UNIVERSITY OF SOUTHAMPTON COMP3008W1

SEMESTER 2 EXAMINATION 2005/2006

MACHINE LEARNING

Duration: 120 mins

Answer ALL *questions from section A (20 marks)*
and ONE *question from section B (25 marks)*
and ONE *question from section C (25 marks).*

This examination is worth 70%. The coursework was worth 30%.

Calculators without text storage MAY be used.

Number of pages 15

Section A

Question 1

(a) Explain what is meant by generalisation error and describe how it is estimated.

Test of very basic knowledge that is central to the whole course.

The generalisation error is the error of a learning machine on unseen data. It is measured using a validation set (i.e. a set of data which is not used in training). In a classification problem this would usually be the fraction of the validation set the learning machine failed on. For a regression task it would be something like the root mean squared error of the validation set.

(2 marks)

(b) Give a Bayesian interpretation for minimising the sum of the mean squared error plus a regularisation term.

The mean squared error can be thought of a log-likelihood given a Gaussian model for the errors, while the regularisation term can be viewed as a log-prior. Thus, minimising the mean squared error together with a regularisation term can been seen as calculating the maximum a posteriori solution.

(3 marks)

(c) Show that a MLP using linear nodes is no more powerful than a linear perceptron.

A simple proof students have seen in the course.

Assuming we have a two layer MLP. The prediction from any MLP using linear nodes with input x will be of the form

$$y = \sum_i w_i^2 \sum_j w_{i,j}^1 x_j .$$

Changing the order of summation we have

$$y = \sum_j x_j \sum_i w_{i,j}^1 w_i^2 = \sum_j x_j \bar{w}_j$$

where $\bar{w}_j = \sum_i w^1_{i,j} w^2_i$, but this is just the form of a single layer linear perceptron. To generalise to more layers we note that the first layer can be reduced to a single layer. We can carry on this reduction so that a linear network with any number of layers reduces to a single layer.

(5 marks)

(d) Describe what is meant by the terms *training set*, *validation set* and *testing set*.

Training set is the data used for learning. Validation set is the data used for determining hyperparameters. Testing set is the data used for estimating generalisation performance.

(3 marks)

(e) Describe what is meant by the terms *classification*, *regression* and *density estimation*.

Classification involves learning the distinction between two or more classes of data. Regression involves learning a mapping from the input space to a real variable. Density estimation involves learning a distribution from a finite set of examples.

(3 marks)

(f) Describe the *kernel trick* and how it is applied in machine learning.

Usually need to build non-linear models of data to reflect the true underlying model. There are two approaches to this: a) directly build this non-linear model, b) map the data into a high dimensional space using a non-linear transformations and then solve a linear problem in that space. The kernel trick notes that often we can construct algorithms of type (b) which only involve inner products between transformed examples and as such provides a compact mathematical expression for this inner product without having to project our data into a possibly very high dimensional space. It can be employed in many algorithms which can be expressed in the form of inner products, such as KNN, SVM, ...

(4 marks)

Section B

Question 2

The linear perceptron with no bias has a response

$$y = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}.$$

We assume we have a set of training data

$$\{(\boldsymbol{x}^k, t^k) | k = 1, \ldots, P\}$$

where $\boldsymbol{x}^k$ are input patterns and t^k are targets.

(a) Write down an expression the mean square training error, $E(\boldsymbol{w})$, for the linear perceptron.

This question tests some of the more mathematical material cover in the course. It starts simply.

$$E(\boldsymbol{w}) = \frac{1}{P}\sum_{k=1}^{P}\left(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}^k - t^k\right)^2$$

(3 marks)

(b) By writing the inputs as a single matrix $\mathbf{X}$ whose k^{th} column is the input $\boldsymbol{x}^k$ and the targets as a vector $\boldsymbol{t}$ whose k^{th} element is t^k, express the mean squared training error in matrix form.

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

(3 marks)

(c) By computing the gradient of the training error find the value of the weight vector which minimises the training error.

$$\boldsymbol{\nabla} E(\boldsymbol{w}) = \frac{2}{P}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - \mathbf{X}\boldsymbol{t}\right)$$

Setting $\boldsymbol{\nabla} E(\boldsymbol{w}) = 0$ we obtain

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

(4 marks)

(d) Explain what it means for $\mathbf{X}\mathbf{X}^{\mathsf{T}}$ to be ill-conditioned and argue why the weights found will be sensitive to the training data if this is the case.

If XX$^{\mathsf{T}}$ is ill-conditioned it will have at least one small eigenvalue relative to its largest eigenvalue. This will happen when the data does not vary much in one direction.

If XX$^{\mathsf{T}}$ is ill-conditioned then its inverse $\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$ will greatly expand any component of X$\boldsymbol{t}$ in the direction of the eigenvector with small eigenvalue. (In fact, this relative expansion is equal to $\lambda_{max}/\lambda_{min}$.) Thus any small fluctuation in X$\boldsymbol{t}$ (which depends on the training data) in this direction will produce a large change in the weight vector.

(5 marks)

(e) Show that by using a modified error function

$$\hat{E}(\boldsymbol{w}) = E(\boldsymbol{w}) + \nu\,\boldsymbol{w}^{\mathsf{T}}\boldsymbol{w}$$

the weight vector of the linear perceptron will be less sensitive to the training data.

Adding a weight decay term, $\nu\,\boldsymbol{w}^{\mathsf{T}}\boldsymbol{w}$ we obtain

$$\hat{E}(\boldsymbol{w}) = \left(\boldsymbol{w}^{\mathsf{T}}\left(\frac{1}{P}\mathbf{X}\mathbf{X}^{\mathsf{T}} + \nu\mathbf{I}\right)\boldsymbol{w} - \frac{2}{P}\mathbf{X}\boldsymbol{w}^{\mathsf{T}}\mathbf{X}\boldsymbol{t} + \boldsymbol{t}^{\mathsf{T}}\boldsymbol{t}\right)$$

and

$$\boldsymbol{\nabla}\hat{E}(\boldsymbol{w}) = 2\left(\left(\frac{\mathbf{X}\mathbf{X}^{\mathsf{T}}}{P} + \nu\,\mathbf{I}\right)\boldsymbol{w} - \frac{\mathbf{X}}{P}\boldsymbol{t}\right)$$

setting $\nabla \hat{E}(w)$ to zero we find

$$\boldsymbol{w} = \left(\frac{\mathbf{X}\mathbf{X}^{\mathsf{T}}}{P} - \nu\,\mathbf{I} \right)^{-1} \frac{\mathbf{X}}{P}\boldsymbol{t}$$

Thus we have replaced $\mathbf{X}\mathbf{X}^{\mathsf{T}}$ by $\mathbf{X}\mathbf{X}^{\mathsf{T}} + \nu P\mathbf{I}$. This will always be better conditioned. That is, if $\mathbf{X}\mathbf{X}^{\mathsf{T}}/P$ has eigenvalues λ_i then $\mathbf{X}\mathbf{X}^{\mathsf{T}}/P + \nu\mathbf{I}$ will have eigenvalue $\lambda_i + \nu$. Since $\mathbf{X}\mathbf{X}^{\mathsf{T}}$ is positive semi-definite this will always improving the conditioning and hence make the weight vector less sensitive to the data.

(6 marks)

(f) Explain in terms of the bias-variance dilemma why introducing a weight decay term can improve the generalisation performance.

The generalisation performance can be shown to be attributable to two competing factors. The first of these is the bias which measures how well a learning machine could model the underlying classification function given an infinite amount of training data. The second term is the variance in the prediction of networks trained with different randomly selected finite training sets. By removing the sensitivity of the learning machine on the data we directly reduce the variance, although we do so at the cost of increasing the bias (because we are in effect minimising the wrong error function).

(4 marks)

Question 3

Assume we have a learning machine of the form

$$F(\boldsymbol{x}; b, \boldsymbol{w}, \mathbf{Q}) = b + \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{x}^{\mathsf{T}}\mathbf{Q}\boldsymbol{x}$$

where the parameters to be learned, b, $\boldsymbol{w}$ and $\mathbf{Q}$, are a scalar, a vector and a symmetric matrix respectively.

(a) Given training data $\mathcal{D} = \{(\boldsymbol{x}^k, t^k)|k = 1, \ldots, P\}$ write down the mean squared error and compute the gradient with respect to b, w_i and $Q_{i,j}$.

The mean squared error is

$$E(b, \boldsymbol{w}, \mathbf{Q}) = \frac{1}{P}\sum_{k=1}^{P} \left\| t^k - F(\boldsymbol{x}^k; b, \boldsymbol{w}, \mathbf{Q}) \right\|^2$$

The gradients are

$$\frac{\partial E}{\partial b} = \frac{2}{P}\sum_{k=1}^{P}\left(t^k - F(\boldsymbol{x}^k; b, \boldsymbol{w}, \mathbf{Q})\right) = \frac{2}{P}\sum_{k=1}^{P}\delta^k$$

$$\frac{\partial E}{\partial w_i} = \frac{2}{P}\sum_{k=1}^{P}\left(t^k - F(\boldsymbol{x}^k; b, \boldsymbol{w}, \mathbf{Q})\right) x_i^k = \frac{2}{P}\sum_{k=1}^{P}\delta^k x_i^k$$

$$\frac{\partial E}{\partial Q_{i,j}} = \frac{2c}{P}\sum_{k=1}^{P}\left(t^k - F(\boldsymbol{x}^k; b, \boldsymbol{w}, \mathbf{Q})\right) x_i^k x_j^k = \frac{2c}{P}\sum_{k=1}^{P}\delta^k x_i^k x_j^k$$

where $\delta^k = t^k - F(\boldsymbol{x}^k; b, w, \mathbf{Q})$ is the error on the k^{th} example and $c = 1$ if $i = j$ and $c = 2$ otherwise.

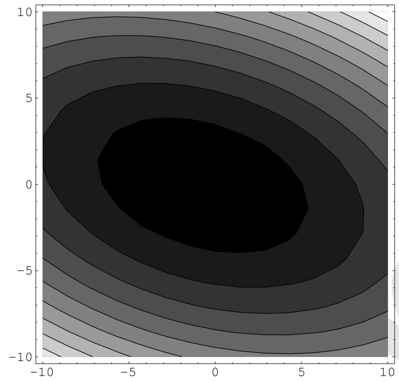
(8 marks)

(b) For two dimensional input patterns, sketch the contour lines where $F(\boldsymbol{x}; b, \boldsymbol{w}, \mathbf{Q})$ is constant. Compare this with similar surfaces for a linear perceptron and an RBF networks.

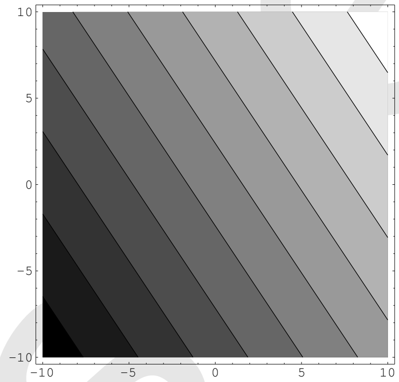
Students will have seen contours for quadratic functions in a different context. The contours for a linear perceptron and RBF will test their understanding of these learning machines.

TURN OVER

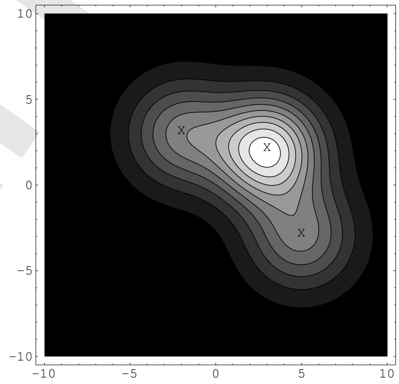
The contour lines for this learning machine will be ellipses (or possibly parabolas if Q is not positive definite). E.g.



The contour for a linear perceptron corresponds to hyper-planes, E.g.



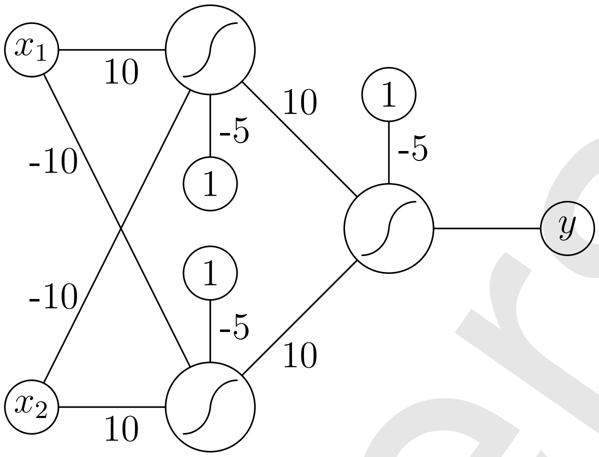
The contour for a RBF will depend on the distance from a set of centres. E.g.



Clearly any rough approximation to these diagrams is acceptable.

(8 marks)

(c) Consider the MLP below



(i) Write an equation describing the response, y, in terms of the input $\boldsymbol{x} = (x_1, x_2)$ assuming a response function $g(x)$.

(ii) Show that the output of the network is constant along the line defined by $x_1 - x_2 = u$ and $x_1 - x_2 = -u$.

(iii) Sketch a contour diagram in the input space showing where the output of the network has equal response.

(i) ***Worth 3 marks.***

$$y = g\left(10\,g(10x_1 - 10x_2 - 5) + 10\,g(-10x_1 + 10x_2 - 5) - 5\right)$$

(ii) ***Worth 3 marks.*** **If we substitute $x_1 = u + x_2$ we get**

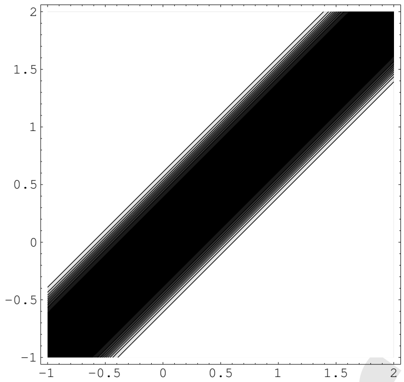
$$y = g\left(10\,g(10u - 5) + 10\,g(-10u - 5) - 5\right)$$

Thus the output depends only on u. If we replace u by $-u$ we get

$$y = g\left(10\,g(-10u - 5) + 10\,g(10u - 5) - 5\right)$$

which is identical to the previous equation.

(iii) ***Worth 3 marks. Should be easy if they get part ii right.***



(9 marks)

Section C

Question 4

(a) What is an ill-posed problem?

Test theoretical understanding An ill-posed problem is one that is not well-posed. A well-posed problem has the following properties:

- **A solution exists**
- **The solution is unique**
- **The solution varies continuously with the data - i.e. a small change in the data with have a small change in the model.**

(7 marks)

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

Test theoretical understanding and algorithm knowledge A good example here would be the Support Vector machine. An SVM introduces regularisation to enforce a maximum margin solution to the problem. ***Sketch example maximum margin solution and explain why it is unique. Explain why the solution will vary continuously with the data. Can contrast his with a perceptron which does not have regularisation or a unique solution. Go on to discuss overlapping classes and the way that the regularisation balances the trade-off between minimising the loss and minimising the length of the weight vector. Finally, a short discussion of how the regularisation parameter is chosen using a data partitioning method such as cross validation. Bonus marks for discussion of bias/variance dilemma and generalisation.***

(9 marks)

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

Test theoretical understanding

g **- data,** f **- model**

The following quantities are defined:

$P(f|g)$ **- the a posteriori probability of the surface f given the data** g**. It is the quantity of interest.**

$P(g|f)$ **- the probability of the data** g **given the surface** f **. It is a model of the noise.**

$P(f)$ **- the a priori probability of the surface** f **. It depends on the a priori knowledge on the surface** f**.**

Bayes' theorem yields:

$$\begin{aligned}
P(f|g) &\propto P(g|f)P(f) \\
P(f) &= e^{-\lambda\phi[f]} \\
P(g|f) &= e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{l}(y_i - f(\mathbf{x}_i))^2} \\
P(f|g) &\propto e^{-H[f]} \\
H[f] &= \frac{1}{2\sigma^2}\sum_{i=1}^{l}(y_i - f(\mathbf{x}_i))^2 + \lambda\phi[f]
\end{aligned}$$

and therefore

$$\arg\max P(f|g) = \arg\min H[f].$$

(9 marks)

Question 5

(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.

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.

(3 marks)

(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.

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

(3 marks)

(c) State the maximum margin principle and derive an expression for the Lagrangian of the resulting optimisation problem.

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,**

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

(10 marks)

(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.

The minimum with respect to w **and** b **of the Lagrangian,** Φ**, is given by,**

$$\begin{aligned}
\frac{\partial \Phi}{\partial b} &= 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
\frac{\partial \Phi}{\partial \boldsymbol{w}} &= \boldsymbol{0} \quad \Rightarrow \quad \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i.
\end{aligned}$$

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 $c_i = -1$.

(9 marks)

END OF PAPER