

58147 Machine Learning (Spring 2005)

Exercise 5 (Wednesday 2 March)

- Suppose P is as in the random classification noise model (Example 3.2) for some target $f_* \in C$. Give a formula for the true error $\text{err}(h; P)$ as a function of $\Pr_{(x,y) \sim P}(h(x) \neq f_*(x))$. Which h minimises this formula?
 - Suppose now that P can be an arbitrary measure on $X \times \{-1, 1\}$, but is actually known to us. Let h be the classifier $X \rightarrow Y$ that minimises $\text{err}(h; P)$. How is $h(x)$ determined?
 - Again let P be a known measure on $X \times \{-1, 1\}$, but now allow hypotheses h to be functions $X \rightarrow \mathbf{R}$, and let h be the one that minimises the expected squared error $\mathbb{E}_{(x,y) \sim P}[(y - h(x))^2]$. How is $h(x)$ determined this time?
- Prove the following lower bound for the true error in the test set setting:

$$\Pr_{S \sim P^m}(\text{err}(h) \leq \overline{\text{Bin}}(m\widehat{\text{err}}(h; S), m, 1 - \delta)) \leq \delta.$$

Then use the Hoeffding bound to obtain an approximated version of this result.

- Complete the proof sketch of Corollary 3.13 on page 145 of lecture notes. (Use the result from previous problem and the union bound.)
- 4.-5. A data set containing 499 images of cars and 499 images of non-cars is available at

<http://www.cs.helsinki.fi/u/jtlindgre/Lcars.mat.bz2> (Matlab format)
<http://www.cs.helsinki.fi/u/jtlindgre/Lcars.RData.bz2> (R format).

A detailed description of the data is given below in Matlab terms. If you prefer R and have problems, please contact Jussi Lindgren.

Split the data into a training set of 600 examples and validation and test sets of 199 examples each. Run the Perceptron algorithm repeatedly through the training set, until it has converged. This should take less than 20 iterations. After each iteration, store your current weight vector. Thus you should end up with up to 20 different weight vectors. Use the validation set to pick the best one, and then use the Test Set Bound (for which you need to implement the $\overline{\text{Bin}}$ procedure) to estimate the error. Visualise the resulting weight vector (by interpreting it as a pixel array).

(This method of not necessarily running until convergence is known as *early stopping*. It is sometimes useful to avoid overfitting, but in this problem we use it just to illustrate the method.)

The data consists of a $998 \times 40 \times 100$ array `instances` and 998×1 vector `labels`. Each instance is a 40×100 pixel image. To convert image number i to a 4000 dimensional vector `xv`, type `xi=instances(:, :, i)` and then `xv= xi(:)`. To visualise, type `colormap gray` and `imagesc(xi)`. To convert 4000 dimensional vector `w` into an 40×100 matrix (so you can visualise the weight vectors), use `wr= reshape(w, 40, 100)`.