Solutions to 4F10 Pattern Processing, 2014

1. *Bayes' Decision Rule and Probability of Error*

(a) Bayes' decision rule for a two class problem is

$$\text{Decide}\left\{\begin{array}{ll}\text{Class } \omega_1 & \text{if } P(\omega_1|\boldsymbol{x}) > P(\omega_2|\boldsymbol{x});\\ \text{Class } \omega_2 & \text{Otherwise}\end{array}\right.$$

[10%]

(b) A point that lies on the decision boundary satisfies

$$\log(p(\boldsymbol{x};\boldsymbol{\mu}_1,\boldsymbol{\Sigma})) + \log(P(\omega_1)) = \log(p(\boldsymbol{x};\boldsymbol{\mu}_2,\boldsymbol{\Sigma})) + \log(P(\omega_2))$$

Substituting in the expressions for the covariance and priors yields

$$\mathbf{a}'\boldsymbol{x} - \frac{1}{2}\mathbf{a}'\mathbf{a} = \mathbf{b}'\boldsymbol{x} - \frac{1}{2}\mathbf{b}'\mathbf{b}$$

Yielding the equation of a straight-line

$$(\mathbf{a}-\mathbf{b})'\,\boldsymbol{x} = \frac{1}{2}\left(\mathbf{a}'\mathbf{a} - \mathbf{b}'\mathbf{b}\right)$$

[20%]

(c) **Simplest solution is to draw a sketch indicates the distances in standard deviations. This is acceptable as a solution**.

The decision boundary in (b) defines the two regions. For this form of distribution the posterior will only be a function of the perpendicular distance from the decision boundary. Projecting the distribution for class along this line

$$\mathcal{N}\left(x;(\mathbf{b}-\mathbf{a})'\,\mathbf{a}/K,1\right)$$

where $K^2 = ||\mathbf{b}-\mathbf{a}||^2$. Perpendicular to this direction will just integrate out to 1 for all positions.

The decision boundary is the projection of the point half way between the means onto this line. This projection point is

$$a = (\mathbf{a}+\mathbf{b})'(\mathbf{b}-\mathbf{a})/2K$$

Considering the first element of the probability of error

$$\int_{\mathcal{R}_2} p(\boldsymbol{x}|\omega_1)P(\omega_1)d\boldsymbol{x} = \frac{1}{2}\int_a^\infty \mathcal{N}\left(x;(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)'\,\boldsymbol{\mu}_1/K;1\right)d\boldsymbol{x}$$

Offsetting the mean of the integral to 0 yields the required form

$$\int_{\mathcal{R}_2} p(\boldsymbol{x}|\omega_1)P(\omega_1)d\boldsymbol{x} = \frac{1}{2}\int_a^\infty \mathcal{N}\left(x;0;1\right)d\boldsymbol{x}$$

where a is now

$$\begin{aligned} a &= \frac{1}{2K}(\mathbf{b}-\mathbf{a})'(\mathbf{a}+\mathbf{b}) - \frac{1}{K}(\mathbf{b}-\mathbf{a})'\mathbf{a} \\ &= \frac{1}{2K}(\mathbf{a}-\mathbf{b})'(\mathbf{a}-\mathbf{b}) \\ &= \frac{1}{2}\sqrt{(\mathbf{a}-\mathbf{b})'(\mathbf{a}-\mathbf{b})} \end{aligned}$$

Since the probability of error for the second expression will be the same this is the value of c. By symmetry the required answer is $-a$. [35%]

(d)(i) Averaging the five values will yield a variance of 0.2. This effectively increases the distance of the to points apart by one over the standard deviation, 1/0.447=2.236.

This will simply scale the value of c. [20%]

(d) (ii) This will yield a binomial expression where the overall probability of error is

$$P_{\mathsf{e}}^5 + 5P_{\mathsf{e}}(1-P_{\mathsf{e}}) + 10P_{\mathsf{e}}^2(1-P_{\mathsf{e}})^3$$

This will be a higher probability of error than the scheme in (i), as the scheme in (i) is by definition the minimum possible error rate. [15%]

2. *Mixture Models*

(a) Log-likelihood of the training data is

$$\log(p(x_1, \ldots, x_n|\lambda_1, \ldots, \lambda_M)) = \sum_{i=1}^{n} \log\left(\sum_{m=1}^{M} c_m \lambda_m^{x_i}(1-\lambda_m)^{(1-x_i)}\right)$$

[15%]

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

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

[15%]

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

$$\mathcal{Q}(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = \sum_{i=1}^{n}\sum_{m=1}^{M} P(\omega_m|x_i, \boldsymbol{\lambda})\left(x_i \log(\hat{\lambda}_m) + (1-x_i)\log(1-\hat{\lambda}_m)\right)$$

Differentiate this with respect to $\hat{\lambda}_q$ give

$$\frac{\partial \mathcal{Q}(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})}{\partial \lambda_q} = \sum_{i=1}^{n} P(\omega_q|x_i, \boldsymbol{\lambda})\left[\frac{x_i}{\hat{\lambda}_q} - \frac{(1-x_i)}{(1-\hat{\lambda}_q)}\right]$$

Equating to zero gives

$$(1-\hat{\lambda}_q)\sum_{i=1}^{n} P(\omega_q|x_i, \boldsymbol{\lambda})x_i = \hat{\lambda}_q \sum_{i=1}^{n} P(\omega_q|x_i, \boldsymbol{\lambda})(1-x_i)$$

Rearranging yields

$$\hat{\lambda}_q = \frac{\sum_{i=1}^{n} P(\omega_q|x_i, \boldsymbol{\lambda})x_i}{\sum_{k=1}^{n} P(\omega_j|x_k, \boldsymbol{\lambda})}$$

[30%]

(c)(i) The posterior probability will be exactly the same as the part associated with z will be the same in both the numerator and denominator. [15%]

(c)(ii) Again EM is used. Plugging in the expressions yields

$$Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = K + \sum_{i=1}^{n}\sum_{m=1}^{M} P(\omega_m|x_i, z_i, \boldsymbol{\lambda}, \alpha)\log(p(x_i, z_i|\omega_m, \hat{\lambda}_m, \alpha))$$

This can then be split into the part involving α

$$\begin{aligned} Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) &= K + \sum_{i=1}^{n}\sum_{m=1}^{M} P(\omega_m|x_i, z_i\boldsymbol{\lambda}, \alpha)\left(z_i\log(\alpha) + (1-z)\log(1-\alpha)\right) \\ &= K + \sum_{i=1}^{n}\left(z_i\log(\alpha) + (1-z)\log(1-\alpha)\right) \end{aligned}$$

Thus the update rule is simply

$$\alpha = \frac{1}{n}\sum_{i=1}^{n} z_i$$

[25%]

3. *Training Logistic Regression and the use of the Hessian*

(a)(i) The log-likelihood of the data from class ω_1 can be written as

$$\begin{aligned}\mathcal{L}(\mathbf{b}) &= \sum_{i=1}^{n}\left(y_i\log(P(\omega_1|\mathbf{x}_i,\mathbf{b})) + (1-y_i)\log(P(\omega_2|\mathbf{x}_i,\mathbf{b})\right) \\ &= \sum_{i=1}^{n}\left(y_i\log(P(\omega_1|\mathbf{x}_i,\mathbf{b})) + (1-y_i)\log(1-P(\omega_1|\mathbf{x}_i,\mathbf{b})\right)\end{aligned}$$

Possible decision boundaries are linear passing through the origin. [15%]

(a)(ii) Differentiating

$$\begin{aligned}\frac{\partial}{\partial\mathbf{b}}P(\omega_1|\mathbf{x},\mathbf{b}) &= \frac{\exp(-\mathbf{b}'\mathbf{x})}{(1+\exp(-\mathbf{b}'\mathbf{x}))^2}\mathbf{x} \\ &= P(\omega_1|\mathbf{b},\mathbf{x})(1-P(\omega_1|\mathbf{b},\mathbf{x}))\mathbf{x}\end{aligned}$$

Thus

$$\begin{aligned}\frac{\partial}{\partial\mathbf{b}}\mathcal{L}(\mathbf{b}) &= \sum_{i=1}^{n}\mathbf{x}_i\left(y_i(1-P(\omega_1|\mathbf{b},\mathbf{x}_i)) - (1-y_i)P(\omega_1|\mathbf{b},\mathbf{x}_i)\right) \\ &= \sum_{i=1}^{n}\mathbf{x}_i\left(y_i - P(\omega_1|\mathbf{b},\mathbf{x}_i)\right)\end{aligned}$$

This can be used in a gradient style approach where

$$\mathbf{b}^{(k+1)} = \mathbf{b}^{(k)} + \eta\left.\frac{\partial}{\partial\mathbf{b}}\mathcal{L}(\mathbf{b})\right|_{\mathbf{b}^{(k)}}$$

[25%]

(b)(i) Letting

$$\boldsymbol{\Delta}\mathbf{b}^{(\tau)} = (\mathbf{b}-\mathbf{b}^{(\tau)})$$

gives the following differential expression

$$\frac{\partial}{\partial\boldsymbol{\Delta}\mathbf{b}^{(\tau)}}E(\mathbf{b} = \mathbf{f} + \mathbf{A}\boldsymbol{\Delta}\mathbf{b}^{(\tau)})$$

Equating this to zero gives

$$\boldsymbol{\Delta}\mathbf{b}^{(\tau)} = -\mathbf{A}^{-1}\mathbf{f}$$

Thus the value is given by

$$\hat{\mathbf{b}} = \mathbf{b}^{(\tau)} - \mathbf{A}^{-1}\mathbf{f}$$

[20%]

(b)(ii) Rather than maximising the likelihood it is possible to minimise the negative likelihood. So

$$E(\mathbf{b}) = -\mathcal{L}(\mathbf{b})$$

[10%]

(b)(iii) The values are

$$\begin{aligned}\mathbf{f} &= -\left.\boldsymbol{\nabla}\mathcal{L}(\mathbf{b})\right|_{\mathbf{b}^{(\tau)}} \\ \mathbf{A} &= -\left.\boldsymbol{\nabla}^2\mathcal{L}(\mathbf{b})\right|_{\mathbf{b}^{(\tau)}}\end{aligned}$$

the gradient and the Hessian respectively at the current model parameters.

Thus the value of $\mathbf{f}$ is can be taken from part (a)(ii).

$$\mathbf{f} = -\sum_{i=1}^{n}\mathbf{x}_i\left(y_i - P(\omega_1|\mathbf{b}^{(\tau)}, \mathbf{x}_i)\right)$$

Element j, k of the $\mathbf{A}$ is

$$a_{jk} = \frac{\partial^2}{\partial b_j \partial b_k}\mathcal{L}(\mathbf{b})$$

Using the above expression

$$-\frac{\partial}{\partial b_j}\left(\sum_{i=1}^{n}\left(y_i - P(\omega_1|\mathbf{b}, \mathbf{x}_i)\right) x_{ik}\right) = \sum_{i=1}^{n} P(\omega_1|\mathbf{b}, \mathbf{x}_i)(1 - P(\omega_1|\mathbf{b}, \mathbf{x}_i))x_{ij}x_{ik}$$

Evaluating this at the current estimate yields

$$a_{jk} = \sum_{i=1}^{n} P(\omega_1|\mathbf{b}^{(\tau)}, \mathbf{x}_i)(1 - P(\omega_1|\mathbf{b}^{(\tau)}, \mathbf{x}_i))x_{ij}x_{ik}$$

[30%]

4. *Product of Gaussian Experts*

This question is similar to an examples paper question, so the students should be aware of this form of product of experts

(a) It is possible to express $\mathbf{A}$ as

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

These expressions all match to experts. As the length of the observation varies, the size of $\mathbf{A}$ changes, but the parameters (and number of experts) remains the same. [20%]

(b) The normalisation term must satisfy

$$Z = \int \mathcal{N}(\mathbf{A}\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x}$$

[10%]

(c) The transformed data allows each of the experts to be used individually.Thus (diag() yields a matrix with the vector as the leading diagonal)

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_1 \\ \mu_2 \\ \mu_1 \\ \mu_2 \end{bmatrix}; \quad \boldsymbol{\Sigma} = \text{diag}\left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right)$$

[20%]

(d) Comparing the expansion to a standard Gaussian yields

$$\begin{aligned} \overline{\boldsymbol{\Sigma}}^{-1} &= \mathbf{A}'\mathbf{A} \\ \overline{\boldsymbol{\mu}} &= \overline{\boldsymbol{\Sigma}}\mathbf{A}'\boldsymbol{\mu} \end{aligned}$$

Expanding this expression out

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

From the equality at the start of the question (and noting $\boldsymbol{\Sigma} = \mathbf{I}$)

$$p(\mathbf{x}) \;=\; \frac{1}{Z}\mathcal{N}(\mathbf{A}\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$
\begin{aligned}
&= \frac{1}{Z(2\pi)^3}\exp\left(-\frac{1}{2}(\mathbf{A}\mathbf{x}-\boldsymbol{\mu})'(\mathbf{A}\mathbf{x}-\boldsymbol{\mu})\right) \\
&= \frac{1}{Z(2\pi)^3}\exp\left(-\frac{1}{2}(\mathbf{x}-\overline{\boldsymbol{\mu}})'\overline{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\overline{\boldsymbol{\mu}})-\frac{1}{2}\boldsymbol{\mu}'\boldsymbol{\mu}+\frac{1}{2}\overline{\boldsymbol{\mu}}'\overline{\boldsymbol{\Sigma}}^{-1}\overline{\boldsymbol{\mu}}\right) \\
&= \frac{(2\pi)^{3/2}|\overline{\boldsymbol{\Sigma}}|^{1/2}}{Z(2\pi)^3}\mathcal{N}(\mathbf{x};\overline{\boldsymbol{\mu}},\overline{\boldsymbol{\Sigma}})\exp\left(-\frac{1}{2}\boldsymbol{\mu}'\boldsymbol{\mu}+\frac{1}{2}\overline{\boldsymbol{\mu}}'\overline{\boldsymbol{\Sigma}}^{-1}\overline{\boldsymbol{\mu}}\right)
\end{aligned}
$$

The normalisation term must yield 1, hence

$$
Z = \frac{|\overline{\boldsymbol{\Sigma}}|^{1/2}}{(2\pi)^{3/2}}\exp\left(-\frac{1}{2}\boldsymbol{\mu}'\boldsymbol{\mu}+\frac{1}{2}\overline{\boldsymbol{\mu}}'\overline{\boldsymbol{\Sigma}}^{-1}\overline{\boldsymbol{\mu}}\right)
$$

Thus the normalisation term can be written on the form given. [35%]

(e) Training Gaussian experts independently is trivial, only necessary to compute means and covariance matrices. If computed in a combined fashion, by optimising the log-likelihood. This still has closed form solutions for the means (not expected them to know this), but is more complicated as the normalisation term is a function of the experts. The independent training is sub-optimal, as the exercise is to train the maximum likelihood combined scheme. [15%]

5. *Non-Parametric Classification - Parzen Window*

(a) Parzen window density estimates make no assumptions about the form of the density. In contrast a Gaussian distribution assumes that the form of the density is known, so only the mean and variance are required. The Parzen window requires the storage of all points and a window function computed for each of those points. In contrast the Gaussian needs only the mean and covariance matrix to be computed. The evaluation of the PDF is then very quick, depending on the dimensionality of the training data, not the number of training samples. [25%]

(b)(i) The value of h determines the smoothness of the probability density estimate. The value of h should vary inversely to the number of points a typical form is

$$h_n = \frac{h}{\sqrt{n}}$$

[15%]

(b)(ii) The window function is valid PDF so

$$\int_{\mathcal{R}^d} \phi(\mathbf{x})d\mathbf{x}; \quad \phi(\mathbf{x}) \geq 0$$

By simple analogy with the hypercube (or consider scaling each dimension)

$$\int_{\mathcal{R}^d} \phi(\frac{\mathbf{x}}{h})d\mathbf{x} = h^d$$

Hence

$$\begin{aligned}
\int_{\mathcal{R}^d} \tilde{p}(\mathbf{x})d\mathbf{x} &= \int_{\mathcal{R}^d} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h^d}\phi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)d\mathbf{x} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h^d}\int_{\mathcal{R}^d}\phi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)d\mathbf{x} \\
&= 1
\end{aligned}$$

[15%]

(c)(i) The form of Gaussian window function and the first order Taylor series expansion is

$$\begin{aligned}
\phi\left(\frac{x-x_i}{h}\right) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-x_i)^2}{2h^2}\right) \\
&\approx \frac{1}{\sqrt{2\pi}}\left(1-\frac{(x-x_i)^2}{2h^2}\right)
\end{aligned}$$

The approximate Parzen window is then

$$\begin{aligned}
\tilde{p}(x) &\approx \frac{1}{hn}\sum_{i=1}^{n}\left(\frac{1}{\sqrt{2\pi}}\left(1-\frac{(x-x_i)^2}{2h^2}\right)\right) \\
&= \frac{1}{hn\sqrt{2\pi}}\sum_{i=1}^{n}\left(1-\frac{x^2}{2h^2}+\frac{xx_i}{h^2}-\frac{x_i^2}{2h^2}\right)
\end{aligned}$$

[30%]

(c)(ii) It is only necessary to store the values of b_0, b_1 and b_2. The Parzen window approximation is then simply computed as calculating the weighted sum for the Taylor series. [15%]