Self-Normalized Linear Tests

Sachin Gangaputra
Dept. of Electrical and Computer Engineering
The Johns Hopkins University
Baltimore, MD 21218
sachin@jhu.edu

Donald Geman
Dept. of Applied Mathematics and Statistics
The Johns Hopkins University
Baltimore, MD 21218
geman@jhu.edu

Abstract

Making decisions based on a linear combination L of features is of course very common in pattern recognition. For distinguishing between two hypotheses or classes, the test is of the form $sign(L - \tau)$ for some threshold τ. Due mainly to fixing τ, such tests are sensitive to changes in illumination and other variations in imaging conditions. We propose a special case, a "self-normalized linear test" (SNLT), hard-wired to be of the form $sign(L_1 - L_2)$ with unit weights. The basic idea is to "normalize" L_1, which involves the usual discriminating features, by L_2, which is composed of non-discriminating features. For a rich variety of features (e.g., based directly on intensity differences), SNLTs are largely invariant to illumination and robust to unexpected background variations. Experiments in face detection are promising: they confirm the expected invariances and out-perform some previous results in a hierarchical framework.

1. Introduction

Numerous methods for pattern detection and classification, including linear discriminant analysis [5], perceptrons [6] and support vector machines [14], involve decisions based on linear combinations of the components of a high-dimensional feature vector $\mathbf{X} = (X_1, ..., X_d) \in \Re^d$. For two classes, denoted $\{-1, 1\}$, the classifier or "test" is then of the form $sign(L(\mathbf{X}) - \tau)$, where L is linear. Generally, the coefficients or weights in L, as well as the threshold τ, are inferred from training data $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where y_i denotes the class of example $\mathbf{x}_i$. With standard methods of inductive learning, the features which dominate the decision are those whose probability distribution differs significantly under the two classes.

In some cases the ultimate prediction $f(\mathbf{X}) \in \{-1, 1\}$ may involve a cascade or hierarchy of linear tests, each dedicated to a sub-problem, for instance a subset of possible object instantiations. In any case, the ultimate goal is to minimize the generalization error $P(f(\mathbf{X}) \neq Y)$, where Y is the true class.

The principal limitation may not be the bias introduced by the simplicity of the decision boundaries. Indeed, this may be an advantage in applications in which n is small relative to d ("*small n, large d dilemma*"). For example, in the analysis of gene microarray data, simple methods, such as naive Bayes, perform quite well as compared to more complex ones [13]. Moreover, more complex boundaries can be obtained within the same framework by mapping into a larger dimensional space via kernels [2].

In our view, a more severe deficiency is insufficient invariance, especially with respect to variations in the imaging conditions, such as changes in illumination and the appearance of highly-structured clutter. In our own experience, the performance of object detectors based on linear tests often degrades under poor contrast (resulting in missed detections) or in high frequency clutter (resulting in false detections). One reason for this is that τ is fixed. *Our objective is to render such linear tests less sensitive to distortions in the observables*. It is doubtful that this can be entirely accomplished by pre-processing based on estimating imaging conditions. Or that invariance can be entirely *learned* from training examples. Some hard-wiring seems preferable. Here, we compare $L(\mathbf{X})$ to a sum of *non-discriminating* features rather than to a fixed threshold, rendering the test at least partially invariant to intensity perturbations which degrade all the features in a similar fashion.

Section 2 provides a rationale and overview of the method. The design strategy is described in more detail in §3, followed, in §4, by an explanation of why SNLTs are invariant to certain image intensity transformations when certain feature classes are used. *Learning* is addressed in §5 - how the design strategy is implemented in practice using a training set. Some experiments on face detection are presented in §6 in order to demonstrate the gains relative to fixed thresholds, as well as to illustrate SNLT stability. Finally, in §7, we critique our approach.

2. Rationale and Overview

By a linear test we mean one of the form

$$f(\mathbf{X}) = sign\left(\sum \alpha_j X_j - \tau\right) \quad (1)$$

Under certain assumptions, such tests are theoretically optimal. A prominent example is when the feature components are binary and conditionally independent, in which case the likelihood ratio test takes the form (1) for certain weights (naive Bayes classifier); by construction, the weight of a feature reflects its level of discrimination. According to the Neyman-Pearson Lemma, this test minimizes false positive error for any prescribed false negative rate. However, such results break down in practice, due not only to the lack of independence but to the fact that τ is fixed whereas the statistics of the features can be very sensitive to common perturbations of the raw data.

Other design strategies for choosing weights and thresholds are provided by perceptron learning and support vector machines, extremely well-known methods which will not be reviewed here; see [3]. Again, despite sound theoretical considerations about generalization error which motivate the design, problems arise in uncontrolled imaging conditions. Only with gigantic training sets can the data variations inevitably encountered be adequately represented.

We propose a special case of (1) for which $\tau = 0$ and all the weights $\alpha_j = \pm 1$. A *self-normalized linear test* (SNLT) is then of the form

$$f(\mathbf{X}) = sign\left(\sum_{j \in J_D} X_j - \sum_{j \in J_S} X_j\right). \quad (2)$$

The basic idea is that uninformative features (with similar distributions under $Y = \pm 1$) should not be discarded as useless but rather kept for normalization – in effect to replace τ. Consequently, the two sums in (2) play distinctly different roles. The set J_D captures discriminating features, typically assuming disparate values under $Y = -1$ and $Y = 1$; the set J_S is composed of *non-discriminating features*, typically assuming similar values under $Y = -1$ and $Y = 1$. The two sets J_D, J_S are chosen to achieve a desired tradeoff between false negative and false positive errors. In principle, if data distortions leaving Y unchanged affect the components of the feature vector $\mathbf{X}$ in a similar way, then the resulting SNLT can be largely unaffected. For instance, the SNLT is invariant to linear contrast changes if the individual features scale multiplicatively, and is largely insensitive to quantization.

Features: We consider large families of features, all of the same basic type and all aimed at detecting local intensity changes in the image I. Other types of features, or heterogeneous collections, could also be envisioned. Our experiments are based on binary features - logical functions of intensity differences $X = |I(u) - I(v)|$, for instance requiring one such difference to be larger than a group of others; see §4. Such features have the property that $X(aI + b) = c(a, b)X(I)$ (except for quantization effects), rendering any resulting SNLT independent of a and b.

Learning: One goal of learning is to divide the features into two pools – those which change considerably and those which are very stable. Since we also want to avoid redundancy in J_D, this is not entirely straightforward. Low false negative (type I) error (corresponding, for instance, to few missed detections) is maintained if the features in L_1 are stochastically larger under class $Y = 1$. False positive (type II) error can be controlled by having more terms in L_1 than in L_2, rendering $f(\mathbf{X}) = 1$ unlikely under $Y = -1$ provided the features in *both* sums are distributed similarly. This type of highly structured test does not seem to emerge naturally from standard learning devices.

Validation: Experiments in face detection are promising, confirming the expected invariances. We have used the hierarchical framework described in [4]; roughly speaking, it involves a coarse-to-fine tree-structured hierarchy of linear tests, each designed for a different level of generality and discrimination. Adopting this detection strategy for a particular choice of features, with everything else fixed, we compare the performance of fixed (learned) thresholds to replacing them by (learned) sums of non-distinguished features. We suspect the observed improvements would persist for other recognition strategies based on aggregating evidence from linear tests.

3. Design

In (2) we choose $J_S, J_D \subset J \doteq \{1, 2, ..., d\}$ based on statistical criteria. That is, the features appearing in L_1 and in L_2 will be characterized by comparing their probability distributions under the two hypotheses $Y \in \{-1, 1\}$.

Let $p_j(x)$ and $q_j(x)$ denote those distributions:

$$p_j(x) = P(X_j = x | Y = 1)$$
$$q_j(x) = P(X_j = x | Y = -1).$$

For the moment we assume these distributions are known; in practice they are estimated from data, indicated by writing $\hat{p}_j, \hat{q}_j$. Without the loss of generality we can assume that $0 \leq X_j \leq 1$ and that

$$\mu_j \doteq E(X_j | Y = 1) \geq E(X_j | Y = -1) \doteq \nu_j, \ \ j \in J \quad (3)$$

by replacing X_j by $1 - X_j$ if necessary.

Let δ be some measure of the disparity between two distributions, perhaps just the difference between the means $|\mu_j - \nu_j|$ or perhaps a more global metric like Kullback-Leibler or L_1 norm. For binary features $X_j \in \{0, 1\}$, let

$$p_j = p_j(1) = P(X_j = 1 | Y = 1),$$

$$q_j = q_j(1) = P(X_j = 1|Y = -1)$$

and take $\delta(p_j, q_j) = |p_j - q_j|$.

One simple possibility for choosing J_D and J_S is to order the (non-negative) values $\mu_j - \nu_j$ and put all those above a certain threshold into J_D and the rest into J_S. However, we can do better, both in terms of error and computation, by judicious sampling: We are going to select

$$J_D \subset \{j : \delta(p_j, q_i) \gg 0\} \tag{4}$$
$$J_S \subset \{j : \delta(p_j, q_j) \approx 0\}. \tag{5}$$

We might wish to choose parameters which quantify $\gg 0$ and ≈ 0 in order to achieve a desired tradeoff between false negatives and false positives. In fact, many designs could be envisioned depending on the nature of the two hypotheses and the feature set.

In our experiments, we limit the size of J_D and use a primitive form of boosting (see §5) to minimize redundancy (i.e., limit dependency) among the features in J_D, thereby increasing efficiency. The selection of J_S is then driven by minimizing the false negative error rate while at the same time maximizing $|J_S|$. In cases in which $Y = -1$ represents a nonspecific, "background" alternative, and the features represent image transitions (see §4), then arranging for $|J_D| \lll |J_S|$ is straightforward (there is generally more "inactivity" than "activity"), and the false positive rate is thereby controlled (see §5).

4. Feature Classes and Invariance

For many features classes, anomalies of the measurement process may, on aggregate, affect the features X_j in such a way that the two sums appearing in (2) are similarly altered, for example their expectations may by reduced by the same factor.

To be concrete, we focus on photometric variations in the imaging conditions. Basically, then, we want features with the property that if I is transformed to $\Psi(I)$, then

$$X_j(\Psi(I)) = c(\Psi)X_j(I), \ \forall j \in J \tag{6}$$

for some positive constant c. In this case (2) remains unchanged: $f(\mathbf{X}(\Psi(I))) = f(\mathbf{X}(I))$. Here are some examples.

1. **Linear functions:**

   $$X = \sum_i \lambda_i I(u_i), \ \ \sum_i \lambda_i = 0.$$

   Then clearly (6) holds for linear transformations $\Psi : I \to aI + b \ (a > 0)$ with $c(\Psi) = a$.

2. **Absolute differences:** $X = |I(u) - I(v)|$ for neighboring pixels u and v, say $\| \ u - v \ \| = 1$. Again we obtain invariance to linear intensity transforms. Here, and in the preceding example, we might take $\delta(p_j, q_j) = |\mu_j - \nu_j|$ (or replace means by medians).

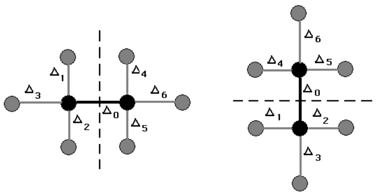

Figure 1. Two examples of comparisons of differences, designed to detect vertical and horizontal edges, respectively. In each case, we demand that the central absolute difference be larger than the other six.

3. **Comparisons of differences:** The features used in [1] and elsewhere are based on comparing one difference to surrounding differences. This is illustrated in Figure 1. Each feature is of the form

   $$X = \begin{cases} 1 & \text{if } \Delta_0 > \max_{1 \le i \le 6} \Delta_i \\ 0 & \text{otherwise} \end{cases}$$

   Hence there is one feature for each pair of neighboring vertical, horizontal and diagonal pixels and two per pair if we retain the polarity of the jump. Again, we achieve invariance to linear transforms with $c(\Psi) \equiv 1$.

4. **Direct intensity comparisons:** Given a small neighborhood N of pixels, and one distinguished pixel $u \in N$, we define

   $$X = \begin{cases} 1 & \text{if } I(u) > \max_{v \in N} I(v) \\ 0 & \text{otherwise} \end{cases}$$

   Varying choices of the location of u relative to the other pixels picks up "edges" of varying orientations. The resulting SNLTs are, modulo resolution, invariant to *all monotone increasing* Ψ.

   Note: Features based on *strict* inequalities between intensities are only invariant in the continuum. Quantization may convert inequalities to equalities. For instance, a linear transformation of greyscales may result in a loss of intensity resolution, in which case we may have $X(\Psi(I)) = 0 < 1 = X(I)$ for some X. This is why, for example, the experiments based on comparisons of differences are not strictly invariant.

5. Learning

Given a family of features $\mathbf{X} = (X_1, ..., X_d)$ and a training set $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, we begin by collecting the empirical statistics for the individual means (or probabilities in the binary case). These are denoted by $\hat{\mu}_j$ (under $Y = 1$) and $\hat{\nu}_j$ (under $Y = -1$) in general and by $\hat{p}_j, \hat{q}_j$ in the binary case. We are of course primarily interested in the *differences*.

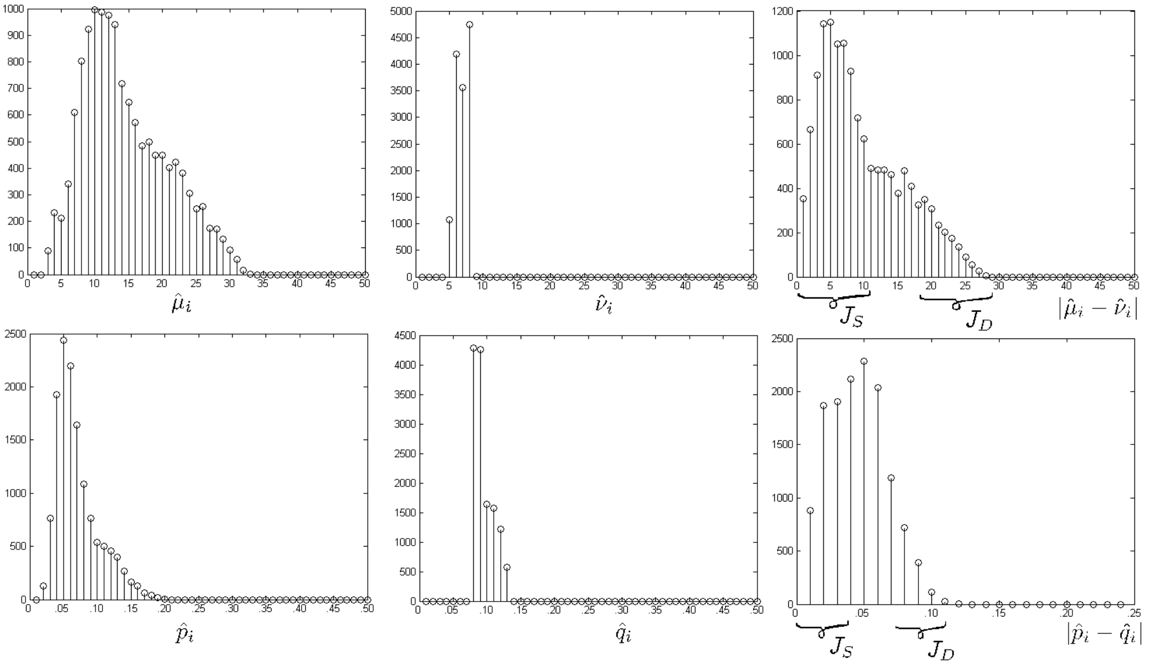

Figure 2. Feature statistics estimated from a face/non-face training set. Top row: The distribution of means over "objects"($\hat{\mu}_j$), "background" ($\hat{\nu}_j$) and "object - background" differences ($|\hat{\mu}_j - \hat{\nu}_j|$) for a scalar family. The index sets J_D and J_S appearing in (2) are drawn from the indicated regions. Bottom row: The same histograms for the binary feature family based on comparisons of differences.

In Figure 2, we show some of the resulting histograms based on examples of faces ($Y = 1$) and non-faces ($Y = -1$) from a database described in the following section. In the top row, from left to right, we see the histograms of $\{\hat{\mu}_j, j \in J\}$, $\{\hat{\nu}_j, j \in J\}$ and $\{|\hat{\mu}_j - \hat{\nu}_j|, j \in J\}$ for the feature family in Example 2 in §4. These histograms are based on the raw values - prior to standardizing to make $\hat{\mu}_j \geq \hat{\nu}_j$ for each feature. Also shown are the ranges of differences from which J_S and J_D, the estimated versions of J_S and J_D, will be selected. The bottom row is the same for the binary family in Example 3 - comparisons of absolute intensity differences.

As indicated in §3, varying objectives could motivate the selections of J_D and J_S. For example, one could calculate the apparent false negative and false positive rates (i.e. on $\mathcal{L}$) for equality in (5) and (4) based on thresholds for $\delta(\hat{p}_j, \hat{q}_j)$, using the empirical statistics, and strike some desired balance. We have taken a somewhat different approach, motivated by i) using binary features; ii) the particular classification scheme we will use to illustrate the ideas, which combines a large number of linear tests in a hierarchical framework (see §6); and iii) trying to anticipate how one might estimate and control the real false positive rate $P(f(\mathbf{X}) = 1 | Y = 1)$.

We want J_D and J_S to have the following properties:

- *Negligible false negative error:*

$$P\left(\sum_{j \in J_D} X_j < \sum_{j \in J_S} X_j \mid Y = 1\right) \approx 0 \qquad (7)$$

- *More "same" than "different" features:* $|J_D| \ll |J_S|$. The motivation is clear: if the second sum is large, and if the features are distributed roughly the same *under* $Y = -1$, then the first sum is unlikely to be larger than the second under $Y = -1$.

As suggested above, it would be desirable to have *stable frequencies under* $Y = -1$, meaning $q_j \approx q$ for all $j \in J_D \cup J_S$, as this would simplify any analysis of the false positive error. Roughly speaking, the false positive rate could then be estimated by standard arguments involving binomial random variables, at least under independence and asymptotic normality assumptions. As it turns out, due to the homogeneity of our feature sets, this is roughly satisfied, as can be seen from the middle histograms in Figure 2. However, this only holds *before* inversion ($X_j \rightarrow 1 - X_j$) for those case in which $X_j = 1$ is rarer under $Y = 1$ than under $Y = -1$; afterwards, the distribution becomes bimodal.

In summary, we attempt to make J_S as large as possible consistent with no misclassifications under $Y = 1$.

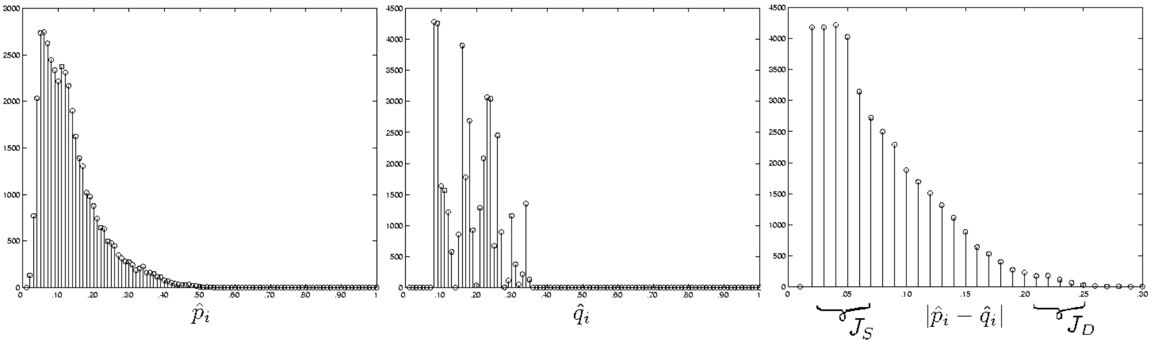

Figure 3. Feature histograms with "spreading" for geometric invariance.

To maintain computational efficiency we restrict J_D to be of order 100. One could pick the 100 features maximizing $|\hat{p}_j - \hat{q}_j|$, but such features might be very dependent and spatially concentrated, resulting in a weak and unstable test. Instead, we borrow a simple, boosting-type device from [4] in order to select both J_D and J_S. The construction is incremental.

Empirically, the constraint (7) is simply

$$\sum_{j \in J_D} x_{i,j} \geq \sum_{j \in J_S} x_{i,j} \quad whenever \quad y_i = 1 \qquad (8)$$

where $x_{i,j}$ is the j'th component of the i'th training sample $\mathbf{x}_i$. The learning algorithm is as follows:

1. Select J_D iteratively. Start with $J_D = J_D(1) = \{j\}$ where j maximizes $\hat{p}_j - \hat{q}_j$. At each iteration $k \geq 2$, determine the positive training examples i for which the current sum, $\sum_{j \in J_D(k-1)} x_{i,j}$, is minimal. Set $J_D(k) = \{j\} \cup J_D(k-1)$ where j is such that $x_{i,j} = 1$ for as many of these examples as possible; in the case of ties, select the index with largest $\hat{p}_j - \hat{q}_j$.
2. Stop when $|J_D| = 100$.
3. Select J_S iteratively. Start with $J_S = J_S(1) = \{j\}$ where j minimizes $\hat{p}_j - \hat{q}_j$. At each iteration $k \geq 2$, choose the index j with smallest difference $\hat{p}_j - \hat{q}_j$ *which preserves the inequality in* (8). Continue adding features to J_S until the inequality can no longer be maintained.

6. Application to Face Detection

Variations in photometry can have a significant effect on the performance of object recognition methods applied to natural scenes. Consequently, we have chosen to illustrate our approach in the context of face detection in greyscale images.

6.1. Review

Standard face detectors apply a face vs. background classifier at several scales and at every image location: different base classifiers, such as neural networks [8], support vector machines [7], Gaussian models [12] and naive Bayesian models [10], have been used. Recent work has focused on serially combining multiple classifiers to yield a faster and more powerful classifier [4, 11, 15]. Each classifier is designed for different levels of invariance, discrimination or computational complexity. Sequential, adaptive testing yields faster rejection of the background and concentrates computation on face-like image patches. Most of these classifiers deal with upright and frontal views of faces. Dealing with large in-plane and out-of-plane rotations is more difficult and will not be considered. Finally, most of these methods use standard image pre-processing techniques to normalize for brightness and contrast variations. Ours does not.

6.2. Training Sets

The standard ORL database is used to synthesize 1600 faces covering different poses. For negative examples, approximately 9000 randomly selected image patches were downloaded from the WWW, from which we synthesized about 90000 "non-faces." *Recall that we only use the negative examples to learn binary statistics*, namely the $\{\hat{q}_j\}$. In particular, the negative examples are not directly utilized during the selection of the sets J_D and J_S.

6.3. Classifier

The base detector is designed to find all faces with tilt restricted to ±20° and size (eye-to-eye distance) $8 - 16$ pixels. To detect larger faces, the original image is downsampled before applying the base detector. With four levels of downsampling we are able to detect faces with sizes from 8 to 128 pixels.

We apply the same basic framework as proposed in [4], where a series of linear classifiers was used in order to gradually reject non-face patterns. The cascade is based on a coarse-to-fine, tree-structured hierarchy of the pose space. Each individual classifier is composed of binary edge variables and is dedicated to faces with poses in a particular cell. The threshold τ for each classifier was chosen to yield

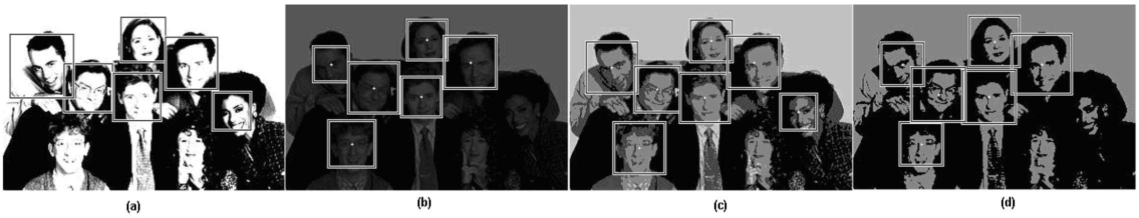

Figure 4. The invariance of SNLT's to image intensity transformations. (a) $\Psi(\mathbf{I}) = 0.02\mathbf{I}^2 + 0.5\mathbf{I} - 100$; (b) $\Psi(\mathbf{I}) = 0.001\mathbf{I}^2 + 0.1\mathbf{I}$; (c) two-bit greyscale quantization; (d) one-bit greyscale quantization.

an apparent null false negative rate; negative training examples are not utilized. The same face detection strategy was employed in [9], except each base classifier is a support vector machine.

In our cascade of face detectors, each classifier is of the form (2) and of course negative examples are utilized. Specifically, the non-face image instances at every cell (node in the tree hierarchy) are those which have responded positively to the preceding classifiers. This ensures that in building the SNLT at each cell we only compete with those background patches which increasingly resemble faces. Exactly the same learning algorithm (§5) is applied for each cell; only the training set changes.

6.4. Image Features

We use the binary family based on comparisons of differences (Example 3 in §4). However, for coarse pose cells (e.g., involving faces with positions spread over an 8×8 block and a nontrivial range of scales and/or tilts), the object feature incidences, $\{\hat{p}_j\}$, will be very small. In order to increase the response rate, and introduce more invariance to *geometric* distortions, we employ the same "spreading" device as in [1] and [4] in which the original features are replaced by disjunctions. In the case of the binary edge detectors based on comparisons of differences, the features are "spread" along a strip orthogonal to the edge direction: the spread edge $X_j^s = 1$ if the original feature is present at any pixel along the strip.

Due to this ORing operation, all the $\hat{p}_j$'s and $\hat{q}_j$'s are increased; the net effect is to create features which are more invariant (appearing on many poses simultaneously) but for which $\hat{p}_j - \hat{q}_j$ can still be large enough for discrimination. Photometric invariance is maintained and nothing changes in the learning algorithm. In Figure 3 we show the same three histograms as in Figure 2 for strips of lengths one, two and three combined. The size of the sets J_S ranges from approximately 200 (coarse pose cells; high spreading) to 600 (fine pose cells; no spreading).

We emphasize that this is but one choice of features that might demonstrate the utility of SNLTs in classification. The optimal set of features for detecting faces is a subject of ongoing research.

6.5. Results

We have implemented our algorithm in C++ on a standard Pentium 4 1.8GHz PC, using the CMU+MIT [8, 12] frontal face test set to estimate performance. As in the cited work on cascades, processing a 320×240 image takes only a fraction of a second.

In order to have an exact comparison between fixed (1) and variable (2) thresholds, we built a fixed-threshold system with exactly the same protocol - same training set, features, architecture (pose decomposition) and choice of discriminating features J_D. In addition, we compare the SNLT-based system with other face detection methods. The results are in Table 1. We achieve a detection rate of 89.6% with 188 false positives on 168 images from the test set. The SNLT-based system attains both a higher detection rate and a lower false positive rate when compared to its fixed-threshold counterpart. Moreover, the results are comparable to other, well-known systems. It should be noted that the results from each system are reported on slightly different subsets of the CMU+MIT test set.

	Detection	False positives / image
SNLT	89.6%	1.11
Constant Threshold	83.5%	2.33
Viola-Jones [1]	90.8%	0.73
Rowley-Baluja-Kanade [1]	89.2%	0.73
Sahbi [2]	89.6%	0.68

Table 1. Detection rates for various face detection systems

We do not show a sample of face detections in the test set because the results would look virtually indistinguishable from those in the cited references. Moreover, the performance of the SNLT-system could very likely be improved by considering a richer training set of faces, employing bootstrapping to reduce the false positive rate, optimizing and/or refining the pose decomposition, and other refine-

[1] Results reported are on 130 images.

[2] Results reported are on 164 images.

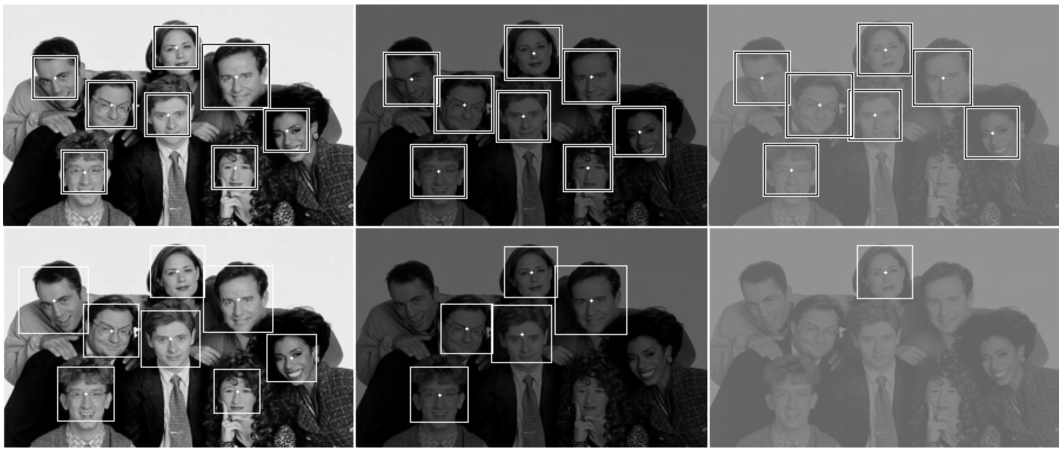

Figure 5. Contrast and brightness invariance for image transformations $\Psi(\mathbf{I}) = \mathbf{aI} + \mathbf{b}$. Top row: Results using SNLTs. Bottom row: Results using a fixed threshold. Column one: Original image; Column two: $\mathrm{a} = 0.4$ and $\mathrm{b} = 0$; Column three: $\mathrm{a} = 0.2$ and $\mathrm{b} = 100$.

ments as in the cited references. This was not our primary objective.

More to the point, in Figure 4 and Figure 5, we show our ability to detect faces under various image intensity transformations. The improvement over fixed-thresholds is clear-cut.

7. Conclusion

Motivated by the lack of robustness of many visual recognition algorithms based on linear classifiers, we have proposed replacing fixed thresholds by variable thresholds based on nondiscriminating features. Normally, such features are discarded as worthless. We retain them for the purpose of normalization. This presupposes that all or most of the features are affected in roughly the same way by varying imaging conditions. Consequently, the performance of SNLTs may depend heavily on the nature of the interaction between the features and the expected data transformations. In the special case of detecting instances from a category of visual objects, and hard-wiring invariance to photometric distortions, SNLTs seem to provide a decided improvement over fixed thresholds when the features themselves scale appropriately.

Acknowledgments

This work was supported in part by ONR contract N000140210053, by ARO grant DAAD19-02-1-0337 and by NSF grant DMS-0219016.

References

[1] Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11:1691–1715, 1999.

[2] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[3] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[4] F. Fleuret and D. Geman. Coarse-to-fine face detection. *International Journal of Computer Vision*, 41:85–107, 2001.

[5] W. Highleyman. Linear decision functions with applications to pattern recognition. *Proceedings IRE*, 50:1501–1514, 1962.

[6] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

[7] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. *Proceedings IEEE CVPR*, pages 130–136, 1997.

[8] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions PAMI*, 20:23–38, 1998.

[9] H. Sahbi. *Coarse-to-fine support vector machines for hierarchical face detection*. PhD thesis, Versailles University, 2003.

[10] H. Schneiderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. *IEEE Proceedings CVPR*, pages 45–51, 1998.

[11] D. A. Socolinsky, J. D. Neuheisel, C. E. Priebe, J. De Vinney, and D. Marchette. Fast face detection with a boosted cccd classifier. Technical report, Johns Hopkins University, 2002.

[12] K. Sung and T. Poggio. Example-based learning for view-based face detection. *IEEE Transactions PAMI*, 20:39–51, 1998.

[13] T. Speed, Ed. *Statistical Analysis of Gene Expression Microarray Data*. Chapman and Hall, 2003.

[14] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[15] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings IEEE CVPR*, 2001.