ENGINEERING TRIPOS PART IIB

Tuesday 25 April 2006 2.30 to 4

Module 4F10

STATISTICAL PATTERN PROCESSING

Answer not more than ***three*** *questions.*

All questions carry the same number of marks.

The ***approximate*** *percentage of marks allocated to each part of a question is indicated in the right margin.*

There are no attachments.

STATIONERY REQUIREMENTS
Single-sided script paper

SPECIAL REQUIREMENTS
Engineering Data Book
CUED approved calculator allowed

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

1 (a) The class-conditional distribution in a classifier for d dimensional data is to be either an M-component Gaussian mixture model (GMM) with each Gaussian component using a diagonal covariance matrix, or a full covariance Gaussian model.

(i) Compare the two alternate forms of distribution in terms of the nature of data that they can model. Discuss how the generalisation of the classifier would be expected to vary as d increases for each of these distribution types. [20%]

(ii) Compare the computational cost of calculating the log likelihood for each of these distributions, stating any assumptions made. [15%]

(b) An M-component mixture model is to be estimated using N samples of single dimensional data $x_1,\ldots,x_N$. The probability density function associated with the m^{th} component is $p(x_k|m)$ and the component prior c_m.

(i) Write down the log likelihood $l(\theta)$ of the training data where θ is the parameter vector of the model. [10%]

(ii) Show that the partial derivative of $l(\theta)$ with respect to a particular parameter associated only with the m^{th} component, θ_m, is [10%]

$$\frac{\partial l(\theta)}{\partial \theta_m} = \sum_{k=1}^{N} P(m|x_k)\frac{\partial \ln [p(x_k|m)c_m]}{\partial \theta_m}$$

(iii) If $p(x_k|m)$ is Gaussian, find the gradient of the log likelihood with respect to both the mean and the standard deviation. Hence state how the maximum likelihood estimates of the mean and variance parameters can be obtained using a *gradient descent* procedure. [25%]

(iv) Compare the use of gradient descent and *Expectation-Maximisation* (EM) for the maximum likelihood estimation of the parameters of a Gaussian mixture model. [20%]

2 For a two-class statistical pattern recognition problem a Bayes' minimum error rate classifier is to be used. The class-conditional probability density functions are $p(x|\omega_1)$ and $p(x|\omega_2)$ and the prior probabilities are $P(\omega_1)$ and $P(\omega_2)$ respectively. The feature vectors are of dimension d.

(a) What is the decision rule for the minimum error classifier? [15%]

(b) The classifier divides the space into two regions. The data is classified as ω_1 in $\mathscr{R}_1$ and ω_2 in $\mathscr{R}_2$. What is the probability of error in terms of class-conditional densities, the class priors and the regions $\mathscr{R}_1$ and $\mathscr{R}_2$? [15%]

(c) The class conditional probability density functions are known to be multivariate Gaussians, both with an identity covariance matrix, I. The mean for class ω_1 is μ_1 and for ω_2 is μ_2. The priors are now known to be equal. Bayes' minimum error rule is used to design the classifier.

(i) Find an equation, expressed in terms of μ_1 and μ_2, that is satisfied by a point, x, that lies on the decision boundary. [20%]

(ii) By considering the distributions in the direction of the line joining the class means, show that the probability of error, P_e, can be expressed as

$$P_e = \frac{1}{\sqrt{2\pi}} \int_a^{\infty} e^{-\frac{v^2}{2}} dv$$

and find an expression for a. [30%]

(d) Why do practical classifiers normally have an error rate higher than the value of the Bayes' minimum error classifier? [20%]

3 A *Multi-Layer Perceptron* (MLP) is to be trained using error back-propagation using a least squares error criterion and for each input pattern, $\mathbf{x}$, the target vector is $t(\mathbf{x})$. The input to the MLP is d dimensional, it has L layers, and $N^{(k)}$ units in the k^{th} layer.

The nodes use a hyperbolic tangent activation function of the form

$$y = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}$$

(a) Find the differential of the node activation function in terms of the output of the node. [15%]

(b) Give an advantage of using the hyperbolic tangent activation function over a sigmoid logistic regression function, for the hidden nodes. [10%]

(c) Find the partial derivative of the output error with respect to a weight going from the final hidden layer to the output layer. [20%]

(d) Show how the partial derivative of the output error with respect to a weight going to the final hidden layer can be computed. [25%]

(e) A gradient descent scheme is to be used to update the weights of the network.

(i) What factors need to be considered in setting the learning rate? [10%]

(ii) What are the differences between *batch* and *sequential* weight updates, and what are the advantages and disadvantages of each method? Consider a range of network weights and training set sizes. [20%]

4 A classifier is required for a two-class problem. There are m, d-dimensional, training samples $\mathbf{x}_1$ to $\mathbf{x}_m$. *Principal Component Analysis* (PCA) is initially to be used to transform the training examples.

(a) What is the assumption that underlies the use of PCA to extract useful features? What are the possible advantages of using the training examples transformed using PCA over using the original feature vector? [20%]

(b) When $d = 3$, the covariance matrix of all the data, Σ, is found to be

$$\Sigma = \begin{bmatrix} 5 & 3 & 0 \\ 3 & 5 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

What are the two principal component directions for this data? Describe how these direction vectors are used to obtain the transformed training samples. What is the covariance matrix for the 2-dimensional transformed feature vector? [35%]

(c) A kernelised version of PCA, kernel-PCA, is to be used. For kernel-PCA the direction of the k^{th} principal component, $\mathbf{v}_k$, can be expressed as

$$\mathbf{v}_k = \sum_{i=1}^{m} \alpha_i^{(k)} \Phi(\mathbf{x}_i)$$

where $\Phi(\cdot)$ is the mapping from the original input-space to the feature-space and the parameters $\alpha_1^{(k)}$ to $\alpha_m^{(k)}$ determine the k^{th} direction. A Gaussian kernel is to be used.

(i) What is the advantage of using this kernelised version of PCA rather than standard PCA? [15%]

(ii) Give the expression for the Gaussian kernel between points $\mathbf{x}_i$ and $\mathbf{x}_j$, $k(\mathbf{x}_i, \mathbf{x}_j)$. What is the relationship between this kernel and the points in the feature-space $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$? [15%]

(iii) How can kernel-PCA be used to obtain the transformed features? Why is a kernel used, rather than using the transformed points in the feature-space? [15%]

5 Figure 1 shows a *Bayesian Network* (BN) for a set of events. The events and associated variables are: an earthquake occured (E); an earthquake was reported on the radio (R); the house alarm was set-off (A); a burglar entered the house (B); and a neighbour telephoned the owner to report that the alarm was ringing, (T).

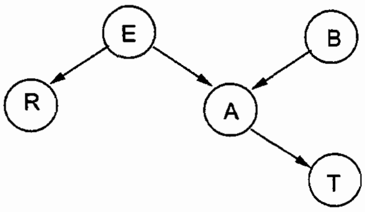

Fig. 1

(a) Describe in words the set of dependencies indicated by the BN in Fig. 1. Write the joint probability distribution, $P(R,E,A,B,T)$ making use of all the conditional independences shown in Fig. 1. [25%]

(b) Message passing is to be implemented for the BN of Fig. 1.

(i) Briefly discuss why message passing is a useful approach to performing inference with complicated BNs. [10%]

(ii) By drawing the *moral* graph associated with the BN, find the set of cliques and separators associated with the BN of Fig. 1. [20%]

(iii) Using the cliques and separators generated in part b(ii), redraw Fig. 1 clearly showing the cliques and the variables associated with the messages being passed between the cliques. [20%]

(c) In practice it is necessary to train the conditional probabilities associated with each of the events. For a particular training set only R, E, B and T are observed. Discuss how the set of probabilities may be trained using *Expectation-Maximisation* (EM), clearly stating the auxiliary function that is to be optimised. [25%]

END OF PAPER