NUIM/CS401 **Fall 2008**

Assignment 3: SVM Digit Classification!

We will make use of the MNIST handwritten digits dataset as cleaned by Yann LeCun and Corinna Cortes, `http://yann.lecun.com/exdb/mnist/`. You are encouraged to work in small teams, and to share I/O code for reading the dataset and displaying images between teams.

Each team should turn in one report describing the results, with the computer code as an Appendix. Please describe difficulties encountered and how they were overcome. Also, please show anything interesting you noticed about the data or algorithm. And *put your name at the top of the first page!*

Support Vector Machines are trained on a *dichotomy*, meaning two classes: $z_p = \pm 1$. Because of this, the weight vector used for classification can be regarded as the difference between a weighted sum of the positive exemplars ($\mathbf{w}^+$) and a weighted sum of the negative exemplars ($\mathbf{w}^-$):

$$\mathbf{w} = \sum_p \alpha_p \, z_p \, \phi(\mathbf{x}^{(p)}) = \underbrace{\sum_{\{p \,|\, z_p=+1\}} \alpha_p \, \phi(\mathbf{x}^{(p)})}_{\mathbf{w}^+} - \underbrace{\sum_{\{p \,|\, z_p=-1\}} \alpha_p \, \phi(\mathbf{x}^{(p)})}_{\mathbf{w}^-}$$

Classifying a new datapoint $\mathbf{x}$ can then be formulated as

$$\begin{aligned} \mathbf{w} \cdot \phi(\mathbf{x}) &\stackrel{?}{>} \theta \\ (\mathbf{w}^+ - \mathbf{w}^-) \cdot \phi(\mathbf{x}) &\stackrel{?}{>} \theta \\ \mathbf{w}^+ \cdot \phi(\mathbf{x}) &\stackrel{?}{>} \theta + \mathbf{w}^- \cdot \phi(\mathbf{x}) \\ \sum_{\{p \,|\, z_p=+1\}} \alpha_p \, K(\mathbf{x}^{(p)}, \mathbf{x}) &\stackrel{?}{>} \theta + \sum_{\{p \,|\, z_p=-1\}} \alpha_p \, K(\mathbf{x}^{(p)}, \mathbf{x}) \end{aligned}$$

With an LSVM (Linear SVM), ϕ is the identity function, so $\mathbf{w}$ can be considered to live in input space. We can think of the LSVM as matching the new input to two "templates" and checking which is a better match, with θ being a handicap for one of the templates.

However, the digits data is in ten classes. The following questions involve use of the above observation, and also ways of dealing with the situation of having ten classes rather than just two.

1. There are many SVM software packages available on the web. Find one. (Please do not try to implement one yourself: that would be a serious endeavor.)
2. Train (using training data from the digits dataset) eight LSVMs to each distinguish between two (randomly chosen) pairs of digits, e.g., machine one to distinguish "1"s from "2"s, machine two to distinguish "3"s from "0"s, etc. For each, calculate the error rate on the same two digit types from the test set, and show the $\mathbf{w}^+$ and $\mathbf{w}^-$.

3. Train (using training data from the digits dataset) eight LSVMs to each distinguish between two (randomly chosen) pairs of pairs of digit types, e.g., machine one to distinguish "2"s&"6"s from "3"s&"5"s, machine two to distinguish "3"s&"8"s from "4"s&"7"s, etc. For each, calculate the error rate on the same digit types from the test set, and show the $\mathbf{w}^+$ and $\mathbf{w}^-$.

4. There are a number of techniques for using SVMs to classify data with more than two classes.

   (a) Train 10 LSVMs to each distinguish one digit class from all the others, i.e., machine one distinguishes "1"s from "2"s&"3"s&"4"s&"5"s&"6"s&"7"s&"8"s&"9"s&"0"s, etc. Measure generalization performance on the test set, using the obvious classification criterion (must be classified correctly by all ten machine to be correct), and show the $\mathbf{w}^+$ and $\mathbf{w}^-$ for each machine.

   (b) Train 21 LSVMs to each distinguish between an even dichotomy over the digit types, with these dichotomies chosen randomly. E.g., machine one distinguishing "3"s&"0"s&"2"s&"6"s&"4"s from "5"s&"8"s&"7"s&"9"s&"1"s, etc. Classify a digit by running all the machines on it and counting the votes, where each machine votes for five digits. Measure generalization on the test set.

   (c) Repeat the above but use different kernels, and different soft margin parameters, and a different number of machines, in an effort to maximize generalization performance.

Due by email to `barak+cs401-3@cs.nuim.ie` before 23:59 Sun 7-Dec-2008.