Version MJFG/4

EGT3
ENGINEERING TRIPOS PART IIB

---

Tuesday 19 April 2016 2 to 3:30

---

**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.*

*Write your candidate number **not** your name on the cover sheet.*

**STATIONERY REQUIREMENTS**
Single-sided script paper

**SPECIAL REQUIREMENTS TO BE SUPPLIED FOR THIS EXAM**
CUED approved calculator allowed
Engineering Data Book

**10 minutes reading time is allowed for this paper.**

**You may not start to read the questions printed on the subsequent pages of this question paper until instructed to do so.**

1 A set of sensors are used to detect whether a signal is generated from class $\omega_1$ or class $\omega_2$. Each sensor measures a different feature of the signal. For each class the feature value is fixed. All the features are known to be independent. For all sensors the noise on the measurement is Gaussian distributed with zero mean and variance $\sigma^2$. For class $\omega_1$ the feature value for sensor $s$ is $\mu_{s1}$ and for class $\omega_2$ it is $\mu_{s2}$. A generative classifier is to be used to classify the source of the signal, the prior probabilities for the two classes are equal.

(a) State Bayes' decision rule for generative classifiers for binary classification tasks. [10%]

(b) Initially $d$ sensors, numbered 1 to $d$, are used to determine the source of the signal. The vector of $d$ sensor measurements will be written as $\mathbf{x}$.

(i) Derive an expression for a sensor measurement $\mathbf{x}$ that lies on the optimal decision boundary. [20%]

(ii) Show that the minimum probability of error for this set of sensors, $P_{\mathsf{e}}$, can be expressed in the form:

$$P_{\mathsf{e}} = \int_{-\infty}^{c} \mathcal{N}(z;0,1)dz$$

What is the value of $c$? [30%]

(c) To improve the accuracy of the classifier, the number of sensors is increased to $2d$, sensors numbered 1 to $2d$.

(i) Derive an expression for the reduction in probability of error rate from increasing the number of sensors. Can increasing the number of sensors increase the probability of error? [15%]

(ii) Rather than using all $2d$ sensors, a subset of $k$ sensors are to be selected for use in the classifier. How can the optimal subset of sensors be selected? [15%]

(iii) In a practical system, where the parameters of the classifier need to be trained, discuss the limitations of continually increasing the number of sensors. [10%]

2 An $M$-component Gaussian mixture model with diagonal covariance matrices is to be used as the probability distribution for a $d$-dimensional feature vector. There are $N$ independent training examples, $\mathbf{x}_1, \dots, \mathbf{x}_N$, to estimate the model parameters. The parameters of the model are to be estimated using Maximum Likelihood (ML) estimation.

(a) Find an expression for the log-likelihood of the training data in terms of the component priors, $c_1, \dots, c_M$, and the component parameters. [10%]

(b) Expectation-Maximisation (EM) is to be used to find the Gaussian component means. The auxiliary function for this problem can be expressed as

$$Q(\theta, \hat{\theta}) = \sum_{i=1}^{N} \sum_{m=1}^{M} P(\omega_m | \mathbf{x}_i, \theta) \log(p(\mathbf{x}_i | \omega_m, \hat{\theta}))$$

where $\theta$ is the set of all the model parameters and $\hat{\theta}$ the parameters to be estimated. Constant terms have been ignored in this expression.

(i) Show that the update formula for the mean of component $\omega_m$ is

$$\hat{\mu}_m = \frac{\sum_{i=1}^{N} P(\omega_m | \mathbf{x}_i, \theta) \mathbf{x}_i}{\sum_{i=1}^{N} P(\omega_m | \mathbf{x}_i, \theta)}$$

[25%]

(ii) The diagonal covariance matrix for all components of the model are restricted to be the same. Derive an expression to estimate the single covariance matrix $\Sigma$. [20%]

(c) The form of the model is now changed so that the means of all the components are restricted to be the same, $\mu$. Additionally the covariance matrices are scalar multiples of a single, diagonal, covariance matrix $\Sigma$. Thus the mean for component $m$ is $\mu$ and the covariance matrix is $a_m\Sigma$.

(i) Derive the update formula for the value of $a_m$. Discuss any issues that need to be considered when estimating $a_m$. [30%]

(ii) Discuss the differences between using this form of model compared to using the form discussed in part (b). [15%]

3 The parameters of a discriminative classifier are to be trained using a quadratic approximation to the error surface. The set of parameters associated with the classifier are denoted as the vector $\theta$. An iterative procedure is used to estimate the parameters where at iteration $\tau+1$

$$\theta^{(\tau+1)} = \theta^{(\tau)} + \Delta\theta^{(\tau)}$$

and $\theta^{(\tau)}$ is the estimate of the model parameters at iteration $\tau$. The value of the cost function with model parameters $\theta$ is $E(\theta)$.

(a) The following quadratic approximation is to be used to estimate the weights

$$E(\theta) \approx E(\theta^{(\tau)}) + (\theta - \theta^{(\tau)})'\mathbf{b} + \frac{1}{2}(\theta - \theta^{(\tau)})'\mathbf{A}(\theta - \theta^{(\tau)})$$

(i) By considering a second-order Taylor series expansion about the point $\theta^{(\tau)}$ find expressions for **b** and **A**. [15%]

(ii) Derive an expression for the value of $\theta$ that will minimise this quadratic approximation. Hence obtain an expression for $\Delta\theta^{(\tau)}$. [25%]

(b) An alternative second-order approximation is to assume that all the elements of $\theta$ are independent. Furthermore, for some scenarios it is only possible to compute the gradient. Using this form of approximation, show that a suitable update for element $i$ at iteration $\tau+1$, using only the gradient at the current point, the gradient at the previous point and the previous change, is

$$\Delta\theta_i^{(\tau)} = \left(\frac{g_i^{(\tau)}}{g_i^{(\tau-1)} - g_i^{(\tau)}}\right)\Delta\theta_i^{(\tau-1)}$$

where

$$g_i^{(\tau)} = \left.\frac{\partial E(\theta)}{\partial \theta_i}\right|_{\theta^{(\tau)}}$$

[35%]

(c) Compare the two forms of update rules derived in parts (a) and (b). You should include a discussion of the practical issues and computational costs of the two approaches as the number of parameters in the classifier gets large. [25%]

4 A classifier is to be built for a $K$-class problem. There are $n$, $d$-dimensional, training samples, $\mathbf{x}_1, \dots, \mathbf{x}_n$, with class labels, $y_1, \dots, y_n$. A 1-of-K coding for the class label is used so that each training example, $\mathbf{x}_i$, has a $K$-dimensional vector, $\mathbf{t}_i$, associated with it. Initially a single layer network with a soft-max activation function is used so that the output for node $j$, $\phi_j(\mathbf{x})$, is

$$\phi_j(\mathbf{x}) = \frac{\exp(\mathbf{w}_j'\mathbf{x})}{\sum_{k=1}^K \exp(\mathbf{w}_k'\mathbf{x})}$$

where the parameters of the classifier, $\lambda$, are $\mathbf{w}_1, \dots, \mathbf{w}_K$.

(a) The parameters of the classifier, $\lambda$, are to be trained using the following criterion

$$L(\lambda) = \sum_{k=1}^K \sum_{i=1}^n t_{ik} \log(\phi_k(\mathbf{x}_i))$$

where $t_{ik}$ is element $k$ of vector $\mathbf{t}_i$.

(i) Why is this criterion appropriate for training this form of network? [15%]

(ii) Derive an expression for the derivative of $L(\lambda)$ with respect to $\mathbf{w}_j$. How can this derivative be used to find the model parameters? [30%]

(b) A regularisation term is added to the criterion in part (a). The parameters are now estimated based on the following function

$$F(\lambda) = L(\lambda) - a \sum_{k=1}^K \mathbf{w}_k'\mathbf{w}_k$$

where $a$ is a fixed scalar value.

(i) Discuss why this form of expression may yield an estimate of $\lambda$ that generalises better. How does the value of $a$ influence the estimate of $\lambda$? [15%]

(ii) Derive an expression for the derivative of $F(\lambda)$ with respect to $\mathbf{w}_j$. [15%]

(c) Instead of adding a regularisation term, an additional layer is added to the network between the input and the existing layer. This additional layer has a linear activation function, $\phi(x) = x$. The number of nodes in the additional layer is $b$.

(i) Discuss how the introduction of this additional layer might improve the generalisation of the network, and any constraints on the value of $b$ for this layer to be useful. [15%]

(ii) Discuss how this additional layer will alter the training of the network, if a similar criterion to part (a) is used. [10%]

5 A classifier is required for a 2-class problem. There are a total of $m$ training samples $\mathbf{x}_1$ to $\mathbf{x}_m$ with associated labels $y_1$ to $y_m$ where $y_i \in \{-1, 1\}$.

(a) Initially a linear classifier is to be constructed. Contrast the training criterion used to train a Support Vector Machine (SVM) classifier with the perceptron criterion when the training data are linearly separable. How is the training criterion for the SVM altered for the case when the training data are not linearly separable? [25%]

(b) Discuss how the use of kernel functions may be used to improve the performance of an SVM classifier. What is the general form for an inhomogeneous polynomial kernel-function? [15%]

(c) The training samples are 1-dimensional. The following mapping is proposed from the 1-dimensional *input-space* to the $(N+1)$-dimensional *feature-space*:

$$\Phi(x) = \left[ \begin{array}{ccccc} 1 & \exp(x) & \exp(2x) & \ldots & \exp(Nx) \end{array} \right]'$$

where $x$ is the point in the input-space.

(i) Compare this form of feature-space with the feature-space associated with an inhomogeneous polynomial kernel. [20%]

(ii) Show that the kernel-function, the dot-product of two vectors in the feature-space, between two points $x_i$ and $x_j$ for this mapping may be expressed in the following form

$$k(x_i, x_j) = \frac{b - \exp(a(x_i + x_j))}{b - \exp(x_i + x_j)}$$

What are the values of $a$ and $b$? [25%]

(d) Express the SVM classification rule using the kernel-function in its dual form which is a function of the support vectors. How does the computational cost of classification vary as the number of support vectors, $S$, the number of training samples, $m$, and $N$ change? [15%]

**END OF PAPER**