ENGINEERING TRIPOS PART IIB

Tuesday 26 April 2011 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

mjfg04

1 An M-component, diagonal covariance matrix, Gaussian mixture model (GMM) is trained on a set of d-dimensional feature vectors. The feature vectors were obtained from a series of measurements. The type of measurement equipment was then changed and a small number of new independent observations, $\mathbf{x}_1,\ldots,\mathbf{x}_n$, were measured. The new instrument is known to introduce a fixed, additive, vector offset, $\mathbf{b}$, to the feature vectors compared to the original instrument.

(a) What are the advantages of using a GMM rather than a Gaussian to model the distribution of the original observations? Contrast the use of gradient descent optimisation and expectation-maximisation (EM) for optimising the parameters of a GMM. [30%]

(b) The log-likelihood can be written in terms of the original GMM with M components and parameters θ (comprising the mean vectors $\mu_1,\ldots,\mu_M$, covariance matrices $\Sigma_1,\ldots,\Sigma_M$, and priors $c_1,\ldots,c_M$) and the offset $\mathbf{b}$.

$$\log\left(p(\mathbf{x}_1,\ldots,\mathbf{x}_n|\theta,\mathbf{b})\right) = \sum_{i=1}^{n}\log\left(\sum_{m=1}^{M} c_m\mathcal{N}(\mathbf{x}_i-\mathbf{b};\mu_m,\Sigma_m)\right)$$

All the covariance matrices are diagonal. Show that the gradient descent update rule to find the maximum likelihood estimate of $\mathbf{b}$ has the form at iteration $k+1$

$$b_j^{(k+1)} = b_j^{(k)} + \eta\sum_{i=1}^{n}\sum_{m=1}^{M}\left(a_{mi}^{(k)}\left(\frac{x_{ij}-\mu_{mj}-b_j^{(k)}}{\sigma_{mj}^2}\right)\right)$$

where η is the learning rate, x_{ij} is element j of vector $\mathbf{x}_i$ and σ_{mj}^2 is the j^{th} element on the diagonal of Σ_m. What is the expression for $a_{mi}^{(k)}$? [35%]

(c) Expectation-maximisation is now to be used to estimate $\mathbf{b}$. Starting from the standard auxiliary function for mixture models, show that the EM estimate at iteration $k+1$ is given by

$$b_j^{(k+1)} = \frac{1}{\sum_{i=1}^{n}\sum_{m=1}^{M}\left(a_{mi}^{(k)}/\sigma_{mj}^2\right)}\left(\sum_{i=1}^{n}\sum_{m=1}^{M} a_{mi}^{(k)}\frac{(x_{ij}-\mu_{mj})}{\sigma_{mj}^2}\right)$$

where $a_{mi}^{(k)}$ has the same form as section (b). [35%]

2 A Support Vector Machine (SVM) classifier is to be built for a two 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) Discuss why kernel-functions are often used with SVM classifiers. What is the form for a Gaussian kernel-function and how can it be tuned to a particular task? [20%]

(b) The training samples are 1-dimensional. The following mapping is proposed from the 1-dimensional *input-space* to the $(2N+1)$-dimensional *feature-space*.

$$\Phi(x) = \left[\begin{array}{cccccccc} \frac{1}{\sqrt{2}} & \cos(x) & \cos(2x) & \ldots & \cos(Nx) & \sin(x) & \sin(2x) & \ldots & \sin(Nx) \end{array} \right]'$$

where x is the point in the input-space.

(i) 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

$$k(x_i, x_j) = \frac{\sin(a(x_i - x_j))}{2\sin(b(x_i - x_j))}$$

What are the values of a and b? [30%]

(ii) Express the classification rule for the SVM using the kernel-function and the set of support vectors. How does the computational cost of classification vary as the number of support vectors, S, number of training samples, m, and N change? [15%]

(iii) Contrast this form of feature-space and associated kernel-function with the Gaussian kernel-function. [15%]

(c) The SVM classifier is to be extended to handle classification problems with more than two classes. Discuss how the SVM training and classification might be modified to allow a *single* SVM classifier to perform multi-class classification. [20%]

3 A linear classifier with parameter a of the form

$$y(x) = ax$$

is to be trained for a one-dimensional, two-class, problem. The data for each of the two classes, ω_1 and ω_2, are Gaussian distributed. For class ω_1 the mean is 0 and variance is 1. For class ω_2 the mean is 1 and the variance is 2. The priors for the two classes are known to be equal. There are N training examples equally split between the two classes.

(a) What is the general form of Bayes' decision rule for a two class problem? [10%]

(b) The linear classifier is to be trained using least squares estimation with target values of 0 for class ω_1 and 1 for class ω_2. For N samples this criterion has the form

$$E(a) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i(ax_i)^2 + (1-y_i)(1-ax_i)^2\right)$$

where y_i is 1 if the observation belongs to class ω_1 and 0 if it belongs to class ω_2. A very large number of training examples, N, are available to estimate the classifier parameter. Calculate the optimal value of the classifier parameter, a, using this criterion. [30%]

(c) A threshold of 0.5 on $y(x)$ is used to classify the data. Using the value of a estimated in part (b) calculate the probability of misclassifying a sample in terms of the Gaussian cumulative density function $F(x)$ where

$$F(x) = \int_{-\infty}^{x} \mathcal{N}(z; 0, 1)dz$$

[30%]

(d) What expression is satisfied by a point x that lies on the optimal decision boundary specified by Bayes' decision rule? Using this expression obtain a new estimate of a that will reduce the probability of error compared with the threshold given in part (c).

[30%]

4 A multilayer perceptron is to be trained using a quadratic approximation to the error surface. The set of weights associated with the network are denoted as the vector θ. An iterative procedure is commonly used to update the weight vector 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 τ. The value of the cost function with model parameters θ 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 θ 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 θ 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 sections (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 multilayer perceptron gets large. [25%]

5 Regression is to be performed using either *basis functions* or a *Gaussian process*. There are n, d-dimensional, training observations, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, with associated output values $\mathbf{y} = [y_1, \ldots, y_n]'$. The outputs are related to the observations by $y_i = f(\mathbf{x}_i) + \varepsilon$ where the prediction noise, ε, is Gaussian distributed, $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

(a) For basis function regression, the *systematic prediction*, $f(\mathbf{x})$, has the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i \phi(||\mathbf{x}_i - \mathbf{x}||)$$

The prior for each weight of this regression process is Gaussian distributed with the following form: $p(w_i) = \mathcal{N}(w_i; 0, \sigma_{\mathrm{w}}^2)$.

(i) Derive an expression for the Maximum A-Posteriori (MAP) estimate of the parameters of the regression process, $\mathbf{w} = [w_1, \ldots, w_n]'$, in terms of the training observations and output values. [40%]

(ii) Hence derive an expression for the distribution of the output y for the observation $\mathbf{x}$ using the MAP-estimated regression process parameters. [10%]

(b) For Gaussian process regression, the systematic prediction, $f(\mathbf{x})$, is jointly Gaussian distributed with the training outputs, $\mathbf{y}$. A squared exponential covariance function is to be used to which an additional term is added for the prediction noise ε. The prior mean function is set to 0.

(i) By deriving an expression for the joint distribution of $f(\mathbf{x})$ and the output values $\mathbf{y}$, show that the mean, μ_{f}, of the distribution of the systematic prediction, $f(\mathbf{x})$, can be written in the form

$$\mu_{\mathrm{f}} = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i)$$

Find expressions for $\alpha = [\alpha_1, \ldots, \alpha_n]'$ and $k(\mathbf{x}, \mathbf{x}_i)$. [25%]

(ii) If the variance of the distribution of the systematic prediction is σ_{f}^2, what is the distribution of the prediction of the output y for observation $\mathbf{x}$? [10%]

(c) Compare the two forms of regression described in parts (a) and (b). You should discuss computational cost, storage, and how accurate the regression process is. [15%]

The following equality for vectors may be useful for this question. If **a** and **b** are jointly Gaussian,

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{\mathtt{a}} \\ \mu_{\mathtt{b}} \end{bmatrix}, \begin{bmatrix} \Sigma_{\mathtt{aa}} & \Sigma_{\mathtt{ab}} \\ \Sigma_{\mathtt{ba}} & \Sigma_{\mathtt{bb}} \end{bmatrix} \right)$$

then

$$\mathbf{a}|\mathbf{b} \sim \mathcal{N}(\mu_{\mathtt{a}} + \Sigma_{\mathtt{ab}}\Sigma_{\mathtt{bb}}^{-1}(\mathbf{b} - \mu_{\mathtt{b}}), \Sigma_{\mathtt{aa}} - \Sigma_{\mathtt{ab}}\Sigma_{\mathtt{bb}}^{-1}\Sigma_{\mathtt{ba}})$$

END OF PAPER