Solutions to 4F10 Pattern Processing, 2005

1. *Mixture Models and ML training*

   (a) Z must satisfy

$$Z = \int \exp\left(\boldsymbol{\alpha}'\mathbf{f}(x)\right) dx$$

   this ensures that the PDF integrates to 1.

   (b) If $\mathbf{f}(x)$ has the form given then considering only the exponential terms

$$\alpha_1 x + \alpha_2 x^2 = -\frac{x^2}{2\sigma^2} + \frac{x\mu}{\sigma^2}$$

   Hence

$$\begin{aligned} \alpha_1 &= \frac{\mu}{\sigma^2} \\ \alpha_2 &= -\frac{1}{2\sigma^2} \end{aligned}$$

   The value of Z will consist of bothe the standard normalisation term and the final quadratic term from the Gaussian, hence

$$\frac{1}{Z} = \frac{\exp(-\mu^2/2\sigma^2)}{\sqrt{2\pi\sigma^2}}$$

   (c)(i) The expression for the log-likelihood is

$$l(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \log\left(\sum_{m=1}^{M} c_m \frac{1}{Z_m} \exp\left(\boldsymbol{\alpha}_m'\mathbf{f}(x_i)\right)\right)$$

   (c)(ii) Using the approach described in lectures the differential can be expressed as

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\alpha}_m} l(\boldsymbol{\alpha}) &= \sum_{i=1}^{N} P(m|x_i) \frac{\partial}{\partial \boldsymbol{\alpha}_m} \left(\log(c_m) - \log(Z_m) + \boldsymbol{\alpha}_m'\mathbf{f}(x_i)\right) \\ &= \sum_{i=1}^{N} P(m|x_i) \left[-\frac{1}{Z_m}\frac{\partial}{\partial \boldsymbol{\alpha}_m} Z_m + \mathbf{f}(x_i)\right] \end{aligned}$$

   (c)(iii) The auxillary function for the general mixture model may be written as

$$\mathcal{Q}(\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}) = \sum_{m=1}^{M}\sum_{i=1}^{N} P(m|x_i) \left[\log\left(c_m \frac{1}{Z_m}\right) + \boldsymbol{\alpha}_m'\mathbf{f}(x_i)\right]$$

   The only term that is actually dependent in the observations is the second term (from part (c)(ii)). Hence the statistics required are for each component

$$\sum_{i=1}^{N} P(m|x_i); \quad \sum_{i=1}^{N} P(m|x_i)\mathbf{f}(x_i)$$

2. *Bayes Decision Rule and Linear Classifier*

(a) Bayes decision rule states that you should pick the class with the greatest posterior

$$\frac{P(\omega_1|x)}{P(\omega_2|x)} \begin{matrix} \omega_1 \\ > \\ < \\ \omega_2 \end{matrix} 1$$

(b) The least squares criterion that must be minimised is (ignoring the priors)

$$E(a) = \frac{1}{2}\int (ax)^2 p(x|\omega_1)dx + \frac{1}{2}\int (ax-1)^2 p(x|\omega_2)dx$$

Differentiate this with respect to the a yields

$$\begin{aligned} \frac{\partial}{\partial a}E(a) &= \int ax^2 p(x|\omega_1)dx + \int (ax-1)xp(x|\omega_2)dx \\ &= a\left(\sigma_1^2+\mu_1^2\right) + a\left(\sigma_2^2+\mu_2^2\right) - \mu_2 \end{aligned}$$

The statistics for each of these is given in the question, so we get

$$a + 6a - 2 = 0$$

Hence the value for a is $2/7$

(c) The threshold is set at 0.5. The value of x that will correspond to this is $x_T = 7/4$. The probability of error is then given by

$$\begin{aligned} P(\text{error}) &= \frac{1}{2}\int_{x_T}^{\infty} \mathcal{N}(z;0,1)dz + \frac{1}{2}\int_{-\infty}^{x_T} \mathcal{N}(z;2,2)dz \\ &= \frac{1}{2}\left((1-F(7/4)) + F(-1/(4\sqrt{2}))\right) \end{aligned}$$

(d) A point lying on the decision boundary will satisfy

$$-\log(\sqrt{2\pi}) - \frac{x^2}{2} = -\log(\sqrt{4\pi}) - \frac{(x-2)^2}{4}$$

This may be written as

$$x^2 + 4x - (4 + 4\log(\sqrt{2})) = 0$$

Solving this expression gives

$$x = -2 + / - 2\sqrt{2 + \log(\sqrt{2})}$$

Solving this yields a value of $x = 1.06$ (a simple sketch illustrates that this is the appropriate root to take). To get the new value of a we need $1.06a = 0.5$ hence $a = 0.47$.

3. *Neural Network Training*

(a) The standard form of gradient descent would set

$$\Delta\boldsymbol{\theta}^{(\tau)} = -\eta \, \nabla E(\boldsymbol{\theta})|_{\boldsymbol{\theta}^{(\tau)}}$$

The problem with this is selecting the value of the learning rate η. Too small yields slow convergence, too large may yield instability.

(b)(i) The values are

$$\begin{aligned} \mathbf{b} &= \nabla E(\boldsymbol{\theta})|_{\boldsymbol{\theta}^{(\tau)}} \\ \mathbf{A} &= \nabla^2 E(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}^{(\tau)}} \end{aligned}$$

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

(b)(ii) Letting

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

gives the following differential expression

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

Equating this to zero gives

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

Thus the value is given by

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

(b)(iii) this approach requires the calculation of the Hessian. For large numbers of parameters this may not be practical (it's a square matrix). It may also not be of full rank hence there can be issues with the inversion. It can also head off towards a maximum as well.

(c) Writing out the quadratic form for a single dimension yields (ignoring the subscript i)

$$E(\theta) \approx E(\theta^{(\tau)}) + \Delta\theta b + \Delta\theta^2 a/2$$

At the new update value require that

$$\begin{aligned} g^{(\tau+1)} &= 0 \\ g^{(\tau)} &= b \\ g^{(\tau-1)} &= b - \Delta\theta^{(\tau-1)} a \end{aligned}$$

From part b the update is given by $-b/a$. Solving these three equations yields the update rule.

4. *Support Vector Machines*

(a) Both solutions will perfectly classify the training data. However the perceptron algorithm will select any solution that classifies the data and then stops. The SVM has a unique solution that maximises the margin.

(b)(i) The constraints are given above for every point,

$$y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) \geq 1$$

for all $i = 1, \ldots, N$.

(b)(ii) (This whole section is basically in the lecture notes) For a linear classifier, the shortest distance from the origin to the decision hyperplane is

$$d = \frac{|b|}{||\mathbf{w}||}$$

There are two margins to consider the x class (+) and the o class (-). These will be labelled d_+ and d_- respectively. For linear classifiers $\mathbf{w}$ and b can be scaled arbitrarily without affecting discrimination. To avoid this problem constants are introduced to fix the scaling. Typically

$$\begin{aligned} \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geq 1, \quad & \text{for } y_i = +1 \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b \leq -1, \quad & \text{for } y_i = -1 \end{aligned}$$

(The use of 1 in the constraint is an arbitrary choice. Any positive real number will do, but must be reflected in all subsequent parts) The respective distances to the origin for each of the margins are

$$\begin{aligned} d_+ &= \frac{|1 - b|}{||\mathbf{w}||} \\ d_- &= \frac{|-1 - b|}{||\mathbf{w}||} \end{aligned}$$

The margin is therefore

$$\frac{2}{||\mathbf{w}||}$$

The value that is commonly optimised is therefore twice the margin squared.

(b)(iii) The *Lagrangian* is

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i \left(y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) - 1 \right)$$

At the maximum margin solution we know that

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0; \quad \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{0};$$

This leads to

$$-\sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = \mathbf{0}$$

and

$$\alpha_i \left(\left(y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) - 1 \right) \right) = 0$$

In addition it must satisfy the constraint from part (b)(i) and

$$\alpha_i \geq 0$$

(b)(iv) Selecting a value of i for which α_i is non-zero will simply yield the value of b. A more accurate estimate is found from averaging all such values.

(c) The use of the Gaussian kernel means that it will not in general be possible to explicitly build a decision boundary in that space. It is therefore necessary to use the kernelised version of the classifier in each case. Hence

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

where

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2} \right)$$

For the SVM the Lagrange multipliers are directly used. For the perceptron algorithm α_i is based on the number of times that the point is misclassified.

5. *Classification and Regression Trees*

(a)(i) The general attributes that should be satisfied by a node impurity function, $\phi()$, are

- $\phi()$ is a maximum when $P(\omega_i) = 1/K$ for all i
- $\phi()$ is at a minimum when $P(\omega_i) = 1$ and $P(\omega_j) = 0, j \neq i$.
- It is symmetric function (i.e. the order of the class probabilities doesn't matter).

(a)(ii) $P(\omega_i)$ should be calculated as the fraction fo the observations belonging to class ω_i associated with that node.

The Gini impurity measure may be written as

$$\begin{aligned}\sum_{i\neq j} P(\omega_i)P(\omega_j) &= \sum_i P(\omega_i)\sum_{j\neq i} P(\omega_j) \\ &= \sum_i^K P(\omega_i)(1 - P(\omega_i)) \\ &= 1 - \sum_{i=1}^K (P(\omega_i))^2\end{aligned}$$

This function satisfies all the attributes. Using the second form of the Gini impurity measure

- The function is a maximum when all values are equal (various acceptable proof including

  $$(x + \delta)^2 + (x - \delta)^2 = 2x^2 + \delta^2$$

  This is greater than $2x^2$ so the impurity measure will be less.)
- The value is at a minimum when $P(\omega_i) = 1$ (all other classes zero).
- By inspection it will be symmetric from the second form.

(b)(i) The two possible splits are for attribute one and attribute 2

- Attribute 1: 1 2/5 ω_1, 0 2/3 ω_1
- Attribute 2: 1 1/5 ω_1, 0 3/3 ω_1

Attribute 2 is clearly better as both splits are more pure. The actual calculation of the change in impurity function is

$$1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 - \frac{5}{8}\left(1 - \left(\frac{1}{5}\right)^2 - \left(\frac{4}{5}\right)^2\right) - \frac{3}{8}\left(1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2\right) = \frac{1}{2} - \frac{1}{5} = \frac{3}{10}$$

(b)(ii) Two of the feature vectors are identical. These cannot be split using this data. All other symbols may be perfectly classified. Thus the lowest impurity measure is given by

$$\frac{1}{4}\left(1-\left(\frac{1}{2}\right)^2-\left(\frac{1}{2}\right)^2\right)=\frac{1}{8}$$

as all other nodes will be correct.

(c) Since misclassification costs are now considered the appropriate criterion would be the misclassification criterion

$$1-\max_i\{P(\omega_i)\}$$

The cost function would then be altered so that the cost is twice this when the maximum is class ω_1.