CM311: Model Answers

Adam Prügel-Bennett and Steve Gunn

9th November 2001

Question 1

a) **What is the difference between supervised and unsupervised learning? Give examples of each.**

(2 marks)

Supervised learning requires labelled examples—we know what we want to learn. An example of supervised learning would be to recognise hand written characters using a SVM. Unsupervised learning is used to find unknown structures in data. An example would be using PCA to find the directions of maximal variation in preprocessing images.

b) **Why is building a multi-layer network using linear perceptrons futile?**

(2 marks)

A multi-layer network using linear perceptrons cannot be more powerful than a single perceptron. This is because the function performed by such a network can always be expanded into terms that are linear in the input variables. The most general such expression is given by the linear perceptron.

c) **There are three dice on the table. Two of them are normal dice, while the other dice has 6 on two faces and 1, 2, 3 and 5 on the other face. A dice is chosen at random and is thrown 10 times. On three occasions the top face is a 6. What is the probability that the dice chosen is the dishonest dice? Show all your workings.**

(5 marks)

We can use Bayes' rule

$$P(\text{dishonest}|\text{data}) = \frac{P(\text{data}|\text{dishonest})\, P(\text{dishonest})}{P(\text{data}|\text{dishonest})\, P(\text{dishonest}) + P(\text{data}|\text{honest})\, P(\text{honest})}$$

Now $P(\text{dishonest}) = 1/3$ and $P(\text{honest}) = 2/3$ and

$$\begin{aligned} P(\text{data}|\text{dishonest}) &= \binom{3}{10}\left(\frac{1}{3}\right)^3\left(\frac{2}{3}\right)^7 \\ P(\text{data}|\text{honest}) &= \binom{3}{10}\left(\frac{1}{6}\right)^3\left(\frac{5}{6}\right)^7 \end{aligned}$$

Thus

$$\begin{aligned} P(\text{dishonest}|\text{data}) &= \frac{1}{1+\frac{P(\text{data}|\text{honest})\,P(\text{honest})}{P(\text{data}|\text{dishonest})\,P(\text{dishonest})}} \\ &= \frac{1}{1+2\left(\frac{1}{2}\right)^3\left(\frac{5}{4}\right)^7} = 0.29549 \end{aligned}$$

d) **What problems arise when training a multi-layer perceptron? In what sense is a support vector machine easier to train?**

(3 marks)

Multi-layer perceptrons are difficult to train because they are usually very high dimensional and can have many local minima. In addition the symmetry of the search space can result in there being very little gradient information. In contrast the support vector machine has a single quadratic minima and can be solved to optimality relatively efficiently.

e) **Describe the difference between learning and generalisation error. Explain why minimising one may not minimise the other.**

(3 marks)

Learning error is the error on the training data. While generalisation is the error on unseen data. Minimising learning error can lead to over fitting the data giving poor generalisation. To minimise the learning error we would want to use a learning machine with many degrees of freedom, but such a machine is unlikely to generalise well.

f) **Suggest how you might encode the following attributes**

   i) **sex (male or female)**

   ii) **final educational qualifications (none, GCSE, A-levels, degree)**

   iii) **marital status (single, married, separated, widowed).**

Give a short explanation of your decision.

(5 marks)

Sex could be encoded as a single input taking the value of one (female) and zero (male). This minimizes the number of inputs and explicitly encodes the mutual exclusion of the two possibilities. The educational qualifications could be put on a graded scale none=0, GCSE=1, A-levels=2, degree=3. This would again reduce the number of inputs and would codify the fact that the data has a natural ordering. The marital status has no-natural ordering and so could be encoded as four binary inputs. This would allow the network to deduce any correlations with the different categories.

Question 2

- **Derive a matrix equation for the coefficients of the cubic polynomial**

$$y = w_1 + w_2 x + w_3 x^2 + w_4 x^3$$

which goes through the points $(x, y) = (0, 1), (1, 2), (2, 1), (3, 2)$**.**

(5 marks)

We can write the four similtaneous equations the coefficience must satisfy in a single matrix equation

$$\mathbf{M}\boldsymbol{w} = \boldsymbol{y}$$

where

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{pmatrix}$$

and $\boldsymbol{y} = (1, 2, 1, 2)^{\mathsf{T}}$. The vector of coefficents are then given by $\boldsymbol{w} = \mathbf{M}^{-1}\boldsymbol{y}$

- **Write down the mean squared error for the regression of the above polynomial with respect to an arbitrary set of points $\{(x_i, y_i)\}_{i=1}^{P}$ in terms of the matrix**

$$X = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_P \\ x_1^2 & x_2^2 & \cdots & x_P^2 \\ x_1^3 & x_2^3 & \cdots & x_P^3 \end{pmatrix}$$

the vector $\boldsymbol{y} = (y_1, y_2, \cdots, y_P)^{\mathsf{T}}$ and the weight vector $\boldsymbol{w} = (w_1, w_2, w_3, w_4)^{\mathsf{T}}$.

(5 marks)

Define the error vector

$$\boldsymbol{\epsilon} = \mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \boldsymbol{y}$$

then the mean square error is given by

$$E = \frac{1}{P}|\boldsymbol{\epsilon}|^2 = \frac{1}{P}|\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \boldsymbol{y}|^2$$

- **Derive the least square solution for the weights.**

(10 marks)

The mean squared error can be written as

$$\begin{aligned} E &= \frac{1}{P}|\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \boldsymbol{y}|^2 \\ &= \frac{1}{P}\left(\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \boldsymbol{y}\right)^{\mathsf{T}}\left(\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \boldsymbol{y}\right) \\ &= \frac{1}{P}\left(\boldsymbol{w}^{\mathsf{T}}\mathbf{X} - \boldsymbol{y}^{\mathsf{T}}\right)\left(\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \boldsymbol{y}\right) \\ &= \frac{1}{P}\left(\boldsymbol{w}^{\mathsf{T}}\mathbf{X}\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \boldsymbol{y}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \boldsymbol{w}^{\mathsf{T}}\mathbf{X}\,\boldsymbol{y} - \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y}\right) \\ &= \frac{1}{P}\left(\boldsymbol{w}^{\mathsf{T}}\mathbf{X}\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - 2\boldsymbol{y}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y}\right) \end{aligned}$$

The minimum is given by $\boldsymbol{\nabla}E(\boldsymbol{w}) = 0$ so that

$$\begin{aligned} \boldsymbol{\nabla}E(\boldsymbol{w}) &= \frac{1}{P}\boldsymbol{\nabla}\left(\boldsymbol{w}^{\mathsf{T}}\mathbf{X}\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - 2\boldsymbol{w}^{\mathsf{T}}\mathbf{X}\boldsymbol{y} - \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y}\right) \\ &= \frac{2}{P}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \mathbf{X}\boldsymbol{y}\right) = 0 \end{aligned}$$

or

$$\boldsymbol{w} = \left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}\mathbf{X}\boldsymbol{y} = \mathbf{X}^{+}\boldsymbol{y}$$

where $\mathbf{X}^{+}$ is the pseudo inverse.

- **Describe what happens as the degree of the polynomial used in regression is increased.**

(5 marks)

Low degree polynomials can describe only simple functions. If we have a complicated function then we will arrive at a poor model irrespective of the amount of data we have available. Increasing the degree of the polynomial will decrease the learning error. However, if we fit a high degree polynomial we are liable to over-fit the data. That is, although the model can fit the data well it usually gives poor generalisation performance. The best polynomial to fit will depend on the true model and the amount of data. Usually the degree of the polynomial should be much lower than the number of data-points.

Question 3

a) **Write down the definition of the covariance matrix.**

(3 marks)

The covariance matrix, $\mathbf{C}$, is given by

$$\mathbf{C} = \frac{1}{P-1} \sum_{p=1}^{P} (\boldsymbol{x}_p - \boldsymbol{\mu})(\boldsymbol{x}_p - \boldsymbol{\mu})^{\mathsf{T}}$$

where $\boldsymbol{x}_p$ is the p^{th} input and $\boldsymbol{\mu}$ is the average of the pattern vectors, i.e.

$$\boldsymbol{\mu} = \frac{1}{P} \sum_{p=1}^{P} \boldsymbol{x}_p$$

Alternatively, we could write the ij^{th} component of $\mathbf{C}$ as

$$C_{ij} = \frac{1}{P-1} \sum_{p=1}^{P} (x_{i,p} - \mu_i)(x_{j,p} - \mu_j)$$

b) **Show that the covariance matrix is positive semi-definite (i.e. the eigenvalues are non-negative).**

(3 marks)

The covariance matrix C can be thought as equal to

$$\mathsf{C} = \frac{1}{P-1} \mathsf{X}\mathsf{X}^\mathsf{T}$$

where

$$\mathsf{X} = (\boldsymbol{x}_1 - \boldsymbol{\mu} | \boldsymbol{x}_2 - \boldsymbol{\mu} | \cdots | \boldsymbol{x}_P - \boldsymbol{\mu})$$

and

$$\boldsymbol{\mu} = \frac{1}{P} \sum_{p=1}^{P} \boldsymbol{x}_p.$$

Thus for any vector $\boldsymbol{a}$

$$\boldsymbol{a}^\mathsf{T} \mathsf{C} \boldsymbol{a} = \boldsymbol{a}^\mathsf{T} \mathsf{X}\mathsf{X}^\mathsf{T} \boldsymbol{a} = |\boldsymbol{b}|^2 \geq 0$$

where $\boldsymbol{b} = \mathsf{X}^\mathsf{T} \boldsymbol{a}$. But if $\boldsymbol{v}$ is an eigenvector then

$$\boldsymbol{v}^\mathsf{T} \mathsf{C} \boldsymbol{v} = \lambda \boldsymbol{v}^\mathsf{T} \boldsymbol{v} = \lambda$$

however, we have shown that the left-hand side is non-negative.

c) **Describe how to perform principal component analysis.**

(4 marks)

To obtain the principal components we form the covariance matrix and perform an eigenvector decomposition on it. We retain the leading eigenvectors only. We form a new set of inputs by taking the inner product of the original data with the eigenvectors that we have retained.

d) **Explain why principal components can be useful in data analysis.**

(5 marks)

The principal components give the directions in input space where the data varies most rapidly. PCA can substantially reduce the dimensions of the inputs with only a small loss of information. This can speed up many algorithms and make some algorithms feasible where they may not be so otherwise. In addition, performing PCA as a preprocessing step before doing classification or regression can improve the generalisation performance as it reduces the complexity of the model through reducing the number of inputs.

e) **Describe the K-means clustering algorithm. How could this be used in training a radial basis function neural network.**

(10 marks)

In the K-means clustering algorithm we compute the centres (means) of K clusters iteratively using some training data $\{\boldsymbol{x}_p\}_{p=1}^{P}$. The K-means clustering algorithm goes as follows.

- Randomly partition the data points into K sets $\{\mathcal{C}_i\}_{i=1}^{K}$
- Do until there are no more changes
  - Calculate the means $\boldsymbol{\mu}_i$ of each set

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{C}_i|} \sum_{p \in \mathcal{C}_i} \boldsymbol{x}_p$$

  - For each data point $\boldsymbol{x}_p$ calculate the distance $|\boldsymbol{x}_p - \boldsymbol{\mu}_i$ to the centre to each cluster. Assign the data-points to the cluster with the smallest distance.

K-means clustering can be used to assign the position (centres) of the radial basis functions. For example, the radial basis function might be of the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{K} w_i \, g\left(\frac{\boldsymbol{x} - \boldsymbol{\mu}_i}{\sigma_i} \right)$$

where $\boldsymbol{\mu}_i$ would be the centres computed using K means, and g is a radial basis function. The variances σ_i (or more generally the covariance matrices) could be computed from the clusters found from the radial basis functions.

Question 4

a) **Explain what is meant by the term over-parameterisation. State how it is removed from the hyperplane, $\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}+b=0$, in the linear Support Vector Machine formulation to produce a canonical hyperplane.**

(3 marks)

Over-parameterisation refers to case when the parameters in an equation are not independent, and hence two different sets of parameters can describe an identical solution. The over-parameterisation in $\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b = 0$ is removed by adding the constraint $\min_i \left|\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}_i + b\right| = 1$ where $\boldsymbol{x}_i$ are the input space coordinates of the training examples.

b) **State the condition for separability of the two-class data-set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n, \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$ with this canonical hyperplane.**

(3 marks)

$y_i \left[\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b\right] \geq 1, \quad i = 1, \ldots, n.$

c) **Describe the maximum margin principle and show that the resulting optimisation problem is given by the Lagrangian,**

$$\Phi(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^n \alpha_i \left(y_i \left[\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b\right] - 1\right), \quad \alpha_i \geq 0.$$

(8 marks)

The maximum margin principle states: "The set of vectors, $\boldsymbol{x}_i$, is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vector to the hyperplane is maximal."

The distance $d(\boldsymbol{w}, b; \boldsymbol{x})$ of a point $\boldsymbol{x}$ from the hyperplane $(\boldsymbol{w}, b)$ is,

$$d(\boldsymbol{w}, b; \boldsymbol{x}) = \frac{\left|\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b\right|}{\|\boldsymbol{w}\|}.$$

The optimal hyperplane is given by maximising the margin, ρ, subject to the separability constraints of b). The margin is given by,

$$\begin{aligned}
\rho(\boldsymbol{w}, b) &= \min_{\boldsymbol{x}_i : y_i = -1} d(\boldsymbol{w}, b; \boldsymbol{x}_i) + \min_{x_i : y_i = 1} d(\boldsymbol{w}, b; \boldsymbol{x}_i) \\
&= \min_{\boldsymbol{x}_i : y_i = -1} \frac{\left|\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b\right|}{\|\boldsymbol{w}\|} + \min_{\boldsymbol{x}_i : y_i = 1} \frac{\left|\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b\right|}{\|\boldsymbol{w}\|} \\
&= \frac{1}{\|\boldsymbol{w}\|}\left(\min_{\boldsymbol{x}_i : y_i = -1} \left|\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b\right| + \min_{\boldsymbol{x}_i : y_i = 1} \left|\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b\right|\right) \\
&= \frac{2}{\|\boldsymbol{w}\|}
\end{aligned}$$

Hence the hyperplane that optimally separates the data is the one that minimises

$$\Phi(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2.$$

subject to $y^i \left[\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b\right] \geq 1$ and hence the Lagrangian is,

$$\Phi(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^n \alpha_i \left(y_i \left[\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b\right] - 1\right), \quad \alpha_i \geq 0.$$

d) **Solve the Lagrangian problem,** $\max_{\boldsymbol{\alpha}} \left(\min_{\boldsymbol{w},b} \Phi(\boldsymbol{w}, b, \boldsymbol{\alpha})\right)$**, to show that the solution for the Lagrange multipliers can be written as a quadratic program,**

$$\min_{\boldsymbol{\alpha}} \tfrac{1}{2}\boldsymbol{\alpha}^{\mathsf{T}} H \boldsymbol{\alpha} + \boldsymbol{c}^{\mathsf{T}}\boldsymbol{\alpha},$$

subject to the constraints,

$$\alpha_i \geq 0, \quad \textstyle\sum_{j=1}^{n} \alpha_j y_j = 0.$$

(8 marks)

The minimum with respect to $\boldsymbol{w}$ and b of the Lagrangian, Φ, is given by,

$$\frac{\partial \Phi}{\partial b} = 0 \;\Rightarrow\; \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial \Phi}{\partial \boldsymbol{w}} = \mathbf{0} \;\Rightarrow\; \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i.$$

Hence, by substitution and rearrangement the dual problem is,

$$\max_{\boldsymbol{\alpha}} -\tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{x}_j + \sum_{k=1}^{n} \alpha_k,$$

and hence the solution to the problem is given by,

$$\min_{\boldsymbol{\alpha}} \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{x}_j - \sum_{k=1}^{n} \alpha_k,$$

with constraints,

$$\alpha_i \geq 0 \quad i = 1, \ldots, n$$

$$\sum_{j=1}^{n} \alpha_j y_j = 0.$$

This is equivalent to the quadratic program formulation by noting $H_{i,j} = y_i y_j \boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{x}_j$ and $c_i = -1$.

e) **What are the Support Vectors and how do these relate to the Lagrange multipliers?**

(3 marks)

The support vectors are the elements in the data-set which lie on the margin. Each data point has an associated Lagrange multiplier and only the support vectors have non-zero Lagrange multipliers.

Question 5

a) **What is an ill-posed problem?**

(5 marks)

An ill-posed problem is one that is not well-posed. A problem is given by a datum g and a solution u, and it is wellposed (in the sense of Hadamard) when 1. For each datum g in a class of functions Y there exists a solution u in a prescribed class X (existence); 2. The solution u is unique in X (uniqueness). 3. The dependence of u upon g is continuous (continuity).

b) **Describe the method of regularisation, making reference to examples in machine learning.**

(9 marks)

Regularisation is a method for restoring well-posedness to an ill-posed problem by introducing a prior on the class of functions. For example, a commonly employed prior is that of smoothness. Hence, functions which are smooth are preferred over functions with large oscillations. This helps avoid problems with overfitting, the phenomenon of fitting an over-complex function. This produces models that typically will generalise better.

Examples in machine learning include, zeroth order regulariser, e.g. SVMs, Gaussian Processes, Bayesian MLPs, second order regularisers which correspond to fitting cubic splines, etc.

(A more detailed discussion should be given for full marks, e.g. including a diagram of a 1D regression function illustrating the limits as the regularisation parameter is varied.)

c) **Show that the solution to the regularisation problem is equivalent to a Maximum A Posteriori (MAP) estimate.**

(6 marks)

The regularisation problem is,

$$\min_f \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2 + \lambda\phi[f]$$

where we illustrate this example with a least squares loss function and ϕ is the regularisation functional. Multiplying by -1 and taking exponents gives,

$$\begin{aligned}
& \max_f \exp\left(-\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i))^2 - \lambda\phi[f]\right) \\
= \; & \max_f \exp\left(-\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i))^2\right)\exp\left(-\lambda\phi[f]\right) \\
\propto \; & \max_f P(D|f)P(f) \\
\propto \; & \max_f P(f|D)
\end{aligned}$$

d) **Discuss how regularisation parameters can be chosen.**

(5 marks)

Regularisation parameters should be chosen to maximise the generalisation error. This can be approximated by splitting the training set into two parts and using the second part to optimise the regularisation parameter. Alternatively, more sophisticated methods using the evidence framework or Monte Carlo methods can be used.