MAS 622J: Pattern Recognition and Analysis

Problem Set 5

For this problem set you will compare three different classification techniques in the same data set. The training and testing data can be obtained from the class Web page. There are two classes with equal prior probabilities and scalar-valued features. Each classification technique has free parameters which you will estimate via the technique of *leave-one-out validation* (to be explained).

1. Use the Parzen window technique in DHS 4.3 to estimate the density of each class. Use the Gaussian window (equation (26) in DHS). Unfortunately, the width parameter h is still to be determined for each class.

   To estimate h, you will use the amazing technique of leave-one-out validation. The idea is to compute an *evidence curve* $v(h)$ which approximately represents the likelihood of h given the data. Then you can choose the h which maximizes $v(h)$. Each class has its own density so this needs to be done separately for each class. Here's what you do to compute $v(h)$ for a particular value of h:

   (a) Break the training data into two parts A and B, where B contains only a single sample. This is what makes the method "leave-one-out" validation.

   (b) Compute the Parzen window density estimate $p_n(x)$ from A only.

   (c) Compute $\log(p_n(B))$, the log-probability of the sample you held out.

   (d) Iterate back to part (a) and break the training data up in a different way. Do this for every possible choice of B. The average of all of the log-probabilities you get in part (c) is defined to be $v(h)$.

   For each class, plot the evidence curve. Pick three values of h from different parts of the curve and plot the estimated density using each. What would your intuition say is the best h? How well does the evidence curve match your intuition?

   Now design a minimum error rate classifier using the h that maximizes $v(h)$ for each class. What is its performance on the test data?

2. Use the generalized linear discriminant in DHS 5.3 in conjunction with the pseudoinverse technique in DHS 5.8.1. In other words, an augmented data vector will be

$$\mathbf{y} = \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^k \end{bmatrix}$$

for some polynomial degree k. Let the margin vector $\mathbf{b} = \mathbf{1}$, so that the weight vector $\mathbf{a}$ can be computed from the pseudoinverse of the data matrix. However, the degree k is still to be determined.

To estimate k, compute an evidence curve $v(k)$ and then choose the k which maximizes $v(k)$. Here's what you do for a particular k:

(a) Break the training data into two parts A and B, where B contains only a single sample.

(b) Compute the generalized linear discriminant from A only.

(c) Determine if the sample you held out is classified correctly or not.

(d) Iterate back to part (a) and break the training data up in a different way. Do this for every possible choice of B. The number of correct classifications in part (c) is defined to be $v(k)$.

In this process, it is helpful to first negate the augmented data from class 2, as suggested in DHS 5.4.1.

Plot the evidence curve from $k = 1$–20. Pick three values of k from different parts of the curve and plot the sign of the discriminant function over the scalar feature space (it should be 1 when choosing class 1 and -1 otherwise). Design a minimum error rate classifier using the k that maximizes $v(k)$. What is its performance on the test data?

3. Use the k-nearest neighbor method in DHS 4.5.4. Again, we do not know the right value of k and will estimate it using an evidence curve $v(k)$. Here's what you do for a particular k:

(a) Break the training data into two parts A and B, where B contains only a single sample.

(b) Determine if B is classified correctly by its k-nearest neighbors from A.

(c) Iterate back to part (a) and break the training data up in a different way. Do this for every possible choice of B. The number of correct classifications in part (b) is defined to be $v(k)$.

Plot the evidence curve for odd values of k from 1 to 19. Pick three values of k from different parts of the curve and plot the resulting discriminant function over the scalar feature space (it should be 1 when choosing class 1 and -1 otherwise). What would your intuition say is the best k? Design a minimum error rate classifier using the k that maximizes $v(k)$. What is its performance on the test data?