

550.640 Project 3; Multiclass SVM. Due on Monday March 31rd.

The solution must be written like you would write a report, with explanations and commented results. A list of numbers and figures is not sufficient.

Providing the program sources is optional. They are not graded (so no direct credit for them), but they can help understanding why results are not correct to decide whether partial credit can be given.

Preparation The goal of the project is to use Support Vector Machines for classification on the USPS digit database. This database is available on the class web site. Because this is a 10, and not a 2 class problem, multiple classifiers will be needed. The project compares the implementation of a 1 vs. all strategy to an redundant coding approach.

As a first step, we want to create a validation set within the training set. For this, simply set aside 20% of the examples from the training; this will be the validation set. This set will be used to select the optimal parameters of the classifiers. The remaining of the training set will be called the reduced training set. All this is independent of the test set.

Several SVM packages are available on the class web site. If you use Schwaighofer's matlab package matlab package SVMlight (you don't have to), the following variables, coming from the NET structure are relevant:

- Net.c: the weight coefficient on the errors; called γ in the lecture notes and in this project.
- Net.kernel is a string that defines the type of kernel: here, you will need 'poly' and 'rbf' (Gauss kernel).
- The parameter that instantiates the kernel is in NET.kernelpar(1); for polynomial kernels, it is q such that

$$K(x, y) = (1 + x^T y)^q$$

and for rbf (gaussian) kernels, it is λ such that

$$K(x, y) = \exp(-|x - y|^2 / (d\lambda)),$$

d being the dimension.

If you use the matlab wrapper to SVMlight, training for polynomial kernels uses

```
model = svmlearn(X, Y, '-c gamma -t 1 -d q')
```

where gamma is the constant before the slack variables and q is the power of the kernel. To test, call

```
[err, pred] = svmclassify(Xtest, Ytest, model)
```

The predicted classes being $\text{sign}(\text{pred})$.

For Gaussian kernels, the call is

$$\text{model} = \text{svmlearn}(X, Y, \text{'-c gamma -t 2 -g a'});$$

and the kernel is $\exp(-a|x - y|^2)$.

Part 1

(1) Let $h(x)$ be the feature associated to x (such that $K(x, y) = \langle h(x), h(y) \rangle$). Given x_1, \dots, x_n , give the expression of

$$\sum_{k=1}^n \|h(x_k) - \frac{1}{n} \sum_{l=1}^n h(x_l)\|^2$$

in function of the values of $K(x_i, x_j)$.

(2) Let $((x_1, y_1), \dots, (x_N, y_N))$ be the training set with $q = 10$ classes. For $j = 1, \dots, q$, let N_j be the number of examples with class j and μ_j be the within class average

$$\mu_j = \frac{1}{N_j} \sum_{k=1, y_k=j}^N h(x_k)$$

and

$$\mu = \frac{1}{N} \sum_{k=1}^N h(x_k).$$

Compute, still in function of the kernel, the within class and between class sums of squares in feature space:

$$\begin{aligned} \sigma_w^2 &= \frac{1}{N} \sum_{k=1}^N \|h(x_k) - \mu_{y_k}\|^2 \\ \sigma_b^2 &= \frac{1}{N} \sum_{j=1}^q N_j \|\mu_j - \mu\|^2 \end{aligned}$$

(3) For the polynomial kernel, compute and plot the values of σ_b^2/σ_w^2 for $q = 1, 2, \dots, 10$ and determine the value of q which minimize the ratio.

(4) Same question with the rbf kernel, taking $\lambda = 0.1, 0.2, \dots, 2$.

Part 2. We will use SVM's with polynomial kernels of order $q = 4$. Each classifier will optimize over the parameter γ that penalizes the slack variables. For each value of γ , train the estimator on the reduced training set and optimize the results on the validation set. First try $\gamma = 1, 5, 10, \dots$ to obtain a rough evaluation of γ before refining the search.

Each time a classifier is trained, provide the optimal for γ as well as the two-class confusion matrix of the optimal classifier on the validation set and on the test set.

(2-a) For each class g between 0 and 9, train a SVM that separates g from the rest of the classes. This corresponds to the two class problem $Y = g$ vs. $Y \neq g$.

(2-b) Build the final classifier as follows: given an input, apply the 10 binary classifiers previously trained. If one and only one of them returns a decision $Y = g$ for some class, let g be the answer. Otherwise, the answer is 'unknown'.

Provide the confusion matrix for this classifier on the test set, adding 'unknown' for the possible output classes.

(2-a) Build 10 binary partitions of the 10 classes (a binary partition being a separation of the 10 classes into 2 sets of 5 classes). A possible implementation of this can use the first and last 5 classes returned by the matlab function *randperm*.

(2-b) For each of the binary partitions, optimize an SVM using the reduced training set and the validation test.

(2-c) Build the final classifier as follows: Each of the 10 previous binary classifiers votes for the 5 classes that it selected. The class that receives the most votes is the final answer. In case of a tie, the answer is 'unknown'.

Provide the confusion matrix for this classifier on the test set, adding 'unknown' for the possible output classes.

(3) Redo question 2 using 20 classifiers instead of 10.

Part 3. Repeat Part 1 using rbf (Gaussian) kernels instead of polynomials, using the optimal $\lambda = 10^{-3}$ for kernels written in the form $K(x, y) = \exp(-\lambda|x - y|^2)$.