# SD 372 Pattern Recognition

*Winter 2009*

Lab 2: Model Estimation and Discriminant Functions
Due Friday, March 13, 2009.

## 1   Purpose

This lab examines the areas of statistical model estimation and classifier aggregation. Model estimation will be performed by implementing parametric and non-parametric estimators. Aggregation is introduced by combining several simple linear discriminants into one more powerful classifier.

## 2   Model Estimation 1-D case

For this part of the lab, use the data set provided on the course homepage:
http://ocho.uwaterloo.ca/~pfieguth/Teaching/372/lab2_1.mat

There are two data sets:

- variable $a$ - a bunch of Gaussian samples, $\mu = 5$, $\sigma = 1$.

- variable $b$ - a bunch of Exponential samples, $\lambda = 1$.

Do the following steps for **each** data set:

1. **Parametric Estimation – Gaussian**: Assume that the unknown density is Gaussian; this leaves two parameters to estimate: the mean and the variance. Use Maximum Likelihood to estimate these parameters. Plot the resulting estimated $\hat{p}(x)$ superimposed on the true $p(x)$.

2. **Parametric Estimation – Exponential**: Assume that the unknown density is Exponential; this leaves one parameter to estimate: parameter $\lambda$. Use Maximum Likelihood to estimate this parameter. Plot the resulting estimated $\hat{p}(x)$ superimposed on the true $p(x)$.

3. **Parametric Estimation – Uniform**: Assume that the unknown density is Uniform; this leaves two parameters to estimate: parameters $a$ and $b$. Use Maximum Likelihood to estimate these parameters. Plot the resulting estimated $\hat{p}(x)$ superimposed on the true $p(x)$.

4. **Non-parametric estimation**: Estimate the density using the Parzen method. Use Gaussian windows having standard deviations of 0.1 and 0.4. Generate two plots, one for each standard deviation. Plot the true density on top of the estimated density.

For each of the two data sets, which of the estimated densities is closest to the original? Give a qualitative comparison of the results. In general, is it possible to always use a parametric approach? When is it better to use a parametric method? When is the non-parametric approach preferred?

# 3   Model Estimation 2-D case

For this part of the lab, use the data set provided on the course homepage: http://ocho.uwaterloo.ca/~pfieguth/Teaching/372/lab2_2.mat

Here we will see the full power of the non-parametric approach. Data points for the three classes are stored in $(x, y)$ format in variables $al$, $bl$, and $cl$.

1. **Parametric estimation:**

   Assume that each cluster is normally (Gaussian) distributed. Using the data, compute the sample mean and sample covariance of each cluster. Since plotting multiple 2D distributions is messy, instead find and plot the ML classification boundaries. Show the cluster data, superimposed with the classification boundaries, on the plot.

2. **Non-parametric estimation:**

   Use a Gaussian Parzen window ($\sigma^2 = 400$) on the learning data to estimate a PDF for each cluster. Apply an ML classifier to the esti-

mated PDFs and plot the classification boundaries together with the cluster data.

(Matlab code to implement 2D Parzen windows has been provided on the course website.)

Give a qualitative comparison of the classification results. In general, is it possible to always use a parametric approach? When is it better to use a parametric method? When is the non-parametric approach preferred?

# 4 Sequential Discriminants

For this part of the lab, use the data set provided on the course homepage: http://ocho.uwaterloo.ca/~pfieguth/Teaching/372/lab2_3.mat

Points for the two classes are stored in $(x, y)$ format in variables $a$ and $b$.

What we want to do is to develop a *sequential* classifier. For any given discriminant $G$, we can use the entire data set to work out the following probabilities:
$$P(\text{true class is } C_i | G \text{ says } C_k)$$
For a good sequential classifier, what we want are discriminants that get some part of some class exactly right, so that

$$P(\text{true class is } C_i | G \text{ says } C_i) = 1$$

for at least *one* class $C_i$.

Here's what we'll do:

1. Let $a$ and $b$ represent the data points in classes $A$ and $B$. Let $j = 1$.

2. Randomly select one point from $a$ and one point from $b$

3. Create a discriminant $G$ using MED with the two points as prototypes

4. Using all of the data in $a$ and $b$, work out the confusion matrix entries

$$n_{aB} = \#\text{times } G \text{ classifies a point from } a \text{ as class } B$$

$$n_{bA} = \#\text{times } G \text{ classifies a point from } b \text{ as class } A$$

5. If $n_{aB} \neq 0$ and $n_{Ba} \neq 0$ then no good, go back to step 2.

6. This discriminant is good; save it as

$$G_j = G, \quad n_{aB,j} = n_{aB}, \quad n_{bA,j} = n_{bA}$$

   Let $j = j + 1$.

7. If $n_{aB} = 0$ then remove those points from $b$ that $G$ classifies as $B$.

8. If $n_{bA} = 0$ then remove those points from $a$ that $G$ classifies as $A$.

9. If $a$ and $b$ still contain points, go back to step 2.

At this point we have a sequence of discriminants, each of which classifies some part of the problem perfectly. The overall classifier for some given point $\underline{x}$ is sequential, passing through $G_1, G_2, \ldots$ until a classification is made:

1. Let $j = 1$

2. If $G_j$ classifies $\underline{x}$ as class B and $n_{aB,j} = 0$ then "Say Class B"

3. If $G_j$ classifies $\underline{x}$ as class A and $n_{bA,j} = 0$ then "Say Class A"

4. Otherwise $j = j + 1$ and go back to step 2.

## Deliverables

1. Learn three sequential classifiers, and for each one plot the resulting classification boundary along with the data points.

2. If we test our classifier on the training data, what will its probability of error be? Discuss.

3. In the above development we did not limit the number of sequential classifiers. Suppose we limit the sequential classifier to $J$ classifiers $G_1, \ldots, G_J$. We want to see how the experimental error rate varies with $J$. For each value of $J = 1, 2, \ldots, 5$, learn a sequential classifier 20 times to calculate the following:

   (a) the average error rate
   (b) minimum error rate

(c) maximum error rate

(d) standard deviation of the error rates

Produce a plot showing these results as a function of $J$.

In our sequential classifier, we assumed that we could keep looking indefinitely for a classifier that would classify elements of some class perfectly. How might the results of the sequential classifier differ if I limited the number of point pairs that you could test?

# 5   Report

Include in your report:

- A brief introduction.

- Discussion of your implementations and results.

- Printouts of pertinent graphs (properly labeled).

- M-files for each section.

- Include responses to all questions.

- A brief summary of your results with conclusions.