# Module: EEEM007 ADVANCED SIGNAL PROCESSING

Year: 2018/2019. Examiner: M D Plumbley

Date Due: 4pm Tuesday 7 May 2019 (Week 10)

# LAB EXPERIMENT: PATTERN RECOGNTION

## 1. INTRODUCTION

The aim of the experiment is to provide a practical support for the main aspects of the material covered in EEEM007 Advanced Signal Processing, namely *Pattern Classification*. The experiment is designed to reinforce the main theoretical results learnt in the course and to enable the student to gain intuitive feeling about the effects of classifier design factors (such as training set size, class separability, feature space dimensionality, classification rule, class conditional distribution model) on the classification system performance. It should also validate experimentally the results derived in the Assignment.

Two popular classifiers are investigated in the experiment: i) the Bayes decision rule for normally distributed classes and ii) the k-Nearest Neighbour decision rule. The latter assigns an unknown pattern $\mathbf{x}$ to the class decided by the majority vote of its k-Nearest Neighbours (k-NN) drawn from the training set. Each training pattern in the set of k-NNs votes for the class to which it belongs. For large values of $k$ the k-NN decision rule will approach the performance of the Bayesian classifier while for $k = 1$ the error probability can be as much as twice the Bayes error. The rule is computationally costly but it has the advantage that it can be applied even when the classes are nonparametrically distributed.

## 2. MATLAB COMMANDS

You may find the following Matlab commands useful:

| | |
|---|---|
| mvnrnd | generate random numbers |
| fitcnb | construct Gaussian classifier |
| fitknn | construct k-NN classifier |
| predict | function for finding class labels |
| mahal | Mahalanobis distance |

Include the Matlab code you used for your experiments in an Appendix to your report.

# 3. EXPERIMENT DESCRIPTION

## 3.1 Experiment 1

The aim of this experiment is to investigate the effect of training sample size on the classifier performance. A Gaussian classifier for discriminating between two 2-dimensional classes will be used for the study.

Use the class parameters $(a, b, c, d, f)$ corresponding to your Assignment. Generate a design set $X_D$ and test set $X_T$, each containing 100 patterns per class, distributed according to the selected class parameters.

Design a Gaussian classifier using, respectively, $N_D = 3,5,10,50,100$ training samples from the design set for each class.

Test the designs with the same training samples to obtain error estimates $e_{\text{design}}(N_D)$. Test each classifier also using the full test set to obtain error estimate $e_{\text{test}}$. Repeat the experiment ten times for independent design sets $X_D^i, i = 1,\ldots 10$ (the same test set may be used) and average the estimated errors to obtain

$$E_{\text{design}}(N_D) = \frac{1}{10}\sum_i e^i_{\text{design}}(N_D) \qquad\qquad E_{\text{test}}(N_D) = \frac{1}{10}\sum_i e^i_{\text{test}}(N_D).$$

Plot the average errors as a function of $N_D$ and compare them with the theoretical error.

Comment on your results.

## 3.2 Experiment 2

In this experiment we shall investigate the dependence of the $E_{\text{test}}(N_D)$ curves on the dimensionality of the pattern recognition problem.

As the determination of the true error probability in high dimensional spaces is difficult, we shall take the estimated error $E_{\text{test}}(500)$ as the true error.

Try to estimate $E_{\text{test}}(N_D)$ for values $N_D = 3,5,10,20,50,100,200$ for two class problems in $d = 5,10,15$ dimensional spaces.

If you are unable to design the classifier, consider what is the minimum number of training samples required as a function of dimensionality and why. You may choose the covariance matrix to be an identity matrix in this exercise. Choose the mean vectors so that the error probability is maintained in the range 5-10%.

How many training samples as a function of dimensionality do you need to achieve a reasonable performance (close to the true error rate)?

## 3.3 Experiment 3

In this experiment we shall investigate the effect of the size of test set on the reliability of the empirical error count estimator.

Design a Gaussian classifier for a two class problem in a five dimensional space so that its error probability is in the range 1-10%. Generate ten independent test sets $X_T^i, i = 1,\ldots 10$ of size 500 using the same class conditional distribution parameters. Choose several different numbers $N_T$ of test patterns covering the range 5 to 500. For each $N_T$ and each $X_T^i$ obtain an estimate $e^i(N_T)$ of the classifier error probability.

Find the mean value $E(N_T)$ and the variance $\sigma^2(N_T)$ of the estimated error, i.e.

$$E(N_T) = \frac{1}{10}\sum_i e^i(N_T) \qquad\qquad \sigma^2 = \frac{1}{9}\sum_i \left[e^i(N_T) - E(N_T)\right]^2$$

Plot your results and comment on how they compare with your theoretical predictions.

### 3.4 Experiment 4

The aim of this experiment is to explore the relationship between class separability and error probability.

Choosing a suitable pattern space dimensionality, $d$, generate a sequence of sets of normally distributed training data containing patterns from two classes. The Mahalanobis distance between the means of the two classes in each set should be different, and it should be monotonically increasing with the rank of the set in the sequence. Estimate the error probability of the classifier in each case and plot it as a function of Mahalanobis distance. Comment on your findings.

In your report, give the details of you error estimation scheme (number of test samples).

### 3.5 Experiment 5

The goal of this experiment is to study the effect, on the classifier error probability, of discrepancies between the true and assumed class probability distribution models. We may simulate this situation very simply by attempting to classify patterns drawn from normal distributions with general covariance matrices using the nearest mean classifier which is Bayes optimal only for classes with an identity covariance matrix.

To keep things simple we shall confine our exercise to a two class problem in a two dimensional space. To design a nearest mean classifier, generate a training data set with means $\mu_i, i = 1,2$ and an identity covariance matrix. Test the corresponding classifier with test patterns from normally distributed classes with the same means and identical covariance matrices which differ from the identity matrix. Make sure that both the Mahalanobis distance between classes and the determinant of the covariance matrix are maintained constant in order to ensure that the class overlap is comparable. This can be achieved by means of a judicial choice of the mean vector components of the two classes, especially if the class covariance matrices are kept diagonal.

Now estimate the classifier error probability and compare it with that of the classifier designed using a training set of patterns drawn from the same distribution as the test set. Comment on your results.

### 3.6 Experiment 6

Repeat Experiment 1 using the k-NN classifier for all odd values of $k$ in the range 1 to 51, noting that $k$ cannot exceed $2N_D$. For a representative range of $k$ compare your results with those obtained for the Gaussian classifier in Experiment 1.

For each value of $N_D$ select the best result $E^*(N_D)$ (smallest error) over all values of $k$ and record the corresponding $k^*(N_D)$. Plot these best results as a function of $N_D$ in the same graph as that used for presenting the results of Experiment 1. Comment on your results.

Plot also $k^*(N_D)$ in the same graph. Comment on your results and try to explain them.

## 4. REPORT FORMAT

The Report should be a single document, consisting of descriptions and results for the experiments (approx 9-12 pages), plus an Appendix containing text listings of your Matlab code.

[Remember: The TurnItIn submission system will identify similar reports, so please write your own Matlab code and experiment descriptions.]

## 4. MARKING SCHEME

For each experiment:

1. Description of experiment, objectives, design choices [20%]
2. Predicted experimental outcome from the theory [20%]
3. Presentation of the experimental results obtained [20%]
4. Analysis of results, discussion and conclusions [20%]
5. Presentation of the Matlab code used [20%]

Each of Experiments 1 to 6 is worth 1/6 of the final mark.