E1 213 Pattern Recognition and Neural networks

Problem Sheet 4

1. Consider a general K-class problem with a general loss function. Let $h(X)$ denote the output of the classifier on X. Let $R(\alpha_i|X)$ denote the expected loss when classifier says α_i and conditioned on X. That is, $R(\alpha_i|X) = E[L(h(X), y(X))|h(X) = \alpha_i, X]$, where, as usual, $y(X)$ denotes the 'true class'. We had only considered deterministic classifiers where h is a function that assigns a unique class label for any given X. Suppose we use a stochastic classifier, h, which, given X, outputs α_i with probability $p_h(\alpha_i|X)$. (Note that we would have $p_h(\alpha_i|X) \geq 0$ and $\sum_i p_h(\alpha_i|X) = 1$). For this classifier, show that the risk is given by

$$R(h) = \int \left[\sum_{i=1}^{K} R(\alpha_i|X) p_h(\alpha_i|X) \right] f(X)\, dX$$

where $f(X)$ is the density of X. Using the above expression, find the best choice of values for all the $p_h(\alpha_i|X)$ and hence conclude that we do not gain anything by making the classifier stochastic.

Answer: The actual algebra involved is not too complicated and I hope you are able to show this.

To understand why stochasticity in predicting class label may not help, consider the following simple situation. Suppose there is a coin whose probability of heads is 0.6 (and we know this). We want a strategy to play a gambling game of predicting the outcome of each toss with this coin. Suppose our strategy is to predict heads with probability p and tails with probability $(1 - p)$, where p is a parameter that we choose. Calculate the probability of correct prediction. For what value of p is this probability maximized?

2. Consider a two class problem with one dimensional feature space. Suppose we have six training samples: x_1, x_2, x_3 from one class and x_4, x_5, x_6 from the other class. Suppose we want to estimate the class conditional densities nonparametrically through a Parzen window estimate with Gaussian window with width parameter σ. Write an expression for the Bayes classifier (under 0–1 loss function) which uses these estimated densities.

Answer: let $\hat{f}_0$ and $\hat{f}_1$ be the two estimated class conditional densities. Then

$$\hat{f}_0(x) = \frac{1}{3}\sum_{i=1}^{3} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-x_i)^2}{2\sigma^2}\right)$$

$$\hat{f}_1(x) = \frac{1}{3}\sum_{i=4}^{6} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-x_i)^2}{2\sigma^2}\right)$$

The bayes classifier is

$$h(x) = \begin{cases} 0 & \text{if } p_0\hat{f}_0(x) > p_1\hat{f}_1(x) \\ 1 & \text{otherwise} \end{cases}$$

3. Consider a non-parametric estimate of a density function given by

$$\hat{f}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_n}\phi\left(\frac{x-x_i}{h_n}\right)$$

Let the function ϕ be Gaussian. That is

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right), \quad -\infty < x < \infty$$

Suppose the true density from which samples are drawn is Gaussian with mean μ and variance σ^2. Calculate $E\hat{f}_n(x)$. What will be its limit as $n \to \infty$?

Answer: $E\hat{f}_n(x)$ is $\mathcal{N}(\mu, \sigma^2 + h_n^2)$. It converges to the true underlying density as $n \to \infty$. (You can show it by using the idea of 'completing the squares' in a gaussian density integral as is done in deriving bayesian estimate for gaussian density in the class).

4. Consider 2-class PR problems with n Boolean features. Consider two specific classification tasks specified by the following: (i) a feature vector X should be in Class-I if the integer represented by it is divisible by 4, otherwise it should be in Class-II; (ii) a feature vector X should be in Class-I if it has odd number of 1's in it, otherwise it is in Class-II. In each of these two cases, state whether the classifier can be represented by a Perceptron; and, if so, show the Perceptron corresponding to it; if not, give reasons why it cannot be represented by a Perceptron.

Answer: (i). Can be represented by a perceptron. (ii) Cannot be represented by a perceptron.

5. Consider the incremental version of the Perceptron algorithm. The algorithm is: at iteration k, if $W(k)TX(k) \leq 0$ and thus we misclassified the next pattern then we correct the weight vector as: $W(k+1) = W(k) + X(k)$.
   (i). By going over the proof presented in class, convince yourself that if we change the algorithm to $W(k+1) = W(k) + \eta X(k)$ for any positive step-size η.
   (ii). In the perceptron algorithm, when we misclassify a pattern and hence correct the weight vector, the algorithm does not necessarily ensure that $W(k+1)$ will classify $X(k)$ correctly. Suppose we want to change the algorithm so that when we misclassify a pattern, we change the weight vector by an amount that ensures that after the correction, the weight vector correctly classifies this pattern. While this may seem like just a matter of choosing a 'step-size', note that if we want to choose η so that the above is ensured at every k then, the 'step-size' may have to vary from iteration to iteration and it may be a function of the feature vector. Hence, the earlier proof does not go through. Design a version of the Perceptron algorithm which effectively ensures the above property and for which the same convergence proof holds.

Answer: Think of the following idea for order of presenting examples in the original perceptron algorithm. We present examples one by one by going over all examples sequentially. However, if an example is misclassified, after correcting the weight vector, we present the same example again.