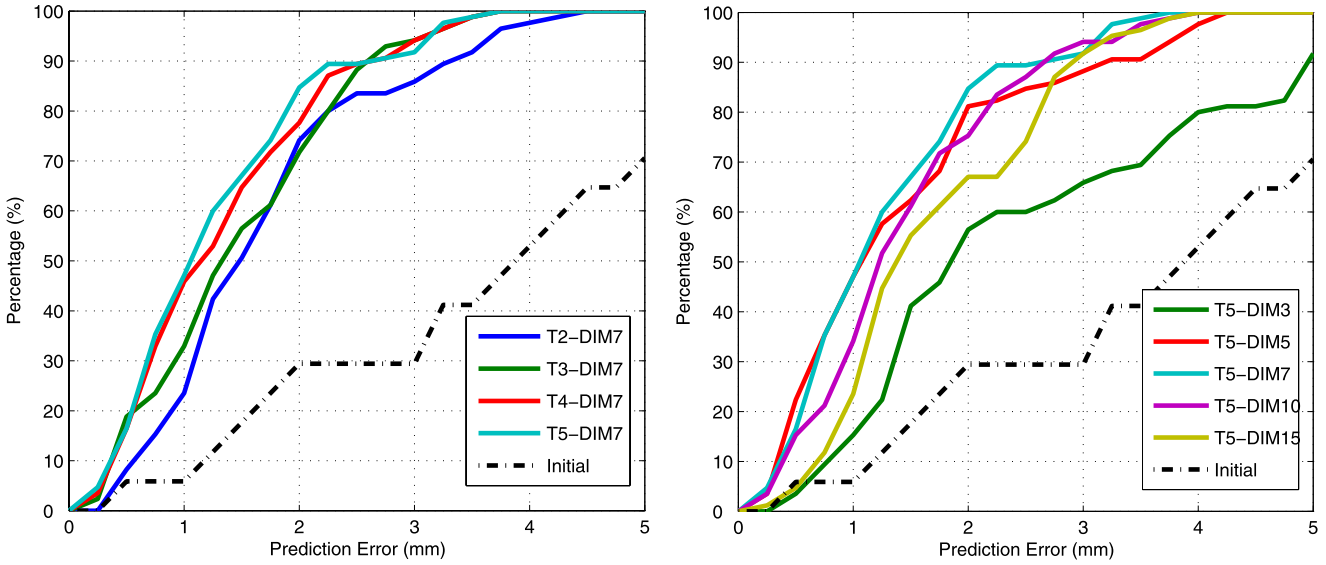
the Gaussian kernel used to model the image deformation. By increasing the number of tissue types, on the one hand it is expected that the precision of the learnt model increases, but on the other hand the number of parameters increases. The size of the Gaussian kernel standard deviation is related to the support of the deformation. If σ is small the tissue pattern used for detection is small too. But if σ increases, so does the size of the tissue pattern. It is expected that the specificity of the detection increases with the kernel width. We already observed this phenomenon in the experiments presented in Sect. 3.5.

We test the algorithm on the detection of SCC1 and SCC2, with J varying between 2 and 5 and with σ varying between 3 and 15 pixels. Similarly to the preceding experiments, the detection is performed 5 times for each image with random initialization. The lowest error for SCC1 is 1.26 mm (0.85 mm) with $J = 5, \sigma = 7$ and for SCC2, 1.04 mm (0.58 mm) with $J = 5, \sigma = 5$. These numerical results are comparable to the performance obtained with DIM, cf. Table 1. Recall that T-DIM contrarily to DIM, does not require any intensity normalization. Figure 12(a) represents the cumulative distribution of the prediction error for different values of the parameters in the case of SCC1. Similar results were obtained for SCC2. We conclude from this experiment that in the case of SCC, the precision increases when the number of tissues in the model increases. The optimal choice of the kernel is related to the amount and the specificity of the information contained around the landmark.

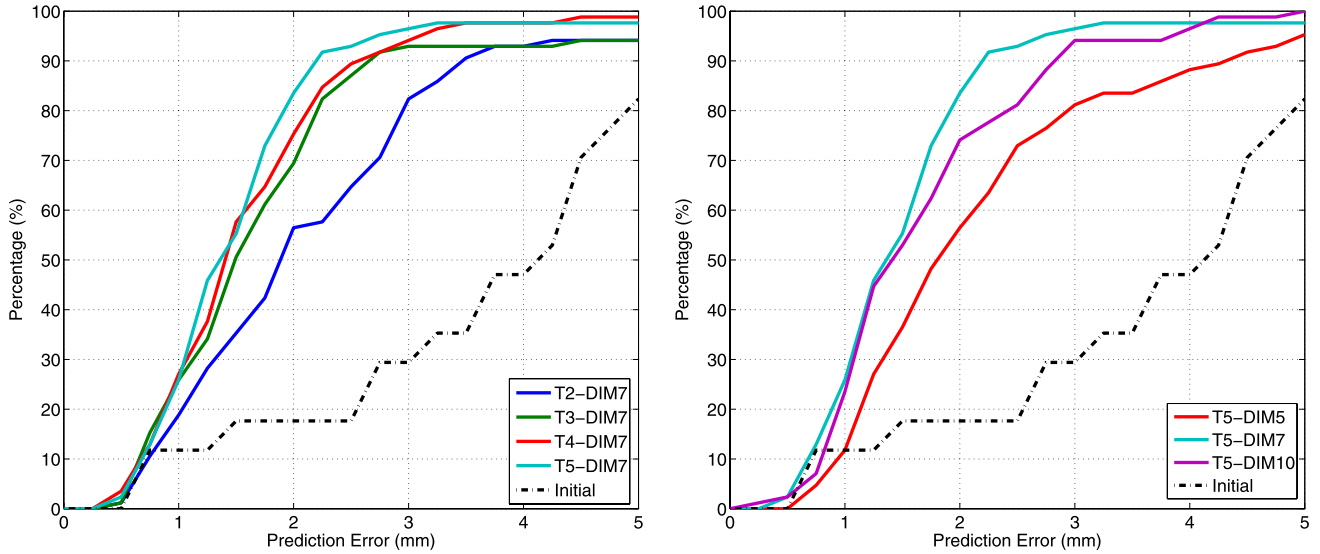
We repeat the experience on 3D-SCC for the detection of SCC1. (Since SCC2 is defined in 2D only, we did not use it in this experiment.) The number of tissues varies from 2 to 5 and the Gaussian kernel parameter from 5 to 10. The experiment is repeated 5 times on each image of the training set. In order to reduce the computational load, in this experiment we compute the likelihood variations using a neighborhood of the landmark of diameter equal to σ. The best performance is achieved for $J = 5$ and $\sigma = 7$. The prediction error is in average 1.48 mm with a standard deviation of 0.82 mm. Before detection the localization error was 3.66 mm (1.69 mm). Figure 12(b) represents the cumulative distribution of the error.

6.5 Performance Evaluation

When assessing the performance of the algorithm in terms of anatomical landmark detection, one needs to keep in mind that the localization of the landmarks, even when located by an expert, is not perfect. To assess the repeatability of the specialist at positioning the landmarks in the anatomy, we asked him several weeks apart to locate again the landmarks in the same images. For SCC1, the average localization error is 0.7 mm with 0.6 mm of standard deviation. Recall that the image resolution is 1 mm^3. The average error for HoH is higher: 1.2 mm with 0.9 mm of standard deviation. Because we use a probabilistic model to represent the geometrical pattern around the landmarks, and learn it from training



(a) Distribution of the prediction error of SCC1 in 2D



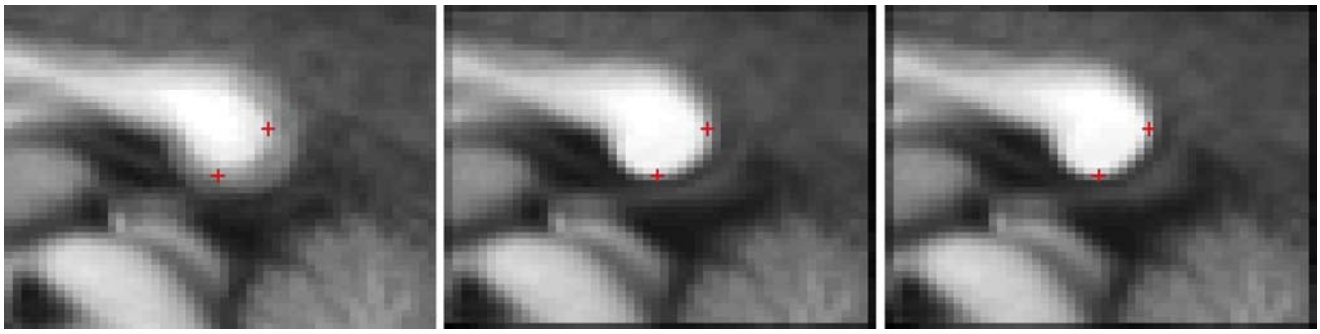
(b) Distribution of the prediction error of SCC1 in 3D

Fig. 12 (Color online) We use the notation T5-DIM7 for example to refer to the T-DIM algorithm with $J = 5$ and $\sigma = 7$. *Initial* in all the graphs represents the distribution of the error before detecting the landmarks. *Left*: Error distribution when the number of tissues varies. *Right*: Error distribution when the standard deviation of the kernel varies

examples, we expect that the initial localization error is averaged out at the time of learning. As for the evaluation of the algorithm performance, we compare the average detection performance of the algorithm with the performance of a trained expert.

6.5.1 Qualitative Assessment

In order to assess the quality of the detection, we present in Fig. 13 the "average" images obtained before registration, when the registration is performed using the automatic landmarks and when the registration is based on the landmarks located manually. We use the same model for registration and for prediction, i.e. the Gaussian spline deformation with $\sigma = 7$. If the images are well registered the corresponding structures should coincide and therefore the average image should be sharp. We observe that the average images obtained using the automatic landmarks and the manual landmarks are similar. This shows that the precision of the detection around the corpus callosum is adequate for registering images based on the automatically detected landmarks.



(a) Before Registration (b) Automatic Registration (c) Expert Registration

Fig. 13 Testing Image Registration. Each subfigure represents the pixel-by-pixel intensity average of the 17 testing images. The *crosses* represent the landmark locations $\bar{y}$. Subfigure (**a**) is computed before detecting the landmarks, i.e. the images have only been globally aligned to Talairach's atlas. Before computing the average image depicted in Subfigures (**b**) and (**c**), the images were registered to the template based on the landmark correspondences, using a Gaussian spline deformation ($\sigma = 7$). In (**b**) the correspondences are set using the automatic landmarks while in (**c**) we use the manual landmarks

Table 3 Prediction performance for each algorithm. "+ Norm" means that the image intensities were normalized before running the algorithm, "+ Flip" means that the intensity of the testing images have been modified as described in Sect. 6.5.2

	Performance (mm)	
	SCC1	SCC2
DIM + Norm.	1.14 (0.88)	1.23 (0.86)
SSD + Norm.	1.61 (0.83)	1.23 (0.71)
DIM	1.95 (1.74)	1.77 (1.12)
SSD	1.88 (1.64)	1.76 (1.25)
T-DIM	**1.31 (0.85)**	**1.26 (0.72)**
T-DIM + Flip	**1.23 (0.86)**	**1.33 (0.85)**
Initial	3.62 (1.80)	2.80 (1.14)

6.5.2 *Robustness to Intensity Variations*

Table 3 summarizes the performance of SSD, DIM and T-DIM for the detection SCC1 and SCC2. Both DIM and SSD lack robustness to intensity variation. In contrast, T-DIM achieves the same performance as DIM + Norm., but without normalizing the image intensities. Therefore T-DIM has the potential to be applicable to images from different modalities. To further evaluate the robustness of T-DIM to the change of intensity range, we create a synthetic data set from the testing images. We modify the image intensity such that the pixels belonging to the white matter appear at low intensity and the pixels belonging to the CSF appear with high intensity. Using the same training set, we learn the model parameters and use the learnt model to predict the location of the landmarks in the synthetic testing set. The results for T5-DIM7 are given in Table 3. Using a paired test, we found no significant differences between the performance on the original testing set and the synthetic testing set, when $\sigma = 5, 7$ or 10 and $J = 2, 3, 4$ or 5.

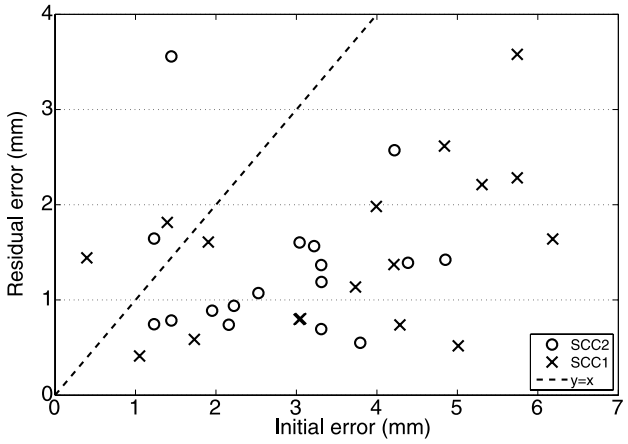


Fig. 14 Prediction performance. x-axis: error before landmark detection, y-axis: residual error after landmark detection. The *dashed line* represents $y = x$. Each symbol corresponds to the detection of SCC1 or SCC2 in one of the 17 testing images

6.5.3 *Robustness to Deformations*

Different measures, e.g., in Schmid et al. (2000), Hartkens et al. (1999), have been proposed to assess the quality of matching algorithms. For example one measures the repeatability of the detection when the image undergoes different types of transformations and/or deformations. Because we used a simple deformation model, we do not expect the resulting algorithm to be robust to large rotations, or changes of scale. Nevertheless, it is possible to look at the prediction performance as a function of the distance between $\bar{y}$, the origin of the gradient descent, and $y^\star$ the actual location of the landmarks. Figure 14 is a scatter plot with the prediction error on the y-axis and the initialization error on the x-axis. Each cross or circle represents the detection of SCC1 or SCC2 in one of the testing images. A vast majority of the detection results are below $y = x$ illustrating the

reduction of the localization error. This plot allows us to determine that, as expected, the prediction of the location of landmarks is more accurate if the initialization is close from the actual landmark location.

It would be possible to improve the robustness of the detection algorithm to affine registration by changing the spline model to the Thin Plate Spline or any other kernels containing an affine component. In both cases though, one needs to reduce the domain of computation as the support of the deformation is infinite.

7 Conclusion

We have illustrated how by building generative models and applying classical statistical learning techniques, it is possible to learn a model from training data and derive an optimal matching algorithm from the learnt model. In the particular case of landmark detection, the method allows us to learn the distinctive intensity pattern automatically by training the model using annotated images, without any prior information on the type of landmarks. It easily adapts to the simultaneous detection of one or more landmarks.

Although the method has been illustrated on MR images, it can be extended to other image modalities and more interestingly to non-scalar image modalities. In the latter case, one may need to build statistical models on non-Euclidean spaces in order to model the likelihood of an image. It is also necessary to understand how deformations act on this type of images.

Finally, in this paper we focus on the problem of landmark detection, which is equivalent to a registration problem with a small number of control points. If the number of control points increases so that the whole image support can be deformed, the proposed methods can be used to derive registration, segmentation or even joint segmentation-registration algorithms.

Acknowledgements This work has been founded by the Graduate Fellowship of the Université des Sciences et Technologies de Lille (Lille, France), as well as general funds of the Center for Imaging Science and the Department of Biomedical Engineering of the Johns Hopkins University (Baltimore MD, USA). The authors are particularly grateful to Dr. Craig Stark for providing the annotated images on which the proposed method has been demonstrated, to Profs. Michael Miller and Elliot McVeigh for supporting this work, as well as to Profs. René Vidal and Jean-Louis Bon for many fruitful conversations.

References

Allassonniere, S., Kuhn, E., Trouvé, A., & Amit, Y. (2006). Generative model and consistent estimation algorithms for non-rigid deformable models. In *Acoustics, speech and signal processing, 2006. ICASSP 2006 proceedings. 2006 IEEE international conference on 5*, V–V.

Allassonnière, S., Amit, Y., & Trouvé, A. (2007). Toward a coherent statistical framework for dense deformable template estimation. *Journal of the Royal Statistical Society B*, *69*, 3–29.

Arad, N., Dyn, N., Reispeld, D., & Yeshurun, Y. (1994). Image warping by radial basis functions: application to facial expressions. *CVGIP: Graphical Models and Image Processing*, *56*, 161–172.

Ashburner, J., & Friston, K. J. (1999). Nonlinear spatial normalization using basis functions. *Human Brain Mapping*, *7*, 254–266.

Ashburner, J., & Friston, K. J. (2005). Unified segmentation. *NeuroImage*, *26*, 839–851.

Bajcsy, R., Kovačič, S. (1989). Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing*, *46*, 1–21.

Barnea, D. I., & Silverman, H. F. (1972). A class of algorithms for fast digital image registration. *IEEE Transactions on Computers*, *21*(2), 179–186.

Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(6), 567–585.

Bookstein, F. L. (1992). *Morphometric tools for landmark data: geometry and biology*. Cambridge: Cambridge University Press.

Bro-Nielsen, M., & Gramkow, C. (1996). Fast fluid registration of medical images. In *Lecture notes in computer science: Vol. 1131. Proceeding of 4th international conference on visualization in biomedical computing (VBC'96)* (pp. 267–276). Berlin: Springer.

Collignon, A., Maes, F., Delaere, D., Vandermeulen, D., Suetens, P., & Marshal, G. (1995). Automated multi-modality image registration based on information theory. In C. B. Y. Bizais & R. D. Paola (Eds.), *Information processing in medical imaging* (pp. 263–274). Dordrecht: Kluwer Academic.

Cox, R. (1996). Afni: Software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research*, *29*, 162–173.

Dalal, N., & Triggs, B. (2005). *Histograms of oriented gradients for human detection* (pp. 886–893).

Davatzikos, C. (1997). Spatial transformation and registration of brain imaging using elastically deformable models. *Computer Vision and Image Understanding*, *2*(66), 207–222.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, *39*, 1–38.

Fischl, B., Salat, D. H., van der Kouwe, A. J., Makris, N., Ségonne, F., Quinn, B. T., & Dale, A. M. (2004). Sequence-independent segmentation of magnetic resonance images. *NeuroImage*, *23*, S69–S84.

Frantz, S., Rohr, K., & Stiehl, H. (2000). Localization of 3D anatomical point landmarks in 3D tomographic images using deformable models. In *Lecture notes in computer science: Vol. 1935. Proc. MICCAI* (pp. 492–501). Berlin: Springer.

Friston, K. J., Ashburner, J., Poline, J. B., Frith, C. D., Heather, J. D., & Frackowiak, R. (1995). Spatial registration and normalisation of images. *Human Brain Mapping*, *2*, 165–189.

Glasbey, C., & Mardia, K. (2001). A penalized likelihood approach to image warping (with discussion). *Journal of the Royal Statistical Society B*, *63*, 465–514.

Goshtasby, A., Staib, L., Studholme, C., & Terzopoulos, D. (2003). Non-rigid image registration: Guest editors' introduction. *Computer Vision and Image Understanding*, *89*(2/3), 109–113.

Grenander, U., & Miller, M. (1998). Computational anatomy: An emerging discipline. *Quarterly of Applied Mathematics*, *4*, 617–694. LVI.

Hartkens, T., Rohr, K., & Stiehl, H. (1999). Performance of 3D differential operators for the detection of anatomical landmarks in MR and CT images. In *Medical imaging 1999: image processing. Proceedings of the SPIE international symposium* (Vol. 5032, pp. 32–43).

Izard, C., Jedynak, B., & Stark, C. (2006). Spline-based probabilistic model for anatomical landmark detection. In R. Larsen, M. Nielsen, & J. Sporring (Eds.), *Lecture notes in computer science: Vol. 4190. Medical imaging computing and computer assisted intervention (MICCAI)* (pp. 849–856). Berlin: Springer.

Joshi, S., & Miller, M. (2000). Landmark matching via large deformation diffeomorphisms. *IEEE Transactions on Image Processing*, *9*, 1357–1370.

Leemput, K. V. (2001). A statistical framework for partial volume segmentation. In W. Niessen & M. Viergever (Eds.), *Lecture notes in computer science: Vol. 2208. MICCAI* (pp. 204–212). Berlin: Springer.

Lester, H., Arridge, S., Jansons, K., Lemieux, L., Hajnal, J., & Oatridge, A. (1999). Non-linear registration with the variable viscosity fluid algorithm. In *Information processing in medical imaging (IPMI'99)* (pp. 238–251).

Levin, A., & Weiss, Y. (2006). Learning to combine bottom-up and top-down segmentation. In *Lecture notes in computer science: Vol. 3954. ECCV* (pp. 581–594). Berlin: Springer.

Li, H., Manjunath, B. S., & Mitra, S. K. (1995). A contour–based approach to multisensor image registration. *IEEE Transactions on Image Processing*, *4*(3), 320–334.

Lowe, D. (2003). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *20*, 91–110.

Maes, F., Collignon, A., Vandermeulen, D., Marshal, G., & Suetens, P. (1997). Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, *16*, 187–198.

Pohl, K. M., Wells, W. M., Guimond, A., Kasai, K., Shenton, M. E., Kikinis, R., Grimson, W. E. L., & Warfield, S. K. (2002). Incorporating non-rigid registration into expectation-maximization algorithm to segment mr images. In T. Dohi & R. Kikinis (Eds.), *Lecture notes in computer science: Vol. 2488. MICCAI* (pp. 564–571). Berlin: Springer.

Pohl, K. M., Fisher, J., Grimson, W. E. L., Kikinis, R., & Wells, W. M. (2006). A Bayesian model for joint segmentation and registration. *NeuroImage*, *31*(1), 228–239.

Pratt, W. K. (1974). Correlation techniques for image registration. *IEEE Transactions on Aerospace and Electronic Systems*, *10*(3), 353–358.

Qiu, A., Younes, L., Wang, L., Ratnanather, J. T., Gillepsie, S. K., Kaplan, G., Csernansky, J., & Miller, M. I. (2007). Combining anatomical manifold information via diffeomorphic metric mappings for studying cortical thinning of the cingulate gyrus in schizophrenia. *NeuroImage*, *37*(3), 821–833.

Roche, A., Malandain, G., & Ayache, N. (2000). Unifying maximum likelihood approaches in medical image registration. *International Journal of Imaging Systems and Technology*, *11*(1), 71–80.

Rohr, K. (2001). *Landmark-based image analysis using geometric and intensity models*. Dordrecht: Kluwer Academic.

Rohr, K., Stiehl, H., Sprengel, R., Buzug, T., Weese, J., & Kuhn, M. (2001). Landmark-based elastic registration using approximating thin-plate splines. *IEEE Transactions on Medical Imaging*, *20*(6), 526–534.

Schmid, C., Mohr, R., & Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, *37*(2), 151–172.

Studholme, C., Hill, D. L. G., & Hawkes, D. J. (1995). Multiresolution voxel similarity measures for MR–PET registration. In C. B. Y. Bizais & R. D. Paola (Eds.), *Information processing in medical imaging* (pp. 287–298). Dordrecht: Kluwer Academic.

Szeliski, R. (2006). Image alignment and stitching: A tutorial. *Fundamental Trends in Computer Graphics and Vision*, *2*(1), 1–104.

Talairach, J., Tournoux, P. (1988) *Co-planar stereotaxic atlas of the human brain*. Stuttgart: Thieme Medical.

Thirion, J. P. (1996). New feature points based on geometric invariants for 3D image registration. *International Journal of Computer Vision*, *18:2*, 121–137.

Twining, C., Marsland, S., & Taylor, C. (2002). *Measuring geodesic distances on the space of bounded diffeomorphisms*.

Viola, P. (1995). *Alignment by maximization of mutual information*. Ph.D. thesis, Massachusetts Institute of Technology.

Wahba, G. (1990). *Spline models for observational data*. Philadelphia: Society for Industrial and Applied Mathematics.

Wang, F., Vemuri, B. C., & Eisenschenk, S. J. (2006). Joint registration and segmentation of neuroanatomic structures from brain mri. *Academic Radiology*, *13*(9), 1104–1111.

Wells, W., Kikinis, R., Grimson, W., & Jolesz, F. (1996). Adaptive segmentation of MRI data. *IEEE Transactions on Medical Imaging*, *15*, 429–442.

Wörz, S., & Rohr, K. (2006). Localization of anatomical point landmarks in 3D medical images by fitting 3d parametric intensity models. *Medical Image Analysis*, *10*(1), 41–58.

Zitová, B., & Flusser, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, *21*, 977–1000.

Chapter 11

Skin detection using pairwise models

Skin Detection Using Pairwise Models ⋆

Bruno Jedynak, Huicheng Zheng and Mohamed Daoudi [1,2]

Abstract

We consider a sequence of three models for skin detection built from a large collection of labeled images. Each model is a maximum entropy model with respect to constraints concerning marginal distributions. Our models are nested. The first model, called the baseline model is well known from practitioners. Pixels are considered independent. Performance, measured by the ROC curve on the Compaq Database is impressive for such a simple model. However, single image examination reveals very irregular results. The second model is a Hidden Markov Model which includes constraints that force smoothness of the solution. The ROC curve obtained shows better performance than the baseline model. Finally, color gradient is included. Thanks to Bethe tree approximation, we obtain a simple analytical expression for the coefficients of the associated maximum entropy model. Performance, compared with previous model is once more improved.

Key words: maximum entropy models, skin detection, Markov random field.

1 Introduction

Skin detection consists in detecting human skin pixels from an image. The system output is a binary image defined on the same pixel grid as the input image.

⋆ This work was partially supported by European Community IAP 2117/27572-POESIA www.poesia-filter.org

[1] Bruno Jedynak is within the Laboratoire de Mathématiques Paul Painlevé, USTL, Bât M2, Cité scientifique, 59655 Villeneuve d'Ascq, France. He is currently Visiting Associate Professor at the Center for Imaging Science, The Johns Hopkins Univesity.
Email: bruno.jedynak@jhu.edu

[2] Huicheng Zheng and Mohamed Daoudi are within MIIRE Group, INT/LIFL (CNRS UMR 8022), Rue G. Marconi, Cité scientifique, 59655 Villeneuve d'Ascq, France.
Email: (Zheng,Daoudi)@enic.fr

Skin detection plays an important role in various applications such as face detection [1], searching and filtering image content on the web [2][3]. Research has been performed on the detection of human skin pixels in color images and on the discrimination between skin pixels and "non-skin" pixels by use of various statistical color models. Some researchers have used skin color models such as Gaussian, Gaussian mixture or histograms [4] [5]. In most experiments, skin pixels are acquired from a limited number of people under a limited range of lighting conditions.

Unfortunately, the illumination conditions are often unknown in an arbitrary image, so the variation in skin colors is much less constrained in practice. This is particularly true for web images captured under a wide variety of conditions. However, given a large collection of labeled training pixels including all human skin (Caucasians, Africans, Asians) we can still model the distribution of skin and non-skin colors in the color space. Recently, in [6], the authors proposed to estimate the distribution of skin and non-skin color using labeled training data. The comparison of histogram models and Gaussian mixture density models estimated with EM algorithm was analyzed for the standard 24-bit RGB color space. The histogram models were found to be slightly superior to Gaussian mixture models in terms of skin pixel classification performance.

A skin detection system is never perfect and different users use different criteria for evaluation. General appearance of the skin-zones detected, or other global criteria might be important for further processing. For quantitative evaluation, we will use false positive rate and detection rate. False positive rate is the proportion of non-skin pixels classified as skin and detection rate is the proportion of skin pixels classified as skin. The user might wish to combine these two indicators his own way depending on the kind of error he is more willing to afford. Hence we propose a system where the output is not binary but a floating number between zero and one, the larger the value, the larger the belief for a skin pixel. The user can then apply a threshold to obtain a binary image. Error rates for all possible thresholding are summarized in the Receiver Operating Characteristic (ROC) curve.

We have in our hands the Compaq Database [6]. It is a catalog of almost twenty thousand images. Each of them is manually segmented such that the skin pixels are labeled. Our goal in this paper is to explore different ways in which this set of data can be used to perform skin detection on new images. We will use Markov random field approach [7] [8] combined with Maximum Entropy Modeling [9] [10], referred to as MaxEnt.

Maximum Entropy Modeling (MaxEnt) is a method for inferring models from a data set. See [9] for the underlying philosophy. It works as follows: 1) choose relevant features 2) compute their histograms on the training set 3) write down the maximum entropy model within the ones that have the feature histograms

as observed on the training set 4) estimate the parameters of the model 5) use the model for classification. This plan has been successfully completed for several tasks related to speech recognition and language processing. When working with images, the graph underlying the model is the pixel lattice. It has many nodes and many loops. Task 4) is much more difficult. A break through appeared with the work in [11] on texture simulation where 1) 2) 3) 4) was performed for images and 5) replaced by simulation.

We adapt this methodology to skin detection as follows: in 1) we specialize in colors and skinness for one pixel and two adjacent pixels. In 2) we compute the histogram of these features in the Compaq manually segmented database. Models for 3) are then easily obtained. In 4) we use the Beth tree approximation, see [12]. It consists in approximating locally the pixel lattice by a tree. The parameters of the MaxEnt models are then expressed analytically as functions of the histograms of the features. This is a particularity of our features. In 5) we use the Gibbs sampler algorithm for inferring the probability for skin at each pixel location.

We consider a sequence of three maximum entropy models with respect to various constraints concerning marginal distributions. The first model imposes constraints on one-pixel marginals. The solution is a baseline model in which pixels are considered independent. This model is well known from practitioners[6]. The baseline model is certainly too loose and does not take into account the fact that skin zones are not purely random but are made of large regions with regular shapes. Hence, in the second model, we add constraints on the distribution of neighboring labels in order to smooth the solution. Finally, color gradient is included in building the third model. We hope that the changes in neighboring colors will help discriminate skin pixels from non-skin ones.

The rest of this paper is organized as follows: After setting up the notations in section 2, we present in section 3 the baseline model. In section 4, we present the second model, which is a hidden Markov Random Field model. A novel method for parameter estimation is explored. In section 5, we examine the third model which takes into account the color gradient. Finally, in Section 7 we present concluding remarks.

2 Notations

Let us fix the notations. The set of pixels of an image is S. The color of a pixel $s \in S$ is x_s. It is a 3 dimensional vector, each component being usually coded on one octet. We notate $C = \{0, \ldots, 255\}^3$. The "skinness" of a pixel s, is y_s with $y_s = 1$ if s is a skin pixel and $y_s = 0$ if not. The color image, which is the vector of color pixels, is x and the binary image made up of the y_s's is

notated y. The letter "p" will denote "probability of". The actual probability measure will depend upon context.

Let us assume for a moment that we knew the joint probability distribution $p(x, y)$ of the vector (x, y), then Bayesian analysis tells us that, whatever cost function the user might think of, all that is needed is the posterior distribution $p(y|x)$. From the user's point of view, the useful information is contained in the one pixel marginal of the posterior, that is, for each pixel, the quantity $p(y_s = 1|x)$, quantifying the belief for skinness at pixel s given the full color image.

In practice the model $p(x, y)$ is unknown. Instead, we have the Compaq Database. It is a collection of samples

$$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$$

where for each $1 \leq i \leq n = 18,696$, $x^{(i)}$ is a color image and $y^{(i)}$ is the associated binary skinness image. We assume that the samples are independent of each other with distribution $p(x, y)$. The collection of samples is referred later as the training data. Probabilities are estimated by using classical empirical estimators and are denoted with the letter q.

In what follows, we build models for the joint probability distribution of color and skinness image using maximum entropy modeling.

3 Baseline Model

3.1 Defining the model

First, we build a model that respects the one pixel marginal observed in the Compaq Database. That is, consider the set of probability distributions $p(x, y)$ that verify:

$$\mathcal{C}_0 : \forall s \in S, \forall x_s \in C, \forall y_s \in \{0, 1\}, p(x_s, y_s) = q(x_s, y_s) \tag{1}$$

In (1), the quantity on the right side of the equal sign is the proportion of pixels with color x_s and label y_s in the training data. The MaxEnt solution under $\mathcal{C}_0$ is the independent model:

$$p(x, y) = \prod_{s \in S} q(x_s, y_s) \tag{2}$$

The proof is postponed to Appendix A. Using Bayes formula, one then obtains:

$$p(y|x) = \prod_{s \in S} q(y_s|x_s) \tag{3}$$

We call the model in (3) the baseline model. It is the most commonly used model in the literature [4] [5].

4 Hidden Markov Model

4.1 Defining the model

The baseline model is certainly too loose and one might hope to get better detection results by constraining it to a model that takes into account the fact that skin zones are not purely random but are made of large regions with regular shapes. Hence, we fix the marginals of y for all the neighboring pixels couples. We use 4-neighbor system for simplicity in all that follows. For 2 neighboring pixels s and t, the expected proportion of times that we observe $(y_s = a, y_t = b)$ should be $q(a, b)$ for $a = 0, 1$ and $b = 0, 1$, the corresponding quantities measured on the training set. We assume that the model is isotropic, aggregating the cases where s and t are in vertical position to the cases where s and t are in horizontal position. Hence let us define the following constraints:

$$\begin{aligned} \mathcal{D} : \forall < s, t > \in S \times S,\ & p(y_s = 0, y_t = 0) = q(0, 0) \text{ and} \\ & p(y_s = 1, y_t = 1) = q(1, 1) \end{aligned} \tag{4}$$

where $< s, t >$ defines a couple of neighbor pixels.

The MaxEnt model under $\mathcal{C}_0 \cap \mathcal{D}$ is then the following Gibbs distribution:

$$p(x, y) \approx \prod_{s \in S} q(x_s|y_s) \exp[\sum_{<s,t>} (a_0(1 - y_s)(1 - y_t) + a_1 y_s y_t)] \tag{5}$$

Here and thereafter, the sign $\approx$ means equality up to a function that might depend on x but not on y. a_0 et a_1 are constant that must be set up such that the constraints are satisfied. The proof is in Appendix A. From (5) one then obtains the following model:

$$p(y|x) \approx \prod_{s \in S} q(x_s|y_s) p(y) \tag{6}$$

with

$$p(y) = \frac{1}{Z(a_0, a_1)} \exp[\sum_{<s,t>} (a_0(1 - y_s)(1 - y_t) + a_1 y_s y_t)] \tag{7}$$

where $Z(a_0, a_1)$ is a normalization function also known in statistical mechanics as the partition function:

$$Z(a_0, a_1) = \sum_y \{\exp[\sum_{<s,t>} (a_0(1-y_s)(1-y_t) + a_1 y_s y_t)]\} \tag{8}$$

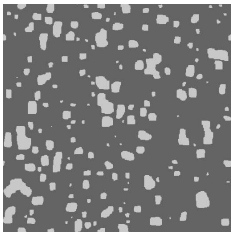
The model in equation (7) is known as a special case of a Potts model, see [7] and [13]. It is a Hidden Markov Model (HMM) if we consider y to be the hidden layer. This model is also simply referred to as a Markov Model elsewhere.

4.2 Parameter estimation

Parameter estimation in the context of MaxEnt is still an active research subject, especially in situations where even the likelihood function cannot be computed for a given value of the parameters. This is the case here since the partition function cannot be evaluated even for very small size images. One line of research consists in approximating the model in order to obtain a formula where the partition function no longer appears: Pseudo-likelihood [14] [15], mean field methods [16] [17], as well as Bethe Trees models [12] are among them. Another possibility is to use stochastic gradient as in [18]. Here we explore a related method based on the concept of Julesz ensembles defined in [19]. We learn from this work that one can sample an image from the model defined in (7) without knowing the parameters a_0 and a_1. This is true only in the asymptotic case of an infinite image but we will apply the result for a large image, say 512x512 pixels. In a second step, we use this sample image in order to estimate the parameters a_0 and a_1. This is done using the quantity $p(y_s = 1|y_{(s)})$ which is the probability to observe the label 1 at pixel s given all the other values y_t, for $t \in S$ and $t \neq s$. For the model in (7), this quantity can be easily analytically computed as

$$p(y_s = 1|y_{(s)}) = \phi((a_1 + a_0)n_s(1) - 4a_0) \tag{9}$$

where $\phi(x) = (1 + e^{-x})^{-1}$ is the sigmoïd (also known as logistic) function and $n_s(1)$ is the number of neighbors of s that take the label 1. This sum can take only five different values. For each one, the quantity $p(y_s = 1|y_{(s)})$ can be estimated from the sample image, leading to five linearly independent equations from which parameters a_0 and a_1 can be estimated. Now, returning to how to obtain a sample from the model in (7). The key idea which originated in statistical physics [20], is that the MaxEnt model we are looking for is, in an appropriate asymptotic meaning, the uniform distribution over the set of images that respect the constraints $\mathcal{D}$. Now, in the absence of phase transition, sampling from this set can be achieved numerically using simulated annealing, see [21].



	database values	image values
$Pr(Y_s = 0, Y_t = 0)$	0.828	0.827991
$Pr(Y_s = 1, Y_t = 1)$	0.159	0.151646

Fig. 1. **Top:** a sample image from the prior distribution used in the Hidden Markov Model. **Bottom:** probabilities estimated from the training set and from the image on the top.

Figure 1 shows a 512×512 sample of the prior model defined in equation (7). One can qualitatively appreciate how well it models skin regions. Notice that vertical and horizontal borders are preferred. This is a bias of the neighborhood system. Choosing 8 neighbors could improve it at the expense of computational load. The quantities $Pr(Y_s = y_s, Y_t = y_t)$, for neighboring pixels s and t are presented in Figure 1, first, as estimated from the training set, and secondly, as estimated from the image in the same Figure. The constraints are nearly respected. Parameter estimation from the image in Figure 1 leads to the numerical values: $a_0 = 3.76$ and $a_1 = 3.94$.

5 First Order Model

5.1 Defining the model

The baseline model was built in order to mimic the one pixel marginal of the joint distribution of color and skinness as observed on the database. Then, in building the HMM model we added constraints on the prior skinness distribution in order to smooth the model. Now, we constrain once more the MaxEnt model by imposing the two-pixel marginal that is $p(x_s, x_t, y_s, y_t)$, for 4-neighbor s and t, to match those observed in the training data. Hence we define the following constraints:

$$\begin{aligned} \mathcal{C}_1 : \forall < s,t > \in S \times S, \forall x_s \in C, \forall x_t \in C, \forall y_s \in \{0,1\}, \forall y_t \in \{0,1\}, \\ p(x_s, x_t, y_s, y_t) = q(x_s, x_t, y_s, y_t) \end{aligned} \tag{10}$$

The quantity $q(x_s, x_t, y_s, y_t)$ is the expected proportion of times we observe the values (x_s, x_t, y_s, y_t) for a couple of neighboring pixels, regardless of the orientation of the pixels s and t in the training set.

Clearly, $\mathcal{C}_1 \subset (\mathcal{C}_0 \cap \mathcal{D}) \subset \mathcal{C}_0$. The solution to the MaxEnt problem under $\mathcal{C}_1$ is then, see Appendix A, the following Gibbs distribution:

$$p(x, y) \approx \exp[\sum_{<s,t>} \lambda(x_s, x_t, y_s, y_t)] \tag{11}$$

where $\lambda(s, t, x_s, x_t, y_s, y_t)$ are parameters that should be set up to satisfy the constraints. From (11), one gets

$$p(y|x) \approx \exp[\sum_{<s,t>} \lambda(s, t, x_s, x_t, y_s, y_t)] \tag{12}$$

Assuming that one color can take 256^3 values, the total number of parameters is $256^3 \times 256^3 \times 2 \times 2$. The previously mentioned parameter estimation methods clearly do not apply. In [12], the authors present a tree approximation to the pixel grid, called "Bethe tree", after the physicist H.A. Bethe who used trees in statistical mechanics problems. Bethe trees permit us to compute analytically an approximation of the parameters in the model (11) and consequently in (12) as we shall see now.

5.2 Parameter estimation and Bethe Tree Approximation

Bethe tree have been introduced in computer vision as a way of approximating estimators in Markov Random Field models in [12]. We shall revisit this work in connection with maximum entropy models. The key idea is to provide a tree that approximates locally the pixel lattice. More precisely, for each pixel s, we consider a sequence of trees $\mathcal{T}_1^{(s)}, \mathcal{T}_2^{(s)}, \ldots$ of increasing depth. The construction is as follows: the root node of the tree is associated with s. For each neighbor t of s in the pixel-graph, a child node indexed by t is added to the root node. This defines $\mathcal{T}_1^{(s)}$. Subsequently, for each u, neighbor of a neighbor of s, (excluding s itself), a grandchild node indexed by u is added to the appropriate child node. This defines $\mathcal{T}_2^{(s)}$, and so on, see [12] for a detailed account. An important remark is that a single pixel might lead to several different nodes in the tree! For example $\mathcal{T}_2^{(s)}$ is built with s, the neighbors of s and the neighbors of these. Using 4-neighbors, and assuming that s is not in the border of the image, this makes up 13 pixels, but the associated tree has 17 nodes, 4 pixels being replicated twice each, see Figure 2.

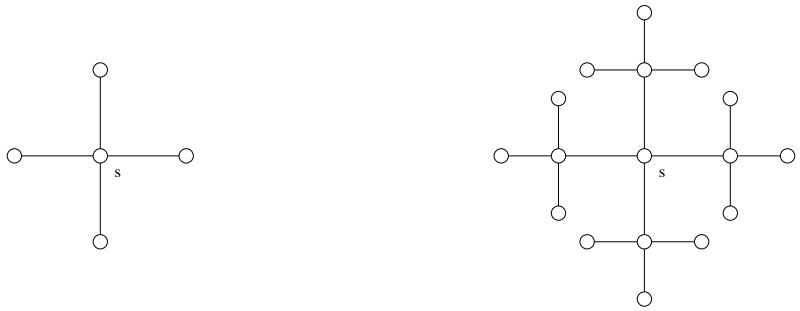


Fig. 2. **Left:** a Bethe tree of depth 1 rooted at s. **Right:** a Bethe tree of depth 2 rooted at s.

Let us consider the following model

$$\begin{aligned} &p(x,y) \approx \exp H(x;y) \text{ with} \\ &H(x;y) = \textstyle\sum_{<s,t>} \log q(x_s, x_t, y_s, y_t) - (n(s)-1) \sum_{s \in \mathring{S}} \log q(x_s, y_s) \end{aligned} \tag{13}$$

where $n(s)$ is the number of neighbors of s and $\mathring{S}$ is the set of interior pixels of S, that is the ones that have exactly four neighbors. First, remark that the model in (13) is a special case of model in (11). Second, under the Beth tree approximation, with arbitrarily finite depth, the model in (13) satisfies the constraints. Indeed, this is a particular case of a more general result, see [22], saying that any pairwise MRF defined on a tree graph can be written as a function of it's marginal distributions as in (13). We can then conclude that under the Bethe Tree approximation, (13) is the MaxEnt solution for $\mathcal{C}_1$.

Now, let us see how in practice one can use the model in (13). As for the HMM model, we fall back on the Markov Chain Monte Carlo algorithm. This requires to compute the conditional distribution of a label y_s given all the other labels and the image of the colors x. For $s \in \mathring{S}$, we obtain

$$\begin{aligned} p(y_s = 1 | y_{(s)}, x) &= \phi(U(x;y)) \text{ with} \\ U(x;y) &= \textstyle\sum_{t \in \mathcal{V}(s)} \log \frac{q(y_s=1, y_t | x_s, x_t)}{q(y_s=0, y_t | x_s, x_t)} - (n(s)-1) \log \frac{q(y_s=1|x_s)}{q(y_s=0|x_s)} \end{aligned} \tag{14}$$

6 Experiments

All experiments are made using the following protocol. The Compaq database contains about 18,696 photographs. It is split into two almost equal parts randomly. The first part, containing nearly 2 billion pixels is used as training data while the other one, the test set, is left aside for ROC curve computation.

Each term of the product on the right side of (3) can be computed using probabilities estimated on the training data as follows using Bayes formula:

$$q(y_s|x_s) = \frac{1}{q(x_s)} q(x_s|y_s)q(y_s) \tag{15}$$

with

$$q(x_s) = \sum_{y_s=0}^{1} q(x_s|y_s)q(y_s)$$

Evaluation of the quantities in (15) is based on two 3-dimension histograms, $q(x_s|y_s = 1)$ and $q(x_s|y_s = 0)$ describing the one pixel color skin regions and non-skin regions respectively. Several authors have tried to get a parametric expression for these histograms as a mixture of Gaussian distribution [6] [1]. Our experience is that the Compaq Database is large enough so that crude histograms made with 512 color value per bin uniformly distributed do not over-fit. Each histogram is then made of 32^3 bins. The ROC curve for this model is presented in Figure 3. Experiments for this model, as well as for the other ones were made using the following protocol. The Compaq database contains about 18,696 photographs. It was split into two almost equal parts randomly. The first part, containing nearly two billion pixels was used as training data while the other one, the test set, was let aside for ROC curve computation. In Figure 4, first column displays test images. The second column displays grey level images. The grey-level is proportional to the quantity $p(y_s = 1|x)$ evaluated with the Baseline model. On the top image, skin pixels are not detected, especially on the neck of the rightmost person. On the bottom image, we notice many false positives. Figure 3 shows ROC curves computed from 100 images (around 10 millions pixels), randomly extracted from the test set. The Baseline model (with crosses) permit to detect more than 80% of the skin pixels with less than 10% of false positive rate.

6.2 Experiments HMM

For a new image x, skin detection requires to compute for each pixel the quantity $p(y_s|x)$. We use Markov Chain Monte Carlo. We generate, using the Gibbs sampler algorithm [7], a sequence of label images

$$y^1, y^2, \ldots, y^{n_0}, \ldots, y^n$$

Algorithm 1. Markov Chain Monte Carlo algorithm

$u \Leftarrow 0$
randomly initialize the binary image y^1
for $j = 1$ to $n - 1$ **do**
  $y \Leftarrow y^j$
  for all $s \in S$ **do**
    if $p(y_s = 1 | y_{(s)}, x) > 0.5$ **then**
      $y_s^{j+1} = 1$
    else
      $y_s^{j+1} = 0$
    end if
  end for
  if $j + 1 > n_0$ **then**
    $u \Leftarrow u + y^{j+1}$
  end if
end for
$u \Leftarrow u/(n - n_0)$

with stationary distribution (6). Then, we estimate the quantity $p(y_s|x)$ by the empirical mean

$$\frac{1}{n - n_0} \sum_{j=n_0+1}^{n} y_s^j$$

The Monte Carlo algorithm used in our experiments is presented in detail in Algorithm 1. Note that u and y are matrices defined on the pixel lattice S and

$$\begin{array}{ll} p(y_s = 1 | y_{(s)}, x) = \frac{p(y|x)}{\sum_{y_s} p(y|x)} = \phi(U(x; y)) \text{ with} \\ U(x; y) \qquad\qquad = \sum_{t \in \mathcal{V}(s)} (a_1 y_t - a_0(1 - y_t)) + \log \frac{q(x_s|y_s=1)}{q(x_s|y_s=0)} \end{array} \tag{16}$$

where ϕ is the logistic function and $\mathcal{V}(s)$ are the neighbors of s. The algorithm is consistent in the sense that as $n \to +\infty, \forall s \in S, u_s \to p(y_s = 1|x)$, see [7].

Our working parameters are $n_0 = 1$ and $n = 100$. Three output images are presented in Figure 4. It compares favorably with the Baseline model. The skin zones detected with the baseline model are generally blended with background false alarms in complex images. The HMM outputs are cleaner with real skin zones emphasized. There is obvious misclassification of non-skin pixels as skin pixels on the dog of the third image for both models. The ROC curve in Figure 3 indicates an increase close to 2% in detection rate for the same false positive rate as the Baseline model. For example, setting 10% of false positive rate, the Baseline model permits to detect 81% of skin pixels in average, while the HMM permits to detect 83% in average. We show now that this is significative. The test set is made of 100 images disjoint from the training set. This amounts to about 10^7 pixels, out of which about 6% are labelled as skin. These 6×10^5 pixels cannot be considered as independent since the color values of the images

are correlated at small distance. Hence, we choose one out of ten of these pixels leading to a sample size of 6×10^4. The standard deviation around the Baseline value is then $\sqrt{0.81(1-0.81)} \times (\sqrt{6 \times 10^4})^{-1} \leq 2 \times 10^{-3}$. The hypothesis that the proportion of 83% was due to random fluctuations is then rejected with a p-value close to 0.

The running time of Algorithm 1 is as follows: there are $n-1$ loops over the image. During each loop, for each pixel, the conditional probability in (16) is evaluated once. The logarithmic operation as well as the logistic function can be tabulated. The labels of the four neighbors as well the color value have to be read from the current image. All these lead to 7 access to look-up tables and 4 additions. Hence the complexity of the algorithm is about $11 \times 100 \times |S|$ operations for an image made of $|S|$ pixels.

6.3 Experiments FOM

Now let us see how each term in (14) can be evaluated. First,

$$\frac{q(y_s = 1|x_s)}{q(y_s = 0|x_s)} = \frac{q(x_s|y_s = 1)}{q(x_s|y_s = 0)} \frac{q(y_s = 1)}{q(y_s = 0)} \tag{17}$$

and the quantities on the right side of (17) are easily obtained from the database as before. Second,

$$\frac{q(y_s = 1, y_t|x_s, x_t)}{q(y_s = 0, y_t|x_s, x_t)} = \frac{q(x_s, x_t|y_s = 1, y_t)}{q(x_s, x_t|y_s = 0, y_t)} \frac{q(y_s = 1, y_t)}{q(y_s = 0, y_t)} \tag{18}$$

Now the quantities on the right side of (18) involving the color values cannot be directly extracted from the database without drastic over-fitting since the histogram involved have a support of dimension six. Hence some kind of dimension reduction is needed.

One natural solution is to assume conditional independence, that is

$$\frac{q(x_s, x_t|y_s = 1, y_t)}{q(x_s, x_t|y_s = 0, y_t)} = \frac{q(x_s|y_s = 1)}{q(x_s|y_s = 0)} \tag{19}$$

The obtained model is then a HMM model, as in equation (6). Hence, Bethe tree method gives another way to estimate parameters a_0 and a_1. Obtained values are $a_0 = 3.94$ and $a_1 = 4$, which are close to the values obtained in section 4. The performances obtained with these values are not distinguishable

to the ones obtained previously, which give some indication of the robustness of the model.

A more promising dimension reduction procedure is the following approximation:

$$q(x_s, x_t|y_s, y_t) \sim q(x_s|y_s)q(x_t - x_s|y_s, y_t) \tag{20}$$

That is, we assume that the color gradient at s, measured by the quantity $x_t - x_s$, is, given the labels at s and t, independent of the actual color x_s. Evaluation of the right side of the sign $\sim$ requires to compute 6 histograms with a support of dimension 3 only. We use 32^3 bins of 512 colors each. Then we have:

$$\begin{aligned} U(x;y) &= \textstyle\sum_{t\in\mathcal{V}(s)} \log \frac{q(x_s|y_s=1)q(x_t-x_s|y_s=1,y_t)q(y_s=1,y_t)}{q(x_s|y_s=0)q(x_t-x_s|y_s=0,y_t)q(y_s=0,y_t)} \\ &\quad -(n(s)-1)\log\tfrac{q(x_s|y_s=1)q(y_s=1)}{q(x_s|y_s=0)q(y_s=0)} \\ &= \textstyle\sum_{t\in\mathcal{V}(s)} \log \frac{q(x_t-x_s|y_s=1,y_t)q(y_s=1,y_t)}{q(x_t-x_s|y_s=0,y_t)q(y_s=0,y_t)} + \log\tfrac{q(x_s|y_s=1)}{q(x_s|y_s=0)} \\ &\quad -(n(s)-1)\log\tfrac{q(y_s=1)}{q(y_s=0)} \end{aligned} \tag{21}$$

Experiments with this model are presented in Figures 3 and 4. The setup is the same as for the HMM. In Figure 4, one can visually appreciate the improvement in localization of the skin zones compared to the HMM. The detected skin regions are more precise. It is easier to recognize the shapes of the faces and hands than with the HMM results. The mouth of the right hand character in the first image is not detected as skin, as well as the eyes in the second image or the mustache in the third image.

Bulk results in the ROC curve of Figure 3 show an improvement of performance of around 1%. At 10% of false positive rate, the HMM permits to detect around 83% of skin pixels and the First Order Model around 84%. This is evaluated in the same setting as described in Section 6.2. In particular, the number of independent skin pixels is around 6×10^4. The standard deviation around the HMM value is then $\sqrt{0.83(1-0.83)} \times (\sqrt{6\times 10^4})^{-1} \leq 2 \times 10^{-3}$. The hypothesis that the proportion of 84% was due to random fluctuations is then rejected with a p-value close to 0.

Another to compare classification algorithms over multiple thresholding values is to compute the area under the roc curve (AUC). Using $[.04; .11]$ for integration interval, the normalized AUC, that is, the AUC divided by the length of the interval of integration is .79 for the baseline model, .81 for HMM and .82 for FOM confirming the results obtained above for a single false positive rate.

The running time for the FOM can be evaluated in the same way as HMM. The only difference is the operations involved in the $U(x;y)$ function in (21). As for the HMM, the logarithmic operation as well as the logistic function can

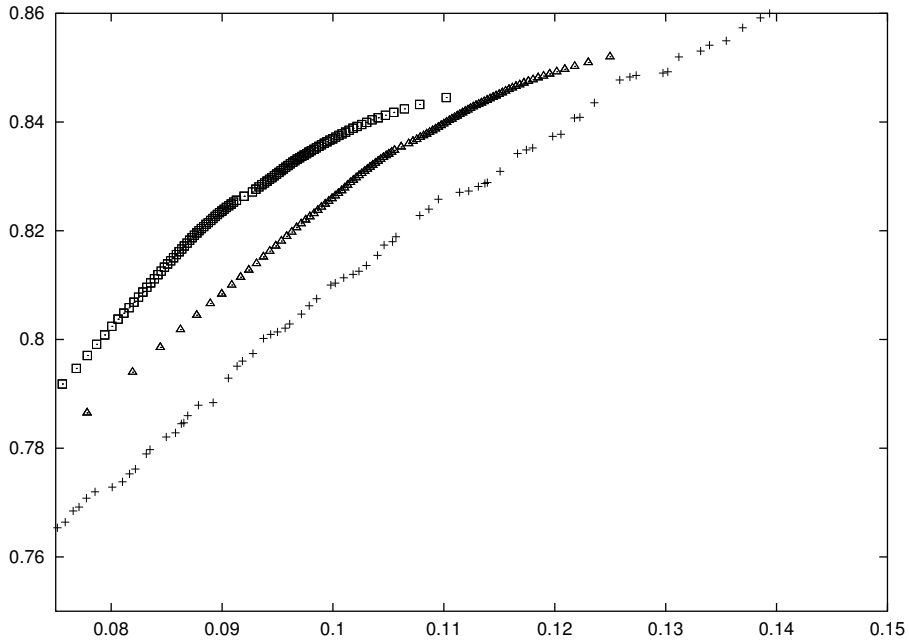


Fig. 3. Receiver Operating Characteristics (ROC) curve for each model. x-axis is the false positive rate, y-axis is the detection rate. Baseline model is shown with crosses, HMM model with triangles, while the First Order Model is shown with squares.

be tabulated. The values of the current pixel and its four neighbors have to be read from the current image. All these lead for an image to about 15 access to look-up tables and 30 additions/subtractions and 1 multiplication. Hence the complexity of the algorithm is about $46 \times 100 \times |S|$ operations which is about 4 times the running time of the HMM. As an example, using a PC with a Pentium 4 processor at 1.7 Ghz and 256 MB memory, the processing time for a 100×100 pixels image is .008 seconds for the baseline model, 1.3 seconds for the HMM and 2.3 seconds for the FOM.

7 Conclusions

We have considered a sequence of three models for skin detection built from a large collection of labeled images. For a given color image, such a model puts weight on binary images defined on the same pixel grid. Each model is a maximum entropy model with respect to constraints. These constraints concern marginal distributions. Our models are nested. The first model, called the baseline model is well known from practitioners. Pixels are considered as independent. Performance, measured by the ROC curve on the Compaq database is impressive for such a simple model. However, single image examination reveals very irregular results. The second model is a Hidden Markov Model. It includes constraints that force smoothness of the solution. The ROC curve

Fig. 4. **First column:** original color images. The image on top is 225×180 pixels . The image on the bottom is 541×361 pixels. **Second column:** Baseline model. **Third column:** hidden Markov model. **Fourth column:** First Order Model. In the computed images, the grey level is proportional to the skin probability evaluated with the specified model.

obtained shows an increase in detection rate from 81% to 83% for the same false positive rate of 10%. Finally, color gradient is included in the set of constraints. Thanks to Bethe tree approximation, we obtain a simple analytical expression for the coefficients of the associated MaxEnt model. The resulting detection rate increases to 84%. The same qualitative behavior is observed when comparing the area under the ROC curve.

For many applications involving skin detection as an intermediate stage, processing time is of major importance. In future work we plan to replace the stochastic sampling algorithm by a deterministic scheme as Mean Field method [16] or Belief Propagation [23] method in order to meet the required time constraints.

Detailed examination of the pictures reveals that the discussed models are still far from reaching human performances. For example, the left arm of the right-most person in the first image of Figure 4 is visible in the baseline model and not in the subsequent ones. Remark that the grey values indicating the probability for skin are very low. A zoom is provided in Figure 5. It is understandable that the regularizing models, HMM as well as the First Order Model, operating at the level of pixels, have produced a posterior probability that put very low likelihood for skin in this region. Indeed, the local evidence for skin is low and the neighboring values are also indicating low evidence. A high level model of limbs might be able to overcome these difficulties.

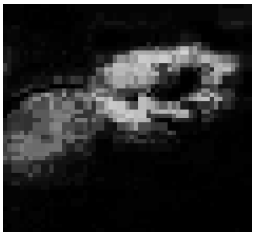


Fig. 5. Zoom of top row, second image in Figure 4. Result of the Baseline model

8 Acknowledgments

We would like to thank the reviewing work. It has helped in improving the overall quality of the paper.

A Appendix

Here we shall derive a MaxEnt solution for the joint distribution $p(x, y)$ under the constraints $\mathcal{C}_0$. See (1).

Remark that the constraints in (1) are expectations with respect to p. Indeed,

$$p(x_s, y_s) = E_p[\delta_{x_s}(X_s)\delta_{y_s}(Y_s)] \tag{A.1}$$

with

$$\delta_a(b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ if } a \neq b \end{cases}$$

Then, following Jaynes' argument [9], the MaxEnt solution under $\mathcal{C}_0$ is unique if it exists, and can be obtained using Lagrange multipliers. One gets:

$$p(x, y) = \exp(\lambda_0 + \sum_{s \in S} \lambda(s, x_s, y_s)) \tag{A.2}$$

Where the parameters λ should be set up such that the constraints are satisfied. Now if

$$\forall x_s \in C, \forall y_s \in \{0, 1\}, q(x_s, y_s) > 0 \tag{A.3}$$

then one can choose

$$\lambda_0 = 0 \text{ and } \lambda(s, x_s, y_s) = \log q(x_s, y_s) \tag{A.4}$$

which leads to the unique solution of the MaxEnt problem:

$$p(x, y) = \prod_{s \in S} q(x_s, y_s) \tag{A.5}$$

Condition in (A.3) is saying that there is no empty bin in the empirical joint histogram $q(x_s, y_s)$. This will be our case. MaxEnt solutions still exist when (A.3) is not verified.

Here we shall obtain a MaxEnt solution for the joint distribution $p(x, y)$ under $\mathcal{C}_0 \cap \mathcal{D}$, see (1) and (4).

As for $\mathcal{C}_0$, the constraints in $\mathcal{D}$ are expectations. Indeed,

$$\forall y_s \in \{0,1\}, \forall y_t \in \{0,1\}, p(y_s, y_t) = E_p[\delta_{y_s}(Y_s)\delta_{y_t}(Y_t)] \tag{A.6}$$

Using once more Lagrange multipliers, one obtains that the MaxEnt solution, if it exists, is

$$\begin{aligned} p(x,y) &= \exp H(x,y,\lambda_0,\lambda_1,\lambda_2,\lambda_3) \text{ with} \\ H(x,y,\lambda_0,\lambda_1,\lambda_2,\lambda_3) &= \lambda_0 + \textstyle\sum_{s\in S} \lambda_1(s,x_s,y_s) + \\ &\quad \textstyle\sum_{<s,t>\in S\times S} \lambda_2(s,t)(1-y_s)(1-y_t) + \\ &\quad \textstyle\sum_{<s,t>\in S\times S} \lambda_3(s,t) y_s y_t] \end{aligned} \tag{A.7}$$

where $< s, t >$ is a couple of 4-neighbors pixels and λ_0, λ_1, λ_2, λ_3 define parameters that should be set up such that the constraints are satisfied. Starting from (A.7),remark that

$$p(x_s, y_s) = \sum_{x_t; t\in S, t\neq s} \sum_{y_t; t\in S, t\neq s} p(x,y) = \exp[\lambda_0 + \lambda_1(s, x_s, y_s)] g(s, y_s) \tag{A.8}$$

with $g(s, y_s)$ a function that doesn't depend on x_s. Now,

$$p(y_s) = \sum_{x_s} p(x_s, y_s) = \exp[\lambda_0] g(s, y_s) \sum_{x_s} \exp[\lambda_1(s, x_s, y_s)] \tag{A.9}$$

hence

$$p(x_s|y_s) = \frac{p(x_s, y_s)}{p(y_s)} = \frac{\exp[\lambda_1(s, x_s, y_s)]}{\sum_{x_s} \exp[\lambda_1(s, x_s, y_s)]} \tag{A.10}$$

Since $p(x, y)$ lies in $\mathcal{C}_0$, it verifies: $p(x_s|y_s) = q(x_s|y_s)$. Assuming positivity (A.3), we can choose

$$\lambda_1(s, x_s, y_s) = \log q(x_s|y_s) \tag{A.11}$$

Now, constraints in $\mathcal{D}$, see (4), do not depend on the location $< s, t >$. Hence, one can reduce to translation invariant models as in (5).

Constraints in $\mathcal{C}_1$, see (10) are also expectations. Indeed,

$$p(x_s, x_t, y_s, y_t) = E_p[\delta_{(x_s)}(X_s)\delta_{(x_t)}(X_t)\delta_{(y_s)}(Y_s)\delta_{(y_t)}(Y_t)] \tag{A.12}$$

Using Lagrange multipliers, one obtains (11).

References

[1] J.-C. Terrillon, M. N. Shirazi, H. Fukamachi, S. Akamatsu, Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images, in: Fourth International Conference On Automatic Face and gesture Recognition, 2000, pp. 54–61.

[2] J. Z. Wang, J. Li, G. Wiederhold, O. Firschein, System for screening objectionable images, Images, Computer Communications Journal 21 (15) (1998) 1355–1360.

[3] J. Z. Wang, J. Li, G. Wiederhold, O. Firschein, Classifying objectionable websites based on image content, Notes in Computer Science, Special issue on iteractive distributed multimedia systems and telecommunication services 21/15 (1998) 113–124.

[4] J.-C. Terrillon, M. David, S. Akamatsu, Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments, in: IEEE Third International Conference on Automatic Face and gesture Recognition, 1998, pp. 112–117.

[5] M. J. Jones, J. M. Rehg, Statistical color models with application to skin detection, Tech. Rep. CRL 98/11, Compaq (1998).

[6] M. Jones, J. M. Rehg, Statistical color models with application to skin detection, in: Computer Vision and Pattern Recognition, 1999, pp. 274–280.

[7] G. Winkler, Image Analysis, Random Fields and Dynamic Monte Carlo Methods, Springer-Verlag, 1995.

[8] R. Chellappa, A. Jain (Eds.), Markov Random Fields: Theory and Applications, Academic Press, 1996.

[9] E. Jaynes, Probablity theory: The logic of science., http://omega.albany.edu:8008/JaynesBook, chapter 11.

[10] Cover, Thomas, Elements of Information Theory, Wiley, 1991, chapter 11.

[11] S. Zhu, Y. Wu, D. Mumford, Filters, random fields and maximum entropy (frame): towards a unified theory for texture modeling, International Journal of Computer Vision 27 (2) (1998) 107–126.

[12] C. Wu, P. C. Doerschuk, Tree approximations to markov random fields, IEEE Trans. on PAMI 17 (4) (1995) 391–402.

[13] G. Cross, A. Jain, Markov random field texture models, IEEE Trans. on PAMI 5 (1) (1983) 25–39.

[14] J. Besag, On the statistical analysis of dirty pictures, Journal of the Royal Statistical Society, B 48 (3) (1986) 259–302.

[15] F. Divino, A. Frigessi, Penalized pseudolikelihood inference in spatial interaction models with covariates, Scandinavian Journal of Statistics 27 (3) (2000) 445–458.

[16] J. Zhang, The mean field theory in em procedure for markov random fields, IEEE Trans. on Signal Processing 40 (10) (1992) 2570–2583.

[17] G. Celeux, F. Forbes, N. Peyrard, Em procedures using mean field-like approximations for markov model-based image segmentation, Pattern Recognition 36 (1) (2002) 131–144.

[18] L. Younes, Estimation and annealing for gibbsian fields, Annales de l'Institut Henry Poincaré, Section B, Calcul des Probabilités et Statistique 24 (1998) 269–294.

[19] Y. Wu, S. Zhu, X. Liu, Equivalence of julesz ensemble and frame models, International Journal of Computer Vision 38 (3) (2000) 247–265.

[20] A. Martin-Lof, The equivalence of ensembles and gibbs'phase rule for classical lattice-systems, Journal of Statistical Physics 20 (1979) 557–569.

[21] S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, IEEE Trans. on PAMI 6 (1984) 721–741.

[22] J. Pearl, Probabilistic Reasoning in intelligent systems: networks of plausible inference, Morgan Kaufmann, 1988.

[23] J. S. Yedida, W. T. Freeman, Y. Weiss, Understanding belief propagation and it's generalisations, Tech. Rep. TR-2001-22, Mitsubishi Research Laboratories (January 2002).

Chapter 12

A computational neurodegenerative disease progression score: method and results with the Alzheimer's Disease Neuroimaging Initiative cohort

NeuroImage 63 (2012) 1478–1486

Contents lists available at SciVerse ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg

A computational neurodegenerative disease progression score: Method and results with the Alzheimer's disease neuroimaging initiative cohort

Bruno M. Jedynak [a,b,*], Andrew Lang [c], Bo Liu [a], Elyse Katz [d], Yanwei Zhang [d], Bradley T. Wyman [d], David Raunig [d,1], C. Pierre Jedynak [e], Brian Caffo [f], Jerry L. Prince [c] for the Alzheimer's Disease Neuroimaging Initiative [2]

[a] Department of Applied Math and Statistics, Johns Hopkins University, Baltimore, MD 21218, USA
[b] Center for Imaging Science, Johns Hopkins University, Baltimore, MD 21218, USA
[c] Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218, USA
[d] Pfizer Inc., Groton, CT 06340, USA
[e] Self, Paris, 75011, France
[f] Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD 21205, USA

ARTICLE INFO

Article history:
Accepted 29 July 2012
Available online 3 August 2012

Keywords:
Neurodegenerative diseases
Alzheimer's disease
Biomarkers
Disease progression score

ABSTRACT

While neurodegenerative diseases are characterized by steady degeneration over relatively long timelines, it is widely believed that the early stages are the most promising for therapeutic intervention, before irreversible neuronal loss occurs. Developing a therapeutic response requires a precise measure of disease progression. However, since the early stages are for the most part asymptomatic, obtaining accurate measures of disease progression is difficult. Longitudinal databases of hundreds of subjects observed during several years with tens of validated biomarkers are becoming available, allowing the use of computational methods. We propose a widely applicable statistical methodology for creating a disease progression score (DPS), using multiple biomarkers, for subjects with a neurodegenerative disease. The proposed methodology was evaluated for Alzheimer's disease (AD) using the publicly available AD Neuroimaging Initiative (ADNI) database, yielding an Alzheimer's DPS or ADPS score for each subject and each time-point in the database. In addition, a common description of biomarker changes was produced allowing for an ordering of the biomarkers. The Rey Auditory Verbal Learning Test delayed recall was found to be the earliest biomarker to become abnormal. The group of biomarkers comprising the volume of the hippocampus and the protein concentration amyloid beta and Tau were next in the timeline, and these were followed by three cognitive biomarkers. The proposed methodology thus has potential to stage individuals according to their state of disease progression relative to a population and to deduce common behaviors of biomarkers in the disease itself.

Introduction

Neurodegenerative diseases such as Alzheimer's disease (AD), Parkinson disease (PD), Huntington disease (HD) and amyotrophic lateral sclerosis (ALS) involve the loss of structure or function of neurons, including neuronal death (see Martin (2002); Shaw (2005)). During the earliest stages of these diseases, the progression is slow, on the time scale of years, (see Sperling et al. (2011) for the case of AD). It is widely believed that these early stages are the most promising for therapeutic intervention, before irremediable neuronal loss occurs. Developing a therapeutic remedy requires a precise measure of disease progression, i.e., a quantity which would be specific to a particular disease and sensitive to subtle changes. However, obtaining accurate measures of disease progression during the earliest phases of the disease is difficult. Indeed, these phases are essentially non-symptomatic and the clinical tests which characterize the acute phase of the disease are not sensitive enough to qualify as a measure of disease progression. In response, the medical research community has contributed to developing and validating biomarkers. Biomarkers for neurodegenerative diseases include protein counts (in the cerebrospinal fluid), blood analysis, brain imaging, including molecular and MR, genetic analysis and neuropsychological tests. Structural imaging biomarkers are unique in that they allow one to characterize the size, shape, and health of various brain substructures at the organ level while being noninvasive (see e.g. Qiu et al. (2008) for AD, Rizk-Jackson et al. (2011) for HD). Functional imaging provides a spatially localized image of the physiological

* Corresponding author at: Whitehead 208B, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD, 21218, USA. Fax: +1 410 516 7459.
E-mail address: bruno.jedynak@jhu.edu (B.M. Jedynak).
[1] Present address: ICON Medical Imaging, Warrington, PA 18976, USA.
[2] Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.ucla.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.ucla.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

1053-8119/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.neuroimage.2012.07.059

processes occurring in the brain. See Brooks and Pavese (2011) for a review of imaging biomarkers in PD and Turner et al. (2011) for ALS. Due to the complexity of the neurodegenerative diseases and variabilities within the human population, research efforts have been pooled in order to create datasets with a large number of subjects, time-points and biomarkers. The Alzheimer's Disease Neuroimaging Initiative (ADNI), see http://adni.loni.ucla.edu/, was launched in 2003 by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, the Food and Drug Administration, private pharmaceutical companies and non-profit organizations, as a $60 million, 5-year public/private partnership. A related effort is taking place for PD. The Parkinson Progression Marker Initiative (PPMI), see http://www.ppmi-info.org/, is a comprehensive observational, international, multicenter study designed to identify PD progression biomarkers both to improve understanding of disease etiology and course and to provide crucial tools to enhance the likelihood of success of PD modifying therapeutic trials. Huntington disease is caused by a mutation in a single gene, HTT, with full penetrance, making it feasible to identify presymptomatic individuals who will develop the disease but do not show yet any clinical symptoms, see Hayden (1981). At least two large studies (Predict-HD, see https://www.predict-hd.net/ and TrackOn-HD, see http://hdresearch.ucl.ac.uk/current-studies/trackon-hd/) are underway to identify sensitive biomarkers for HD. Similar efforts are recently taking place for ALS, see Turner et al. (2009); Labbe (2012). The availability of large datasets for neurodegenerative diseases opens new opportunities for computational methods which could have a strong impact in the study, the development of therapeutics and the follow-up of patients with neurodegenerative diseases.

We present in this article a generic computational method for computing a disease progression score (DPS) by combining biomarkers. ADNI is, as of today, the largest publicly available longitudinal dataset of biomarkers related to a neurodegenerative disease. It is therefore the dataset which we have chosen to evaluate our method. Since we will work with the ADNI dataset, we recall some preliminary information on AD as well as the validated biomarkers for AD in Section 2. The method for computing a DPS, which is the main contribution of this paper, is presented in Section 3. Results with the ADNI dataset appear in Section 4 and finally in Section 5, we discuss the results in the context of ADNI, and their consequence in the study of AD and other neurodegenerative diseases.

Alzheimer's disease

Although this paper describes a method applicable to any neurodegenerative disease, our current evaluation involves the ADNI dataset and therefore it is informative to use this disease as a framework for motivating the method. The classical characterization of late-onset Alzheimer's disease progression is a time-ordered succession of three stages: normal (N), mild cognitive impairment (MCI), and AD. Physical measurements of disease progression, i.e., *biomarkers*, are used to classify patients into these three stages, but it has been challenging to reliably define finer stages of the disease. As a result, staging of the disease remains coarse and the evaluation of therapies are difficult at the earliest stages when intervention is most likely to be effective, see Hampel et al. (2008).

Cognitive biomarkers such as the clinical dementia rating sum-of-boxes (having scores from 0 to 18) and the mini-mental state exam (having integer scores from 0 to 30) have finer discrete levels, see Berg et al. (1988); Folstein et al. (1975). But it has been reported in Mungas and Reed (2000) and Duara et al. (2011) that these measurements have poor dynamic range in the earliest stages of AD. On the other hand, Mosconi et al. (2007) has shown that the early stages of AD can be characterized using both imaging and biochemical biomarkers. Following these observations, Jack et al. (2010) proposed that there is a single disease progression and that different biomarkers characterize the disease during different stages. They hypothesized the biomarker changes and disease progression shown in Fig. 1 (reproduced with permission from Jack et al. (2010)). In this hypothesized model, the amyloid beta ($A\beta_{42}$) protein changes first, followed by changes in the protein Tau, then structural changes in the brain (gray matter loss), and lastly a deterioration of cognitive function resulting in dementia. Based on Fig. 1 we expect to find that no single biomarker has the dynamic range to cover the full spectrum of the disease. Given the limitations of any single biomarker, there is likely benefit in developing methods that can combine multiple biomarkers in a nonlinear fashion in order to represent—using a single measure—progression throughout the entire disease. This is a key motivation for the process we report in this paper. An important byproduct of this effort is a plot similar to that of Fig. 1, but derived from data using multiple biomarkers which reveal key differences in the ordering of the biomarker dynamics over the course of disease.

Method

Principles for temporal standardization of multiple biomarkers

The available data are longitudinal measurements of multiple biomarkers for hundreds of subjects. Our research first describes and then evaluates a disease progression score, notated DPS, which standardizes subject time-lines onto a common temporal scale. The DPS

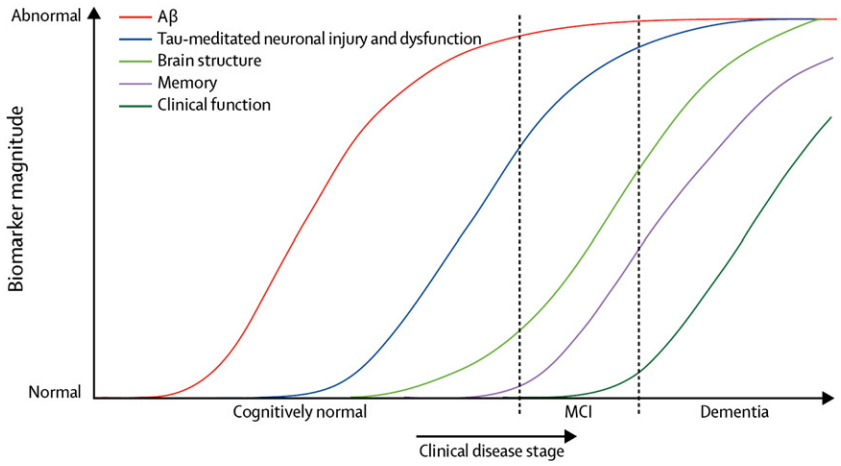


Fig. 1. This graph represents a conceptualization of the timing of key biomarkers transitions from "Normal" to "Abnormal" as subjects go through the three stages of Alzheimer's disease: "Cognitively Normal", "MCI", and "Dementia." This plot is reproduced from "Hypothetical model of dynamic biomarkers of the Alzheimer's pathological cascade," Jack CR Jr, Knopman DS, Jagust WJ, Shaw LM, Aisen PS, Weiner MW,Petersen RC, Trojanowski JQ., Lancet Neurol. 2010 Jan;9(1):119-28.

serves as a new (derived) biomarker enabling both disease staging in single subjects and a data-driven characterization of biomarker dynamics in the entire population.

The method we use to achieve standardization is based on three assumptions:

1. All subjects follow a common disease progression but differ in their age of onset and rate of progression;
2. As the disease progresses, each biomarker changes continuously and monotonically following a sigmoid shaped curve; and
3. In the longitudinal period over which biomarkers are observed, the rate of progression of a given subject is constant.

The proposed computation assigns to each subject and each time-point a score denoted the DPS. Note that all subjects are expected to undergo the same biological and cognitive changes when they reach the same DPS.

Statistical model for DPS

The age t of subject i is to be transformed into the DPS s_i as follows

$$s_i(t) = \alpha_i t + \beta_i \tag{1}$$

upon estimation of the subject dependent parameters α_i and β_i, which indicate rate and onset of disease, respectively. A linear transformation is justified when the interval over which longitudinal observations of subjects occur is short relative to disease duration (true at present in the ADNI database). This could be generalized to nonlinear functions in the case of cohorts with longer longitudinal base. Our objective is to standardize all I subjects by estimating $\alpha = (\alpha_1, \ldots, \alpha_I)$ and $\beta = (\beta_1, \ldots, \beta_I)$. The subject dependent parameters α and β are deliberately modeled as fixed effects, not random effects, as the DPS may ultimately be used as a covariate.

The longitudinal dynamic of each biomarker is assumed to be the same across the population and can be represented as a sigmoidal function f of DPS s. Sigmoidal functions capture the relative quiescent states of a biomarker in the early and late parts of the disease progression while being parsimonious. Using $\theta_k = (a_k, b_k, c_k, d_k)$ to represent the vector of sigmoid function parameters for the k-th biomarker, we can write the form of the k-th biomarker as

$$f(s;\theta_k) = a_k\left(1 + e^{-b_k(s-c_k)}\right)^{-1} + d_k. \tag{2}$$

The minimum and maximum values of the sigmoid function are d_k and $d_k + a_k$, and the value of s for which the biomarker is the most dynamic, having maximum slope $a_k b_k/4$ corresponding to its inflection point, is c_k. A closely related model is the trilinear model in Brooks et al. (1993). Caroli et al. (2010) and Sabuncu et al. (2011) noticed that sigmoids offer a parsimonious parametric model which is often a better fit than linear models for biomarkers. Sigmoids are also similar in form to the conceptual evolution of biomarkers envisioned in Jack et al. (2010) for AD (Fig. 1). Among parametric models, alternatives include the generalized sigmoid in Richards (1959) and polynomials of low order.

Databases for neurodegenerative diseases contain measurements y_{ijk} of biomarker k for subject i at visit j. Since there are often irregularities in data collection, we use $\mathcal{I}$ to denote the set of triples (i, j, k) for which measurements are available. Each biomarker observation can then be written as

$$y_{ijk} = f\left(\alpha_i t_{ij} + \beta_i; \theta_k\right) + \sigma_k \epsilon_{ijk}, \quad (i,j,k) \in \mathcal{I}, \tag{3}$$

where t_{ij} is the age of subject i at visit j. Observation noise in each biomarker is modeled for simplicity by the product of ϵ_{ijk}, which are independent random variables with zero mean and unit variance. σ_k is the standard deviation of biomarker k. The collection of standard deviations $\sigma = (\sigma_1, \ldots, \sigma_K)$ comprise another unknown that must be estimated.

The unknowns in this problem are α, β, θ, and σ and the least squares problem associated with the observation model in (3) is

$$l(\alpha, \beta, \theta, \sigma) = \sum_{(i,j,k)\in\mathcal{I}} \log\sigma_k + \frac{1}{2\sigma_k{}^2}\left(y_{ijk} - f\left(\alpha_i t_{ij} + \beta_i; \theta_k\right)\right)^2 \tag{4}$$

Parameter fitting

Parameter fitting is performed using alternating least squares wherein the parameters θ, α, β, and σ are optimized iteratively starting from the values computed in the previous step. The details of the fitting algorithm are shown in Alg. 1. Because of the additive form of (4), optimization over θ is done serially over each of the K biomarkers. Similarly, optimization over (α, β) is performed serially over each of the I subjects. Fitting of θ, α, and β requires optimization of continuously differentiable nonconvex functions, which is carried out using the Levenberg–Marquardt algorithm (Lines 4 and 8), see Levenberg (1944). $\mathcal{I}_k$ (line 4) is the number of subjects and visits available for biomarker k. The denominator in the equation of Line 5 is the number of degrees of freedom. Because unconstrained optimization can produce unfeasible parameters, parameters are projected onto the feasible space after the main loop (Lines 12–16), see (5) below. This does not change the value of the objective function in (4). Our experiments presented in Section 4 confirm that successful fitting is accomplished in 15 iterations for the ADNI dataset; i.e., $L = 15$ on Line 2, standard optimization stopping criteria can be used otherwise. The parameters α and β are centered and rescaled in Lines 17–19 in Alg. 1 for identifiability reasons which are explained in the next section.

Identifiability

The units of DPS are arbitrarily defined, which implies that we must choose two specific numerical values in order to fully specify the DPS. This situation is analogous to the selection of a scale for temperature, where the numerical values of the freezing and boiling points of water determine the scale. Note that calibration is not specific to the DPS. It is in fact needed for most if not all biomarkers (see Hughes et al. (1982)). In our experiments with ADNI, we chose to fix the DPS such that after computation of DPS for the entire population, the computed DPS for all visits of subjects with normal clinical assessment - subjects of type N -had a median (m_N) and a median absolute deviation (σ_N) which are set respectively to zero and one. This is accomplished in Lines 17–19 in Alg. 1.

Algorithm 1. Algorithm for fitting of the parameters

1: Inititialize $\alpha^{(0)}, \beta^{(0)}$
2: **for** $l = 1$ to L **do**
3: **for** $k = 1$ to K **do**
4: $\theta_k^{(1)} = \arg\min_{\theta_k} \sum_{(i,j)\in\mathcal{I}_k} (y_{ijk} - f(\alpha_i^{(0)} t_{ij} + \beta_i^{(0)}; \theta_k))^2$
5: ${\sigma_k^{(1)}}^2 = \frac{1}{|\mathcal{I}_k - 2I - 4|} \sum_{(i,j)\in\mathcal{I}_k} (y_{ijk} - f(\alpha_i^{(0)} t_{ij} + \beta_i^{(0)}; \theta_k^{(1)}))^2$
6: **end for**
7: **for** $i = 1$ to I **do**
8: $(\alpha_i^{(1)}, \beta_i^{(1)}) = \arg\min_{\alpha_i,\beta_i} \sum_{(j,k)\in\mathcal{I}_i} \frac{1}{\sigma_k^{(1)2}} (y_{ijk} - f(\alpha_i t_{ij} + \beta_i; \theta_k^{(1)}))^2$
9: **end for**
10: $\alpha^{(0)} = \alpha^{(1)}, \beta^{(0)} = \beta^{(1)}$
11: **end for**
12: **for** $k = 1$ to K **do**
13: **if** $b_k < 0$ **then**
14: $a_k^{(1)} = -a_k^{(1)}, b_k^{(1)} = -b_k^{(1)}, d_k^{(1)} = d_k^{(1)} + a_k^{(1)}$
15: **end if**
16: **end for**
17: **for** $i = 1$ to I **do**
18: $\alpha_i^{(1)} = \frac{\alpha_i^{(1)}}{\sigma_N}, \beta_i^{(1)} = \frac{\beta_i^{(1)} - m_N}{\sigma_N}$
19: **end for**

Note that (3) is invariant with respect to the following two transformations, for two constants $\gamma_1 \neq 0$ and γ_2:

$$\begin{aligned}(a_k, b_k, c_k, d_k, \alpha_i, \beta_i, \sigma_k) &\mapsto \left(a_k, \gamma_1 b_k, \gamma_1^{-1} c_k, d_k, \gamma_1^{-1}\alpha_i, \gamma_1^{-1}\beta_i, \sigma_k\right) \\ &\mapsto (a_k, b_k, \gamma_2 + c_k, d_k, \alpha_i, \gamma_2 + \beta_i, \sigma_k)\end{aligned}$$

Note also that the sigmoid function verifies

$$f(t; -a_1, -b_1, c_1, d_1 + a_1) = f(t; a_1, b_1, c_1, d_1) \tag{5}$$

In order to build an identifiable model, we define the restricted parameter set

$$\varrho = \left\{ \rho = (a, b, \alpha, \beta, \sigma); I^{-1}\sum_{i=1}^{I} \alpha_i = \alpha_0, I^{-1}\sum_{i=1}^{I} \beta_i = \beta_0, b_k > 0, a_k \neq 0 \text{ for all } k \in \mathcal{I} \right\}$$

for some $\alpha_0 \neq 0$ and β_0. Necessary conditions on the available data $\mathcal{I}$ for guaranteeing the identifiability of the parameters are as follows:

1. For each biomarker, there is at least one subject i with $\alpha_i \neq 0$ and with at least 4 distinct time-points in $\mathcal{I}$.
2. For each subject, there is at least one biomarker which is available at 2 time points in $\mathcal{I}$

A proof is provided in the Appendix A. In practice, a sufficient number of data points per parameter are needed in order to obtain tight estimators. Examining first the case with no missing data, the number of equations in (3) is IJK. The number of parameters is $2I + 5K$, counting two parameters per subject, and five per biomarkers: four for the sigmoid and one for the standard deviation. In applications where I is large compared to K, the number of data points per parameter is close to $JK/2$. Note that longitudinal data ($J > 1$) is critical for such modeling. However, a small number J of time-points together with a small number K of biomarkers is acceptable. The subset of ADNI that we used in our results has numerous missing data points. Nevertheless, the identifiability conditions are met. The tightness of the estimators of the biomarker parameters is measured using bootstrapping as reported in the Results section.

The ADNI dataset

Data used in the preparation of this article were obtained from the ADNI database (adni.loni.ucla.edu). The ADNI was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations, as a $60 million, 5- year public-private partnership. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). Determination of sensitive and specific markers of very early AD progression is intended to aid researchers and clinicians to develop new treatments and monitor their effectiveness, as well as lessen the time and cost of clinical trials.

The Principal Investigator of this initiative is Michael W. Weiner, MD, VA Medical Center and University of California – San Francisco. ADNI is the result of efforts of many co-investigators from a broad range of academic institutions and private corporations, and subjects have been recruited from over 50 sites across the U.S. and Canada. The initial goal of ADNI was to recruit 800 adults, ages 55 to 90, to participate in the research, approximately 200 cognitively normal older individuals to be followed for 3 years, 400 people with MCI to be followed for 3 years and 200 people with early AD to be followed for 2 years. For up-to-date information, see www.adni-info.org.

The ADNI, ADNI GO, and ADNI 2 biomarker datasets were downloaded from the ADNI server (http://adni.loni.ucla.edu/) on November 24, 2011. The following seven biomarkers were selected for use based on their relevance in assessing the progression of AD. *HIPPO*is the sum of the two lateral hippocampal volumes (Freesurfer version 4.4.0 for longitudinal data http://surfer.nmr.mgh.harvard.edu) normalized by dividing by the intracranial volume. *ADAS* is the Alzheimer's Disease Assessment Scale-cognitive subscale. *MMSE* is the Mini-Mental State Examination score. *TAU* and *ABETA* (our abbreviation for $A\beta_{42}$) are protein levels measured from the cerebrospinal fluid. *CDRSB* is the Clinical Dementia Rating Sum of Boxes score and *RAVLT30* is the Rey Auditory Verbal Learning Test, 30 minute recall. A detailed description of the ADNI population, protocols and biomarkers is provided at http://adni.loni.ucla.edu/. Of the seven biomarkers, only *ADAS* and *RAVLT30* were available at the time of download from the ADNI 2/GO dataset. The protocol for these biomarkers is the same in ADNI, ADNI 2, and ADNI GO. All visits without date information were removed. Subjects not having at least two measurements for at least one of the seven biomarkers were also removed. Finally, subjects not having at least two measurements of the *HIPPO* biomarker were removed. The total number of subjects remaining was 687, where 389 were male, 275 were female, and 23 had unknown gender. The total number of visits was 3658, and the clinical diagnoses at these visits were 1103 N, 1513 MCI, and 1010 AD. There is an average of 26.92 (sd = 5.52) and a minimum of 11 data points available per subject for estimating the parameters of the model.

Results

DPS computed for ADNI subjects

The Alzheimer's DPS (ADPS) was computed for all subject visits in the combined ADNI, ADNI 2, and ADNI GO datasets (with minimal exclusions as was described in Section 5). Seven biomarkers—*HIPPO*, *MMSE*, *TAU*, *ABETA*, *CDRSB*, *RAVLT30*, and *ADAS*—were used together in the computation in order to compute an ADPS score for each visit of each subject (Fig. 2). The initial values (Line 1 of Alg. 1) are obtained as follows: firstly, we set $\alpha^{(0)} \equiv 1$ and $\beta^{(0)} \equiv 0$; secondly, the sigmoids are replaced by linear functions. The main loop (line 2), is then executed 15 times. In this case, the optimization problems in lines 4 and 8 are least squares problems which are solved exactly. At the end of this initialization step, $\alpha^{(0)}$ and $\beta^{(0)}$ are set to the corresponding values obtained and the sigmoids are initialized using the linear fits. The running time of the Algorithm 1, which was coded in Matlab, was 125 seconds using an Intel Core i7 Q820 running at 1.73 GHz (quadcore). In Fig. 2, overall, N subjects (black) have the smallest ADPS, MCI subjects (red) have moderate ADPS, and AD subjects (green) have the largest ADPS. Lower ADPS scores are therefore consistent with the normal population and higher ADPS scores are indicative of increased presence of dementia. Those subjects whose clinical status changes from MCI to AD (blue) are found mostly between the red and green colors.

The estimated sigmoidal behaviors of each biomarker were also computed as part of the normalization process (gray curves on each plot in Fig. 2). It is observed that individual subject trajectories fall near these curves and have similar slopes in most cases. This is expected due to the nature of the optimization criterion used to define ADPS. However, since ADPS is computed as a joint optimization considering all seven biomarkers, some data falls fairly far from the estimated characteristic biomarker curves.

We used bootstrapping via Monte Carlo resampling to quantify the variance of the estimated parameters. We drew 100 resamples of the observed dataset by random sampling (with replacement) from the original collection of subjects, and then recomputed the ADPS for the entire population. Bootstrap replicates of the estimated biomarker sigmoids are shown in Fig. 3 and 90% confidence intervals

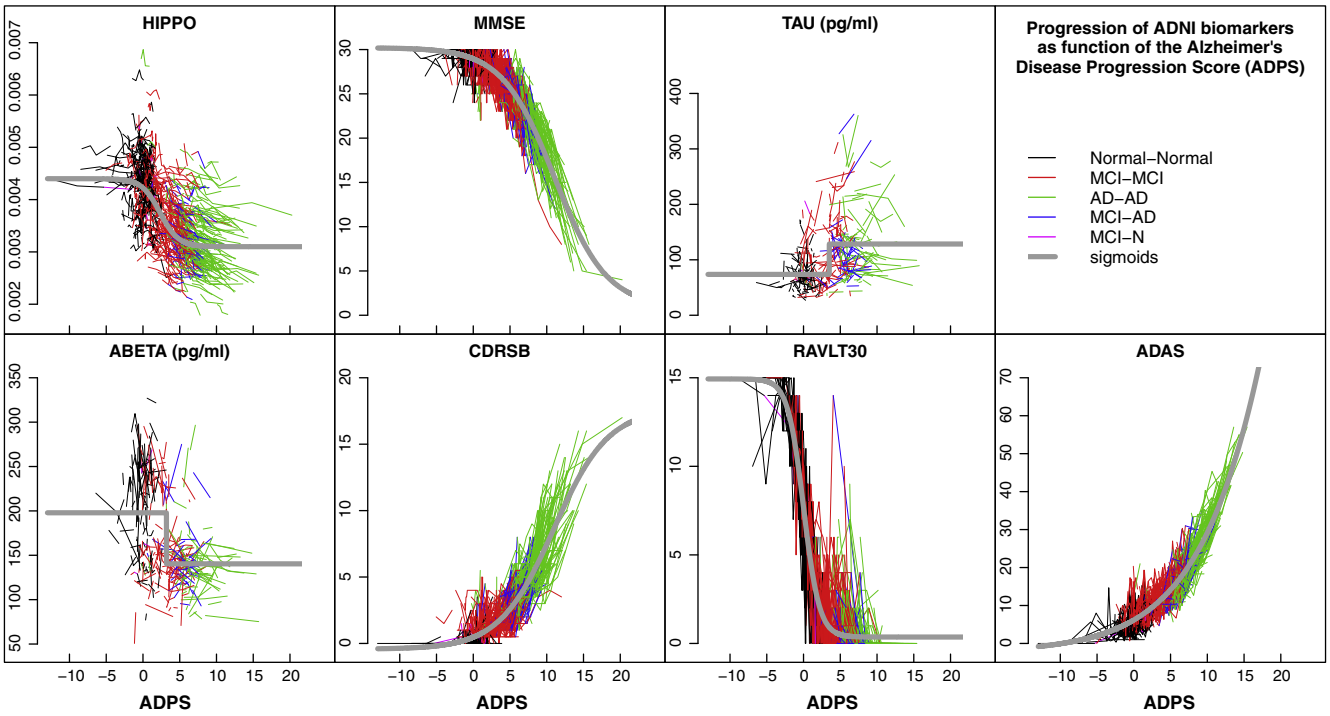


Fig. 2. The values of seven biomarkers, measured at all visits of all ADNI subjects, are plotted on the normalized ADPS. Each connected polyline represents the consecutive visits of a single subject, and each line segment is colored according to the subject's clinical diagnoses between visits (see legend). The gray curves are the sigmoid functions representing the fitted behavior of each biomarker in the normalized space.

for the parameter c_k, i.e. the inflection point of each sigmoid, are presented in Fig 5(b).

The empirical variance of the residuals ϵ_{ijk} in (3) is the component of the variance which is unexplained by the model. It accounts for about 38% of the total variance. Hence the model explains 62% ($\pm$1.37%) of the total variance (i.e., 62% = 100% − 38%.), the standard deviation (sd) of 1.37% being computed using the bootstrap samples. If instead of the ADPS, ADAS or MMSE was used as a disease progression score, fitting sigmoid curves as previously described, the percentage of explained variance would be respectively 49.4% ($\pm$1.4%) and 46% ($\pm$1.4%). The percentage of explained variance is larger with the ADPS than with the ADAS (p-value$<$0.01) or the MMSE (p-value$<$0.01); p-values being obtained using the bootstrap replicates in both cases.

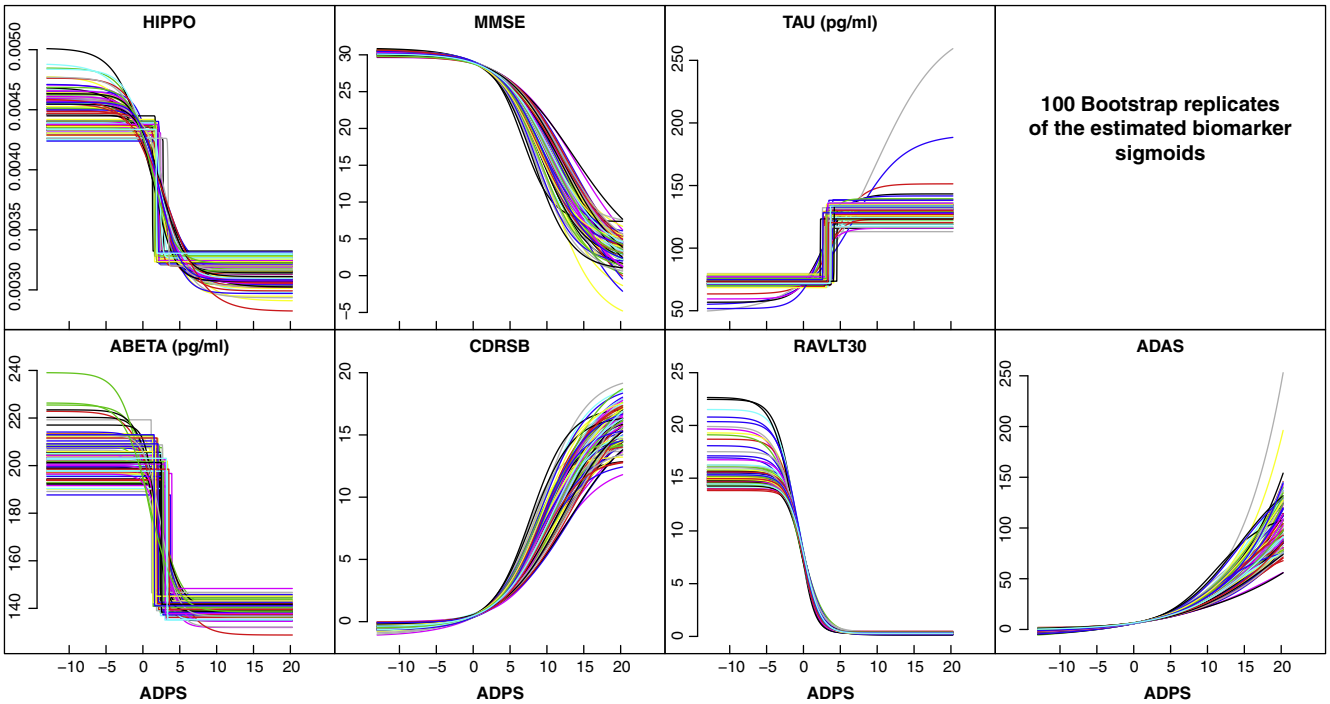


Fig. 3. Bootstrapping yields different biomarker sigmoids with each random substitution. These plots give all the computed sigmoids over the entire bootstrapping exercise. Tight agreement overall is observed.

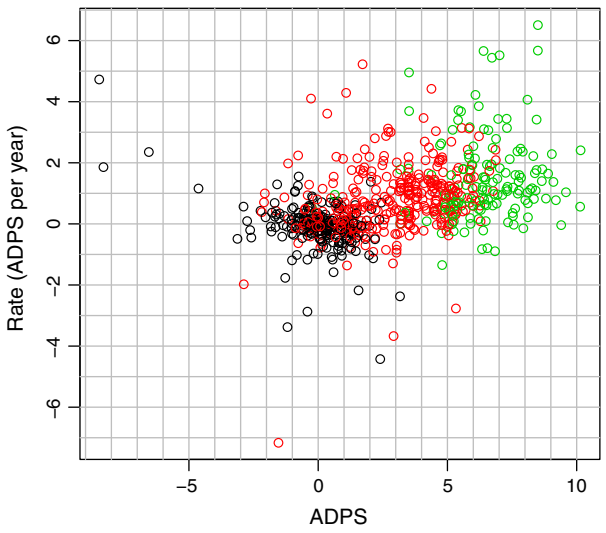


Fig. 4. Rate of the ADPS as function of the ADPS for baseline visits. Black: Normal subjects. Red: MCI subjects. Green: AD subjects.

Table 1
Mean value (standard deviation) of ADPS and rate of change of ADPS for N,MCI and AD subjects in ADNI at baseline.

	ADPS: Mean (sd)	Rate of change of ADPS: Mean (sd)
N	−0.03 (1.48)	−0.08 (0.81)
MCI	2.85 (1.98)	0.76 (1.11)
AD	6.49 (1.61)	1.46 (1.38)

Relation between ADPS and rate of progression

The rate of progression α_i of each subject i is also computed as part of the ADPS parameter fitting algorithm. We plotted the rate of progression of each subject against their ADPS at baseline to see whether a relationship might exist (Fig. 4). A clear trend of increasing rate of ADPS as a function of ADPS is observed. The third column of Table 1 provides the mean rate of change of ADPS in unit of years for each status. AD subjects progress faster on average than MCI subjects. MCI subjects progress faster on average than N subjects. Observed during 3 years, an MCI subject would progress on average at 0.76 ADPS per year. The corresponding ADPS would then increase by $0.76 \times 3 = 2.28$ units. In our model, the ADPS of each subject is a linear function of age, or equivalently the rate of change of ADPS is constant over the time a subject is observed. Retrospectively, it is therefore a reasonable approximation for N and MCI subjects. It might be too simple a model for AD subjects. It is important to recall that these observations are made in light of the optimization criterion of ADPS, which uses the commonality of biomarker trends as a basis for determining rate. Thus, an increasing rate of ADPS truly means that subjects are progressing through degrading biomarkers at a faster rate.

Biomarker dynamics

The sigmoidal functions representing common behavior of biomarker dynamics of the entire ADNI population can be compared by scaling (and inverting if necessary) each of them independently to range from −1 (Normal) to +1 (Abnormal). Plotted as a function of the normalized ADPS (Fig. 5(a)), these scaled sigmoidal functions provide a plot similar to the conceptual plot in Jack et al. (2010) (Fig. 1). Our plot is data driven, of course, representing what the entire ADNI dataset predicts under our model assumptions. Its sigmoidal functions also provide information about the time of initial biomarker change (represented by the heels of the sigmoidal functions), the time of maximum biomarker change (represented by the inflection point of the sigmoidal functions), and the rate of biomarker change over the course of its activation (represented by the slopes of the sigmoidal functions).

In addition to their interpretation as the time of maximum biomarker change, the inflection points also could represent a threshold between normal and abnormal. Therefore, we use them as an indicator

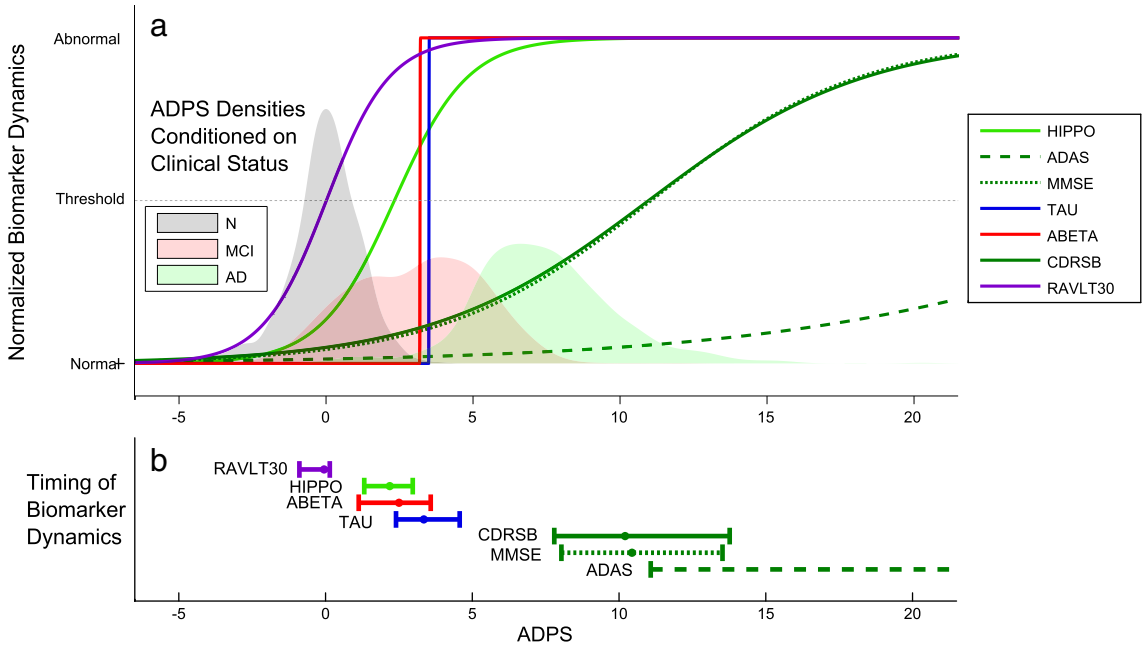


Fig. 5. (a) Estimated biomarker dynamics as a function of the normalized ADPS. Estimation of the normalized ADPS for all ADNI subjects was carried out, and common biomarker dynamics represented by sigmoidal functions were simultaneously fitted as part of the ADPS normalization algorithm. Each sigmoidal function was scaled and flipped in order to fit on a scale going from -1 representing "Normal" to 1 representing "Abnormal". The positions of vertical lines representing progression from Normal to MCI and MCI to AD were fitted as optimal separating thresholds between the clinical diagnoses provided in the ADNI database. (b) 90% confidence intervals for the inflection point of each biomarker.

of biomarker timing in the disease process. We recomputed the inflection point of the normalized biomarker sigmoids for each bootstrap sample and plotted 90% confidence intervals (Fig 5(b)). Furthermore, counting pairwise ordering within the bootstrap samples, we find that *RAVLT30* precedes all other 6 other biomarkers (p-value<0.01) and *HIPPO*, *ABETA* and *TAU* precede *MMSE* and *ADAS* (p-value<0.02).

Relation between ADPS and clinical status

Conditional probability densities of ADPS given the clinical status of each subject were computed using Gaussian kernel density estimation (Fig. 5(a)). Since N subjects tend to have a smaller ADPS than MCI subjects who in turn tend to have a smaller ADPS than AD subjects, this plot confirms that ADPS provides a scale that correlates strongly with clinical classification of disease. The mean and standard deviation of the baseline ADPS for N, MCI and AD subjects in ADNI is provided in Table 1, column 2. The means are well separated from each other. There is overlap in the baseline ADPS value between N and MCI and also between MCI and AD, but essentially not between N and AD. It is worth restating the clinical diagnosis is not used in computing the ADPS except to determine its units.

Discussion

We combine multiple biomarkers to provide a neurodegenerative disease progression. In contrast, in the case of AD, Brooks et al. (1993); Stern et al. (1994); Ashford et al. (1995); Mitnitski et al. (1999) and others use *MMSE* or *ADAS* as measure of disease progression. In Yang et al. (2011a), the authors synchronize subjects onto a time-line constructed using ADAS scores. The core assumption is that the rate of change of ADAS is linear with respect to the ADAS score, resulting in an exponential model of disease progression. In Walhovd et al. (2010); Hinrichs et al. (2011), multiple biomarkers are combined to diagnose AD. In Fonteijn et al. (2011) the progression of AD is divided into discrete events based on the atrophy of different structures in the brain providing a probabilistic framework for estimating the global progression of AD as well as for estimating the position of a single subject's measurements. Longitudinal measurements are not used. In Ververidis et al. (2010), a Bayesian classifier selects the set of biomarkers which are most informative for classifying the current state of the disease. Time-series models are used to predict the future state of the disease. Yang et al. (2011b) use independent component analysis and support vector machines to classify subjects into N versus MCI or AD. Our statistical model is related to so-called single index models (see Hardle et al. (1993); Carroll et al. (1997) and the references therein). However, our models differ from these, as we assume parsimonious parametric forms for the index function and allow for multivariate outcomes.

Our modeling technique applied to the ADNI has provided confirmation of existing results: Jack et al. (2011) binarized each biomarker into either normal or abnormal using a threshold or cut point. Cut points were determined for each biomarker at autopsy and with an independent cohort. When using these cut point to determine the ADPS at which a biomarker changes from normal to abnormal, we find that *ABETA* precedes both *HIPPO* and *TAU* which is consistent with the results in Jack et al. (2011). We have also obtained surprising results. The fact that the inflection of *RAVLT30* precedes that of all other biomarkers, and in particular that of *ABETA* is surprising, compared to Fig. 1, but consistent with some predictions. Jicha and Carr (2010) refer to the study in Bennett et al. (2006) stating, "Retrospective analysis of their neuropsychological test performance demonstrated significant differences in only delayed recall tasks between subjects with pathological AD autopsy findings and those with normal autopsy findings, suggesting that memory decline may be present, albeit subtly, in persons with (preclinical) AD before sufficient cognitive decline to warrant the diagnosis of either MCI or dementia."Also, Dubois et al. (2007) advocate that the presence of an early and significant episodic memory impairment should constitute one of the core diagnostic criteria for AD.

Conclusion

We report a multiple biomarker, data-driven approach to assess time-dependent changes of biomarkers in neurodegenerative disease and to localize subjects on a scale of disease progression, the DPS, over the entire range of progression. The statistical model is shown to be identifiable and bootstrap replicates show that the parameters are estimated tightly in case of the ADNI dataset. The DPS integrates information from multiple biomarkers into a single composite biomarker. Using this approach the conceptual plot of Jack et al. (2010) can be recreated using the ADNI data. The sequence of biomarkers obtained by comparing the inflection point of each biomarker is similar to that in Jack et al. (2010) with an exception: the *RAVLT30* becomes dynamic before all other biomarkers. The DPS provides a continuous measure of progression over the whole course of disease, and it could therefore be used to stage individuals for prognosis and to evaluate the effects of novel drugs at all stages of the disease. The method is generic and is applicable to all neurodegenerative diseases pending availability of the data.

Acknowledgments

Personnel costs for this research were partially supported by a grant from Pfizer Inc. Other support came from grants numbered P41EB015909 and R01EB012547 from the National Institute of Biomedical Imaging and Bioengineering as well as from an Ossoff Scholar Award. Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health grant U01 AG024904). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: Abbott; Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Amorfix Life Sciences Ltd.; AstraZeneca; Bayer HealthCare; BioClinica, Inc.; Biogen Idec Inc.; Bristol-Myers Squibb Company; Eisai Inc.; Elan Pharmaceuticals Inc.; Eli Lilly and Company; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; GE Healthcare; Innogenetics, N.V.; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Medpace, Inc.; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Servier; Synarc Inc.; and Takeda Pharmaceutical Company. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Disease Cooperative Study at the University of California, San Diego. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of California, Los Angeles. This research was also supported by NIH grants P30 AG010129 and K01 AG030514. The first author would also like to thank Patrick Slama for his insightful remarks.

Appendix A. Proof of Identifiability

Theorem 1. *The model* $\{P_\rho; \rho \in \varrho\}$ *is identifiable as long as the following 2 conditions are verified:*

1. *For each biomarker, there is at least one subject* i *with* $\alpha_i \neq 0$ *and with at least 4 distinct time-points at which this biomarker is available.*
2. *For each subject, there is at least one biomarker which is available at 2 time points.*

The proof uses the invertibility of a multivalued function closely related to f. This property is deferred to lemma 1.

Proof of Theorem 1. Let us assume that the model is not identifiable. Then there exists 2 sets of parameters in ϱ, $\rho=(a, b, c, d, \alpha, \beta, \sigma)$ and $\rho'=(a', b', c', d', \alpha', \beta', \sigma')$ which differed by at least 1 component, while verifying $P_\rho = P_{\rho'}$. Equivalently,

$$f\left(\alpha_i t_{ij}+\beta_i; a_k, b_k, c_k, d_k\right) = f\left(\alpha'_i t_{ij}+\beta'_i; a'_k, b'_k, c'_k, d'_k\right) \quad (A.1)$$

for all $(i,j,k)\in\mathcal{I}$ and $\sigma_k=\sigma'_k$ for all k

We proceed in steps until we verify that necessarily $\rho=\rho'$. Since $\sigma_k=\sigma'_k$, for all $k=1\ldots K$, we concentrate on the other parameters. For each k, let i be a subject such that $\alpha_i>0$ and for which biomarker k is observed at four different time points t_{i1}, t_{i2}, t_{i3}, t_{i4} . Notate $u_{ik}=b_k\alpha_i$, $v_{ik}=b_k(\beta_i-c_k)$,$u'_{ik}=b'_k\alpha'_i$ and $v'_{ik}=b'_k(\beta'_i-c'_k)$. Rearanging the arguments of f and using (A.1),

$$f\left(t_{ij}; a_k, u_{ik}, -{u_{ik}}^{-1}v_{ik}, d_k\right) = f\left(t_{ij}; a'_k, u'_{ik}, -{u'_{ik}}^{-1}v'_{ik}, d'_k\right)$$

for $j=1\ldots 4$. Note that since $a_i\neq 0$ and $b_k\neq 0$, $u_{ik}\neq 0$ and $u'_{ik}\neq 0$. Now, using Lemma 1, $a_k=a'_k$, $d_k=d'_k$, $u_{ik}=u'_{ik}$, ${u_{ik}}^{-1}v_{ik}={u'_{ik}}^{-1}v'_{ik}$. Summing up over i and dividing by I in $b_k\alpha_i=b'_k\alpha'_i$, we obtain $b_k\alpha_0=b'_k\alpha_0$, and since $\alpha_0\neq 0$, $b_k=b'_k$. Since $b_k\neq 0$, it follows that $\alpha_i=\alpha'_i$ and $u_{ik}=u'_{ik}$. Replacing in $v_{ik}=v'_{ik}$ and summing up over i and dividing by I, we obtain that $c_k=c'_k$. We have then obtained that for all biomarkers, $a_k=a'_k$, $b_k=b'_k$, $c_k=c'_k$, $d_k=d'_k$ and $\sigma_k=\sigma'_k$. Now, for each subject i, there is at least one biomarker k for which two time-points t_{i1} and t_{i2} are available. Replacing in (A.1),

$$f\left(\alpha_i t_{ij}+\beta_i; a_k, b_k, c_k, d_k\right) = f\left(\alpha'_i t_{ij}+\beta'_i; a_k, b_k, c_k, d_k\right) \quad (7)$$

for $j=1,\ldots,2$. Since $a_k\neq 0$ and $b_k\neq 0$, $t\to f(t; a_k, b_k, c_k, d_k)$ is invertible which, together with (7), implies that $\alpha_i=\alpha'_i$ and $\beta_i=\beta'_i$ concluding the proof.

Lemma 1. *The vector values function $R^4\to R^4$ for fixed $x_1<x_2<x_3<x_4$: defined by*

$$(a,b,c,d)\to(f(x_1;a,b,c,d), f(x_2;a,b,c,d), f(x_3;a,b,c,d), f(x_4;a,b,c,d))$$

with $a\neq 0$, $b>0$ is invertible.

Proof of Lemma 1. We verify that the Jacobian determinant of this function is nonzero, which is enough to prove invertibility using the inverse function theorem of multivariate calculus. Let $c'=e^{bc}$

$$f(x;a,b,c',d)=\frac{a}{1+c'e^{-bx}}+d$$

It is equivalent to show the Jacobian determinant of

$$(a,b,c',d)\to(f(x_1;a,b,c',d), f(x_2;a,b,c',d), f(x_3;a,b,c',d), f(x_4;a,b,c',d))$$

is non zero.

The ith row of the Jacobian matrix is:

$$\left(1+c'e^{-bx_i}\right)^{-2}\left[1+e^{-bx_i}, ac'x_ie^{-bx_i}, -ae^{-bx_i}, 1+2c'e^{-bx_i}+c'^2e^{-2bx_i}\right]$$

Column linear transformation will not change the singularity of the Jacobian matrix. After some linear transformations, the ith row is:

$$\left(1+c'e^{-bx_i}\right)^{-2}\left[1,\quad x_ie^{-bx_i},\quad e^{-bx_i},\quad e^{-2bx_i}\right]$$

Suppose the Jacobian matrix is singular, i.e. there exists (not all zero) coefficients k, l, m, n such that

$$k+lx_ie^{-bx_i}+me^{-bx_i}+ne^{-2bx_i}=0; i=1,\ldots,4$$

then the function

$$g(x)=k+lxe^{-bx}+me^{-bx}+ne^{-2bx}$$

must have four real roots. Differentiating twice,

$$2b^2ne^{-bx}-lb$$

would need to have 2 real roots. Since it is not the case, the Jacobian matrix is invertible, which concludes the proof.

References

Ashford, J., Shan, M., Butler, S., Rajasekar, A., Schmitt, F., 1995. Temporal quantification of Alzheimer's disease severity: 'Time Index' model. Dementia 6 (5), 269–280.

Bennett, D.A., Schneider, J.A., Arvanitakis, Z., Kelly, J.F., Aggarwal, N.T., Shah, R.C., Wilson, R.S., 2006. Neuropathology of older persons without cognitive impairment from two community-based studies. Neurology 66 (12), 1837–1844.

Berg, L., Miller, J.P., Storandt, M., Duchek, J., Morris, J.C., Rubin, E.H., Burke, W.J., Coben, L.A., 1988. Mild senile dementia of the alzheimer type: 2. Longitudinal assessment. Ann. Neurol. 23 (5), 477–484.

Brooks, D.J., Pavese, N., 2011. Imaging biomarkers in Parkinson's disease. Progress in Neurobiology. http://dx.doi.org/10.1016/j.pneurobio.2011.08.009. Aug.

Brooks III, J., Kraemer, H., Tanke, E., Yesavage, J., 1993. The methodology of studying decline in Alzheimer's disease. J. Am. Geriatr. Soc. 41 (6), 623–628.

Caroli, A., Frisoni, G., the Alzheimer's Disease Neuroimaging Initiative, 2010. The dynamics of Alzheimer's disease biomarkers in the Alzheimer's disease neuroimaging initiative cohort. Neurobiol. Aging 31 (8), 1263–1274.

Carroll, R., Fan, J., Gijbels, I., Wand, M., 1997. Generalized partially linear single-index models. J. Am. Stat. Assoc. 92 (438), 477–489.

Duara, R., Loewenstein, D.A., Greig, M.T., Potter, E., Barker, W., Raj, A., Schinka, J., Borenstein, A., Schoenberg, M., Wu, Y., Banko, J., Potter, H., 2011. Pre-MCI and MCI: neuropsychological, clinical, and imaging features and progression rates. Am. J. Geriatr. Psychiatry 19 (11), 951–960.

Dubois, B., Feldman, H.H., Jacova, C., DeKosky, S.T., Barberger-Gateau, P., Cummings, J., Delacourte, A., Galasko, D., Gauthier, S., Jicha, G., Meguro, K., O'Brien, J., Pasquier, F., Robert, P., Rossor, M., Salloway, S., Stern, Y., Visser, P.J., Scheltens, P., 2007. Research criteria for the diagnosis of Alzheimer's disease: revising the NINCDS-ADRDA criteria. Lancet Neurol. 6 (8), 734–746.

Folstein, M.F., Folstein, S.E., McHugh, P.R., 1975. "Mini-mental state": a practical method for grading the cognitive state of patients for the clinician. J. Psychiatr. Res. 12 (3), 189–198.

Fonteijn, H.M.J., Clarkson, M.J., Modat, M., Barnes, J., Lehmann, M., Ourselin, S., Fox, N.C., Alexander, D.C., 2011. An event-based disease progression model and its application to familial Alzheimer's disease. Proc. Information Processing in Medical Imaging (IPMI). Vol. 6801 of Lecture Notes in Computer Science. Springer, pp. 748–759.

Hampel, H., Burger, K., Teipel, S.J., Bokde, A.L., Zetterberg, H., Blennow, K., 2008. Core candidate neurochemical and imaging biomarkers of Alzheimer's disease. Alzheimers Dement. 4 (1), 38–48.

Hardle, W., Hall, P., Ichimura, H., 1993. Optimal smoothing in single-index models. Ann. Stat. 21 (1), 157–178.

Hayden, M., 1981. Huntington's Chorea. Springer-Verlag.

Hinrichs, C., Singh, V., Xu, G., Johnson, S.C., 2011. Predictive markers for AD in a multi-modality framework: an analysis of MCI progression in the ADNI population. NeuroImage 55 (2), 574–589.

Hughes, C., Berg, L., Danziger, W., Coben, L., Martin, R., 1982. A new clinical scale for the staging of dementia. Br. J. Psychiatry 140, 566–572.

Jack Jr., C.R., Knopman, D.S., Jagust, W.J., Shaw, L.M., Aisen, P.S., Weiner, M.W., Petersen, R.C., Trojanowski, J.Q., 2010. Hypothetical model of dynamic biomarkers of the Alzheimer's pathological cascade. Lancet Neurol. 9 (1), 119–128.

Jack, C.R.J., Vemuri, P., Wiste, H.J., Weigand, S.D., Aisen, P.S., Trojanowski, J.Q., Shaw, L.M., Bernstein, M.A., Petersen, R.C., Weiner, M.W., Knopman, D.S., for the Alzheimer's Disease Neuroimaging Initiative, 2011. Evidence for ordering of alzheimer disease biomarkers. Arch. Neurol. 68 (12), 1526–1535.

Jicha, G.A., Carr, S.A., 2010. Conceptual evolution in Alzheimer's disease: implications for understanding the clinical phenotype of progressive neurodegenerative disease. J. Alzheimers Dis. 19 (1), 253–272.

Labbe, A., 2012. ALS biomarkers study seeks 250 participants. MDA ALS News Magazine. January.

Levenberg, K., 1944. A method for the solution of certain non-linear problems in least squares. Q. Appl. Math. 2, 164–168.

Martin, L.J., 2002. Ch. Neurodegeneratives disorders of the human brain and spinal cord. Encyclopedia of the Human Brain, vol. 3. Elsevier Science Academic press, pp. 441–463.

Mitnitski, A., Graham, J., Rockwood, K., 1999. Modeling decline in Alzheimer's disease. Int. Psychogeriatr. 11 (02), 211–213.

Mosconi, L., Brys, M., Glodzik-Sobanska, L., Santi, S.D., Rusinek, H., de Leon, M.J., 2007. Early detection of Alzheimer's disease using neuroimaging. Exp. Gerontol. 42 (1–2), 129–138.
Mungas, D., Reed, B.R., 2000. Application of item response theory for development of a global functioning measure of dementia with linear measurement properties. Stat. Med. 19 (11–12), 1631–1644.
Qiu, A., Younes, L., Miller, M.I., Csernansky, J.G., 2008. Parallel transport in diffeomorphisms distinguishes time-dependent hippocampal surface atrophy in healthy aging and Alzheimer's disease. NeuroImage 40, 68–76.
Richards, F., 1959. A flexible growth function for emperical use. J. Exp. Bot. 10, 290–300.
Rizk-Jackson, A., Stoffers, D., Sheldon, S., Kuperman, J.M., Dale, A.M., Goldstein, J., Corey-Bloom, J., Poldrack, R.A., Aron, A.R., 2011. Evaluating imaging biomarkers for neurodegeneration in pre-symptomatic Huntington's disease using machine learning techniques. NeuroImage 56 (2), 788–796.
Sabuncu, M.R., Desikan, R.S., Sepulcre, J., Yeo, B.T.T., Liu, H., Schmansky, N.J., Reuter, M., Weiner, M.W., Buckner, R.L., Sperling, R.A., Fischl, B., ADNI, 2011. The dynamics of cortical and hippocampal atrophy in Alzheimer disease. Arch. Neurol. 68 (8), 1040–1048.
Shaw, P.J., 2005. Molecular and cellular pathways of neurodegeneration in motor neurone disease. J. Neurol. Neurosurg. Psychiatry 76 (8), 1046–1057.
Sperling, R.A., Aisen, P.S., Beckett, L.A., Bennett, D.A., Craft, S., Fagan, A.M., Iwatsubo, T., Jack, C.R., Kaye, J., Montine, T.J., Park, D.C., Reiman, E.M., Rowe, C.C., Siemers, E., Stern, Y., Yaffe, K., Carrillo, M.C., Thies, B., Morrison-Bogorad, M., Wagster, M.V., Phelps, C.H., 2011. Toward defining the preclinical stages of Alzheimer's disease: recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. Alzheimers Dement. 7 (3), 280–292.
Stern, R., Mohs, R., Davidson, M., Schmeidler, J., Silverman, J., Kramer-Ginsberg, E., Searcey, T., Bierer, L., Davis, K., 1994. A longitudinal study of Alzheimer's disease: measurement, rate, and predictors of cognitive deterioration. Am. J. Psychiatry 151 (3), 390–396.
Turner, M.R., Kiernan, M.C., Leigh, P.N., Talbot, K., 2009. Biomarkers in amyotrophic lateral sclerosis. Lancet Neurol. 8 (1), 94–109 (January).
Turner, M.R., Grosskreutz, J., Kassubek, J., Abrahams, S., Agosta, F., Benatar, M., Filippi, M., Goldstein, L.H., van den Heuvel, M., Kalra, S., Lul, D., Mohammadi, B., 2011. Towards a neuroimaging biomarker for amyotrophic lateral sclerosis. Lancet Neurol. 10 (5), 400–403 URL http://www.sciencedirect.com/science/article/pii/S1474442211700497.
Ververidis, D., Van Gils, M., Koikkalainen, J., Lotjonen, J., 2010. Feature selection and time regression software: application on predicting Alzheimer's disease progress. Proc. European Signal Processing Conference (EUSIPCO).
Walhovd, K., Fjell, A., Brewer, J., McEvoy, L., Fennema-Notestine, C., Hagler Jr., D.J., Jennings, R., Karow, D., Dale, A., the Alzheimer's Disease Neuroimaging Initiative, 2010. Combining MR imaging, positron-emission tomography, and CSF biomarkers in the diagnosis and prognosis of Alzheimer disease. AJNR Am. J. Neuroradiol. 31 (2), 347–354.
Yang, E., Farnum, M., Lobanov, V., Schultz, T., Raghavan, N., Samtani, M.N., Novak, G., Narayan, V., DiBernardo, A., 2011a. Quantifying the pathophysiological timeline of Alzheimer's disease. J. Alzheimers Dis. 26 (4), 745–753.
Yang, W., Lui, R., Gao, J., Chan, T., Yau, S., Sperling, R., Huang, X., 2011b. Independent component analysis-based classification of Alzheimer's disease mri data. J. Alzheimers Dis. 24 (4), 775–783.