Solutions to 4F10 Pattern Processing, 2015

1. *Bayes' Decision rule and generative models*

(a)(i) Bayes' decision rule states

$$\text{Decide } \arg\max_{\omega_j} \left\{ P(\omega_j|\boldsymbol{x}) \right\}$$

which can be expressed for the generative classifiers here as

$$\text{Decide } \arg\max_{\omega_j} \left\{ p(\boldsymbol{x}|\omega_j)P(\omega_j) \right\}$$

[10%]

(a)(ii) A number of points should be discussed

- Generative models use Bayes' decision rule to express the posterior class probability in term of the likelihood and class priors
- Generative models are minimum error classifiers is if there is
  - infinite training data
  - correct models (likelihood and priors)
  - appropriate training algorithm
- Discriminative models directly model the class posteriors.

[20%]

(b)(i) The expression for the probability of error is

$$\begin{aligned} P(\text{error}) &= P(\boldsymbol{x} \in \Omega_2, \omega_1) + P(\boldsymbol{x} \in \Omega_1, \omega_2) \\ &= P(\boldsymbol{x} \in \Omega_2|\omega_1)P(\omega_1) + P(\boldsymbol{x} \in \Omega_1|\omega_2)P(\omega_2) \\ &= \int_{\Omega_2} p(\boldsymbol{x}|\omega_1)P(\omega_1)d\boldsymbol{x} + \int_{\Omega_1} p(\boldsymbol{x}|\omega_2)P(\omega_2)d\boldsymbol{x} \end{aligned}$$

[15%]

(b)(ii) From the inequality given, $a \leq \sqrt{ab}$, if $a \leq b$

$$\int_{\Omega_2} p(\boldsymbol{x}|\omega_1)P(\omega_1)d\boldsymbol{x} \leq \int_{\Omega_2} \sqrt{p(\boldsymbol{x}|\omega_1)P(\omega_1)p(\boldsymbol{x}|\omega_1)P(\omega_1)}d\boldsymbol{x}$$

as by definition in the region where class 2 is labelled

$$p(\boldsymbol{x}|\omega_1)P(\omega_1) \leq p(\boldsymbol{x}|\omega_2)P(\omega_2)$$

A similar expression can be obtained for region Ω_1. Thus

$$P(\text{error}) \leq \int \sqrt{p(\boldsymbol{x}|\omega_1)P(\omega_1)p(\boldsymbol{x}|\omega_2)P(\omega_2)}d\boldsymbol{x}$$

[25%]

(b)(iii) An expression can be obtained based on the inequality in part (b)(ii). The product of two Gaussians is an un-normalised Gaussian. Consider

$$\mathcal{N}(\mathbf{x}; \mu_1, \Sigma)\mathcal{N}(\mathbf{x}; \mu_2, \Sigma) = \\ \frac{1}{(2\pi)^d|\Sigma|} \exp\left(-\frac{1}{2}\left(2\mathbf{x}'\Sigma^{-1}\mathbf{x} - 2(\mu_1 + \mu_2)'\Sigma^{-1}\mathbf{x} + \mu_1'\Sigma^{-1}\mu_1 + \mu_2'\Sigma^{-1}\mu_2 \right) \right)$$

Taking the square-root of this gives

$$\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}'\Sigma^{-1}\mathbf{x} - 2\frac{(\mu_1 + \mu_2)'}{2}\Sigma^{-1}\mathbf{x} + \frac{1}{2}(\mu_1'\Sigma^{-1}\mu_1 + \mu_2'\Sigma^{-1}\mu_2) \right) \right)$$

Integrating a Gaussian yields 1, so

$$P(\text{error}) \leq K \int \mathcal{N}(\mathbf{x}; \frac{(\mu_1 + \mu_2)}{2}, \Sigma) d\mathbf{x} = K$$

where the constant K can be expressed as (not forgetting the prior)

$$K = \frac{1}{2} \exp\left(\frac{1}{8}(\mu_1 + \mu_2)'\Sigma^{-1}(\mu_1 + \mu_2) - \frac{1}{4}\mu_1'\Sigma^{-1}\mu_1 - \frac{1}{4}\mu_2'\Sigma^{-1}\mu_2 \right)$$

[It was also acceptable to find an expression based on the equality in part (b)(i). This yields an expression in terms of cumulative density functions and requires finding the decision boundary.] [30%]

Assessor's Comments: This question examined Bayes' decision rule and the associated probability of errors. Generally a well answered question with candidates showing a good understanding of the probability of error.

2. *Mixtures and Product of Experts*

(a) A mixture of experts (for generative models) has the form

$$p(\mathbf{x}|\boldsymbol{\theta}) \propto \sum_{m=1}^{M} P(\omega_m|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|\omega_m, \theta)$$

The product of experts has the form

$$p(\mathbf{x}|\boldsymbol{\theta}) \propto \prod_{m=1}^{M} p(\mathbf{x}|\omega_m, \theta)$$

[additional weights can be placed on the experts if desired]

Discussion should unclude

- mixture of experts can model model data
- mixtures of experts act as a union (only one expert required to give high likelihood)
- product of experts acts as an intersect (all experts need to be high) [20%]

(b) Need to compute

$$\frac{1}{Z}\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) = \frac{1}{K}\exp\left(\mathbf{x}'\left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)\mathbf{x} - 2\left(\boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\mu}_2'\boldsymbol{\Sigma}_2^{-1}\right)\mathbf{x} + c\right)$$

Equating co-efficients

$$\boldsymbol{\Sigma} = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}$$

and

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right)$$

It is not necessary to compute any other terms. [25%]

(c)(i) The form of the matrix us

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

(c)(ii) This uses the following equality

$$\exp\left(-\frac{1}{2}(\mathbf{A}\mathbf{x} - \boldsymbol{\mu})'(\mathbf{A}\mathbf{x} - \boldsymbol{\mu})\right) = \exp\left(-\frac{1}{2}\left(\mathbf{x}'\mathbf{A}'\mathbf{A}\mathbf{x} - 2\boldsymbol{\mu}'\mathbf{A}\mathbf{x} + \boldsymbol{\mu}'\boldsymbol{\mu}\right)\right)$$

The simply equate co-efficients to get solution.

Using this transformation and noting that variances for each of the experts is 1, and the resulting mean has the form

$$\mathbf{v} = \begin{bmatrix} \mu_a \\ 0 \\ \mu_c \end{bmatrix}$$

This now has the form of the LHS of the expression. Using the RHS shows that the values of **A** and **v** above are the ones combined. Inverse covariance matrix has the form

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}' \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

[30%]

(c)(iii) The inverse covariance matrix has zeros in the bottom-left and top-right hand corners. This implies conditional independence between x_1 and x_3 given x_2. [this is not explicitly discussed in lectures, but can be infered from the nature of the experts]. [10%]

Assessor's Comments: A question examining the attributes of a product of experts model. A well answered question, with candidates showing a good understanding of how to combine experts in a product framework.

3. *Gaussian Mixture Models* (a) Log-likelihood of the training data is

$$\log(p(x_1, \ldots, x_N|\theta)) = \sum_{i=1}^{N} \log \left(\sum_{m=1}^{M} c_m \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \right)$$

[10%]

(b)(i) EM is an iterative approach to estimating the model parameters. Given the current estimates of the model parameters, $\boldsymbol{\theta}$, the new estimates, $\hat{\boldsymbol{\theta}}$, are found using

- Compute component posteriors, $P(\omega_m|x_i, \boldsymbol{\theta})$, using current parameters.
- Using the Auxiliary function, $\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$, compute the new parameters.

[15%]

(b)(ii) Substituting in the expression for the likelihood to the auxiliary function

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \sum_{m=1}^{M} P(\omega_m|x_i, \boldsymbol{\theta}) \log \left(\mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \right)$$

Differentiate this with respect to $\boldsymbol{\mu}_q$ gives

$$\frac{\partial \mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\mu}}_q} = \sum_{i=1}^{n} P(\omega_q|x_i, \boldsymbol{\theta}) \left[\hat{\boldsymbol{\Sigma}}^{-1} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_q) \right]$$

Equating to zero gives

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

[30%]

(c)(i) The sequential estimate can be expressed as

$$\begin{aligned}
\boldsymbol{\mu}_m^{(n)} &= \boldsymbol{\mu}_m^{(n-1)} + \left(\boldsymbol{\mu}_m^{(n)} - \boldsymbol{\mu}_m^{(n-1)} \right) \\
&= \boldsymbol{\mu}_m^{(n-1)} + \frac{b_{n-1}\mathbf{a}_n - b_n \mathbf{a}_{n-1}}{b_n b_{n-1}} \\
&= \boldsymbol{\mu}_m^{(n-1)} + \frac{b_{n-1} P(\omega_m|\mathbf{x}_n, \boldsymbol{\theta})\mathbf{x}_n - P(\omega_m|\mathbf{x}_n, \boldsymbol{\theta})\mathbf{a}_{n-1}}{b_n b_{n-1}} \\
&= \boldsymbol{\mu}_m^{(n-1)} + \frac{P(\omega_m|\mathbf{x}_n, \boldsymbol{\theta})}{b_n} \left(\mathbf{x}_n - \boldsymbol{\mu}_m^{(n-1)} \right)
\end{aligned}$$

where

$$\mathbf{a}_n = \sum_{i=1}^{n} P(\omega_m|\mathbf{x}_i, \boldsymbol{\theta}) \mathbf{x}_i$$

$$b_n = \sum_{i=1}^{n} P(\omega_m|\mathbf{x}_i, \boldsymbol{\theta})$$

thus

$$\eta_m^{(n)} = \frac{P(\omega_m|\mathbf{x}_n, \boldsymbol{\theta})}{\sum_{i=1}^{n} P(\omega_m|\mathbf{x}_i, \boldsymbol{\theta})}$$

[30%]

(c)(ii) The problem with the exact form is that the denominator for $\eta_m^{(n)}$ is a function of all the observations and the current model parameters. This means that every observation must be stored, making the update of little use for a sequential update. This is not the case for the approximate form. The motivation for the approximate form is that at the correct solution (using n observations)

$$c_m = \frac{1}{n}\sum_{i=1}^{n} P(\omega_m|\mathbf{x}_i, \boldsymbol{\theta})$$

[15%]

Assessor's Comments: This questions examined the students' knowledge of expectation maximisation (EM) and how it can be used to derive model parameter estimates. Most candidates showed reasonable knowledge of EM and how it could be applied to the standard mixture model. This was the most popular question.

4. Multi-Layer Perceptrons

(a) Should mention

- the size of the input d and the output K are fixed
- number of hidden layers - theoretically one is sufficient, but often many hidden layers (deep) networks used
- nodes per layer
- generalisation
- node-activation-functions for hdden-layers (sigmoid, tanh, ReLu)
- soft-max activation function on output layer for sum to one constraint (and positive)

[20%]

(b)(i) Description

- $I = n$ $J = K$ (or vice-versa)
- $a_{ij} = t_{ij}$
- $b_{ij} = y_j(\mathbf{x}_i)$ (output layer node value - lecture notes)

[10%]

(b)(ii) The output layer value can be expressed as ($\mathbf{c}$ is the bias term)

$$b_{ij} = \frac{\exp(\mathbf{w}'_j\mathbf{x} + c_j)}{\sum_{k=1}^{K}\exp(\mathbf{w}'_k\mathbf{x} + c_k)}$$

Dropping this expression into E extend $\mathbf{x}$ and $\mathbf{w}_j$ to subsume bias term

$$\begin{aligned} E &= -\sum_{i=1}^{I}\sum_{j=1}^{J} a_{ij}\left[\mathbf{w}'_j\mathbf{x} - \log(\sum_{k=1}^{K}\exp(\mathbf{w}'_k\mathbf{x}))\right] \\ &= -\sum_{i=1}^{I}\left[\left(\sum_{j=1}^{J} a_{ij}\mathbf{w}'_j\mathbf{x}\right) - \log(\sum_{k=1}^{K}\exp(\mathbf{w}'_k\mathbf{x}))\right] \end{aligned}$$

Differentiate wrt $\mathbf{w}_j$ yields

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{w}_j} &= -\sum_{i=1}^{I}\left[a_{ij}\mathbf{x}_i - b_{ij}\mathbf{x}_i\right] \\ &= -\sum_{i=1}^{I}\left(a_{ij} - b_{ij}\right)\mathbf{x}_i \end{aligned}$$

[25%]

(c)(i) Element ij of the Hessian is defined as

$$h_{ij} = \frac{\partial E}{\partial w_i \partial w_j}$$

The Hessian can be used for optimsation with

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta} + \mathbf{H}^{-1}\mathbf{g}$$

where both the gradient and the Hessian are evaluated at the current model parameters $\boldsymbol{\theta}$. [15%]

(c)(ii) To find element of Hessian matrix need if $j \neq k$

$$\begin{aligned} \frac{\partial b_{ij}}{\partial \mathbf{w}_k} &= -\frac{\exp(\mathbf{w}'_j\mathbf{x}_i)\exp(\mathbf{w}'_k\mathbf{x}_i)\mathbf{x}_i}{\left(\sum_{k=1}^{K}\exp(\mathbf{w}'_k\mathbf{x}_i)\right)^2} \\ &= -b_{ij}b_{ik}\mathbf{x}_i \end{aligned}$$

and if equal

$$\begin{aligned}\frac{\partial b_{ij}}{\partial \mathbf{w}_j} &= \frac{\left(\exp(\mathbf{w}_j'\mathbf{x}_i)\sum_{k=1}^{K}\exp(\mathbf{w}_k'\mathbf{x}_i) - \exp(\mathbf{w}_j'\mathbf{x}_i)^2\right)\mathbf{x}_i}{\left(\sum_{k=1}^{K}\exp(\mathbf{w}_k'\mathbf{x}_i)\right)^2} \\ &= (b_{ij} - b_{ij}^2)\mathbf{x}_i\end{aligned}$$

Substituting this back in yields

$$\mathbf{H}_{jk} = \sum_{i=1}^{I} b_{ij}(I_{kj} - b_{ik})\mathbf{x}\mathbf{x}'$$

[20%]

(c)(iii) For each pair of weights the Hessian needs to be computed. The total number of weights in the $K \times N$ where N is the number of nodes in the final hidden layer. Computing the Hessian block for a pair of weights is $\mathcal{O}(n \times N \times N)$ and $\mathcal{O}(K^2)$ pairs of weights. This is clearly highly expensive. [10%]

Assessor's Comments: This question was based on neural network training and was the least popular question. The performance on this question was disappointing, with some candidates showing little knowledge of this form of classifier.

5. *Support Vector Machines and Kernels*

(a)(i) For a general kernel the form is

$$\sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b = 0$$

where α are the Lagrange multipliers for each training sample i. [15%]

(a)(ii) Comments to make:

- large dimensional feature-spaces can be generated with kernels
- SVMs are based on large-margin training, good generalisation for high dimensions
- computational cost is determined by the number of support vectors, where $\alpha_i > 0$

[20%]

(b)(i) for non-zero a this yields a inhomogeneous polynomial kernel. (The value of a influences the weight value, not the space). c determines the order of the space. For a linear kernel $c = 1$. For values $c > 1$ the decision boundaries are non-linear. [15%]

(b)(ii) From the answer to part (a)(i)

$$\sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b = \sum_{i=1}^{m} \alpha_i y_i \sum_{j=1}^{n} k_j(\mathbf{x}_i, \mathbf{x}) + b = 0$$

[10%]

(b)(iii) The kernel operation is the dot product of the vectors in the feature-space. So

$$\begin{aligned}
k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) &= \boldsymbol{\Phi}\left(\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix}\right)' \boldsymbol{\Phi}\left(\begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \end{bmatrix}\right) \\
&= \begin{bmatrix} x_1^{(1)3} \\ \sqrt{3}x_1^{(1)2}x_2^{(1)} \\ \sqrt{3}x_1^{(1)}x_2^{(1)2} \\ x_2^{(1)3} \end{bmatrix}' \begin{bmatrix} x_1^{(2)3} \\ \sqrt{3}x_1^{(2)2}x_2^{(2)} \\ \sqrt{3}x_1^{(2)}x_2^{(2)2} \\ x_2^{(1)3} \end{bmatrix} \\
&= x_1^{(1)3}x_1^{(2)3} + 3x_1^{(1)2}x_2^{(1)}x_1^{(2)2}x_2^{(2)} + 3x_1^{(1)}x_2^{(1)2}x_1^{(2)}x_2^{(2)2} + x_2^{(1)3}x_2^{(2)3} \\
&= (x_1^{(1)}x_1^{(2)} + x_2^{(1)}x_2^{(2)})^3
\end{aligned}$$

[20%]

(b)(iv) The dimensionality of the inhomogeneous polynomial kernel can be expressed as a sum of homogeneous ones (from question). Thus the complete dimensionality is the sum of the dimensionalities of all the individual homogeneous ones. Since the dimensionality for power c is simply $c + 1$ for $d = 2$, then the complete dimensionality is

$$\sum_{c=0}^{c}(c+1) = \frac{1}{2}(c+1)(c+2) = \left(\begin{array}{c} 2+c \\ c \end{array}\right)$$

As d increases so the dimensionality of all the individual polynomial terms increases.

For full marks some additional discussion of the dimensionality of an individual term for d and i is required (note not the complete discussion below)

$$\left(\begin{array}{c} d+i-1 \\ i \end{array}\right)$$

this can then be used a recursion based on

$$\left(\begin{array}{c} d+c \\ c \end{array}\right) = \left(\begin{array}{c} d+c-1 \\ c-1 \end{array}\right)\left(\begin{array}{c} d+c-1 \\ c \end{array}\right)$$

[35%]

Assessor's Comments: This question examined the candidates knowledge of kernels and the resulting feature space. A disappointingly large number of candidates were unable to analyse polynomial kernels, despite being described in the lecture notes.