UNIVERSITY OF SOUTHAMPTON COMP3008W1

SEMESTER 2 EXAMINATION 2010/2011

MACHINE LEARNING

Duration: 120 mins

Answer all parts of the question in section A (20 marks) and TWO *questions from section B (25 marks each)*

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

University approved calculators MAY be used.

 Number of pages 11

Section A

Question 1

(a) Briefly describe what principal component analysis (PCA) does and how it is performed.

Test of general knowledge of preprocessing.

PCA is a method to find a subspace of the input features which describes most of the variation in the set of input patterns. It is performed by computing the covariance matrix and finding eigenvectors with the largest eigenvalues. These eigenvectors form the basis for the subspace into which input patterns are projected.

(6 marks)

(b) Briefly describe K-means clustering.

K-means clustering takes a set of input patterns and finds K clusters following an iterative process. It firstly assigns each pattern to a randomly selected cluster. It then finds the centroids of the clusters. It then assigns each pattern to the closest centroid. Once again the centroids of these new clusters are found. This is repeated until there is no change in which patterns are assigned to which cluster.

(6 marks)

(c) Describe (without mathematics) the Bias-Variance Dilemma.

Test understanding of core machine learning theory.

The expected generalisation error can be seen as arising from two sources.

The bias which is the error of the mean machine over all training data sets in the prediction of the output from training data set to training data set

The variance in the prediction given different training data sets

Too simple machines are likely to have high bias (but low variance) while too complex machines can have lower bias but higher variance.

(6 marks)

(d) Explain how dimensionality reduction (e.g. using PCA or K-means) can reduce the generalisation error.

Tests integration of theory and practice.

Dimensionality reduction will typically reduce the sensitivity of the machine to variations in the training data set and thus reduce the variance. If it retains the salient information it should not strongly affect the bias.

(2 marks)

Section B

Question 2

(a) How would you represent the following categories in a numerical feature vector (input pattern)? Explain your decisions.

(i) Over 18 or not

(ii) Colour preference (red, green, yellow, blue)

(iii) Experience (none, little, medium, very)

This is the non-technical question that covers a lot of issues to do with representing data and measuring performance.

(i) This is a binary attribute that could easily be coded as a binary variable. Stupid to do anything else.

(ii) These attributes have no order. We could use four binary variables such that 0010 would represent yellow. This representation does not impose ordering (although it is not very efficient).

(iii) These categories clearly have an ordering and could be represented by a single variable. e.g. none=0, little=1, medium=2 and very=3. Clearly capturing this ordering is likely to be helpful.

(6 marks)

(b) Describe three different methods for handling missing data (i.e feature vectors with missing features). Discuss their relative benefits and problems.

This has briefly been touched on in the notes. Any three of the following will do.

(i) We could ignore the input, which is simple and unbiased, but could throw away valuable training data.

(ii) We could replace the missing values by their mean, which is again simple and allows us to use all the data, but could be very misleading if the true value of the missing feature is a long way from the mean.

(iii) We could add an additional flag to the feature to show that it is missing. This might allow a sufficiently powerful learning machine to ignore the feature, but it increases the complexity of the machine.

(iv) We could use a full probabilistic model where we integrate out the unknowns. This is the principled approach, but could be very complicated to implement.

(6 marks)

(c) Describe K-fold cross validation and explain why it is used?

In K-fold cross validation we partition all our data into K partitions and use $K-1$ partitions for training and one partition for testing. This is repeated K times holding out different partitions for testing. On each test the learning machine is retrained from scratch.

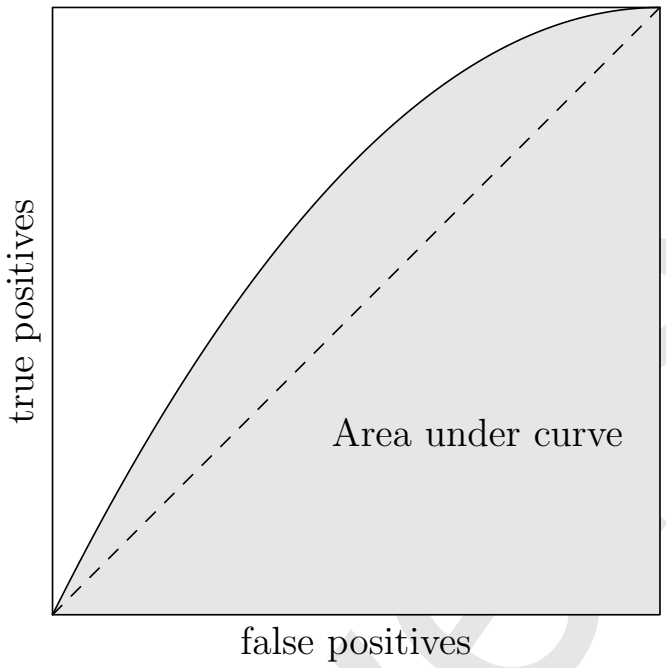
The purpose of K-fold cross validation is to obtain a good estimate of the generalisation error at the same time as using a large proportion of the data for training.

(6 marks)

(d) Describe how to compute an ROC curve and how it can be interpreted?

The ROC curve is used in binary classification problems where there is an adjustable threshold which allows a play off between true positive rate and the false positive rate. We assume we have a positive test set and a negative test set and some outputs from a learning machine for the positives and negatives. We set the threshold so that all our data is considered positive. The true positive rate (the number of predicted positive in the positive test set divided by the number of elements in the test set) is 1, as is the false positive rate (the number of elements in the negative set that are predicted to be positive divided by the total number in the negative test set). As we slide the threshold our false positive rate decreases although so does the true positive rate. The ROC curve plots the true positive rate versus the false positive rate.

TURN OVER



The area under the ROC can be seen as a measure of quality of classifier. It can also be interpreted as the probability that a randomly chosen sample from the positive set has a higher output than a randomly chosen sample from the negative set.

(7 marks)

Question 3

(a) Assume you have a data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ write down the squared training error for a **linear perceptron** with weights $\boldsymbol{w}$.

Tests core knowledge of errors and the linear perceptron.

$$E = \sum_{i=1}^{n} \left(\boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{w} - y_i\right)^2$$

(3 marks)

(b) By writing the training patterns as a matrix $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ and the targets in a vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^{\mathsf{T}}$ write down an expression for the squared training error in matrix form.

This is straight from the notes

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

(3 marks)

(c) Compute the weight vector $\boldsymbol{w}^*$ that minimises the sum of the squared training error plus a regularisation term $\nu\|\boldsymbol{w}\|^2$.

A harder calculation

$\boldsymbol{w}^*$ **satisfies** $\boldsymbol{\nabla}(E + \nu\|\boldsymbol{w}\|^2) = 0$

$$\begin{aligned}\boldsymbol{\nabla}(E + \nu\|\boldsymbol{w}\|^2) &= \boldsymbol{\nabla}\left(\boldsymbol{w}^{\mathsf{T}}\mathbf{X}\mathbf{X}^{\mathsf{T}}\boldsymbol{w} - 2\boldsymbol{w}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{y} + \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y} + \nu\|\boldsymbol{w}\|^2\right)\\ &= 2\left((\mathbf{X}\mathbf{X}^{\mathsf{T}} + \nu\mathbf{I})\boldsymbol{w} - \boldsymbol{X}\boldsymbol{y}\right)\end{aligned}$$

or

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

(8 marks)

TURN OVER

(d) Explain why without regularisation the $\boldsymbol{w}^*$ is ill defined if there are fewer training patterns than features (i.e. the size of the input vectors) and how adding a regularisation term cures this.

Test deep understanding of ill-posed problems and regularisers from equations.

If there are fewer training patterns than features then $\mathbf{XX}^\mathsf{T}$ is singular (it is an $N \times N$ matrix where N is the number of features with rank n). Thus its inverse is ill defined. In contrast, $\mathbf{XX}^\mathsf{T} + \nu\mathbf{I}$ will never be singular (assuming $\nu > 0$ and since $\mathbf{XX}^\mathsf{T}$ is positive semi-definite).

(5 marks)

(e) Explain why adding a regularisation term would make a linear perceptron less sensitive to the training data. Why might this improve the expected generalisation performance?

Show integration of many topics around generalisation.

One major source of sensitivity in the data comes from inverting $\mathbf{XX}^\mathsf{T}$. This is likely to be poorly conditioned if there is not so many training patterns. Adding a regularisation term we have to invert $\mathbf{XX}^\mathsf{T} + \nu\mathbf{I}$ which is better conditioned than $\mathbf{XX}^\mathsf{T}$. To see this let λ_{min} and λ_{max} be the minimum and maximum eigenvalues of $\mathbf{XX}^\mathsf{T}$. The condition number is $\lambda_{max}/\lambda_{min}$. Since $\mathbf{XX}^\mathsf{T}$ is positive definite (we assume it is invertible) the eigenvalues are all strictly positive. The eigenvalues of $\mathbf{XX}^\mathsf{T} + \nu\mathbf{I}$ are equal to $\lambda_i + \nu$ where λ_i are eigenvalues of $\mathbf{XX}^\mathsf{T}$. Thus the condition number will be

$$\frac{\lambda_{max} + \nu}{\lambda_{min} + \nu} < \frac{\lambda_{max}}{\lambda_{min}}.$$

That is the inverse is better conditioned and less sensitive to the training data.

Decreasing the sensitivity to training data will reduce the variance and thus can increase the generalisation performance. Of course there is a cost in that by minimising the learning error plus a regularisation term we are introducing a bias.

(6 marks)

Question 4

(a) Given training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n, \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$ what condition is required that the two classes can be separated by a perceptron.

Easy opening question test basic knowledge.

The data must be linearly separable, i.e. there must exist a dividing plane which separates the two sets.

(2 marks)

(b) Assuming the data is separable there are usually many hyper-planes, $\boldsymbol{w}^\mathsf{T}\boldsymbol{x} - b = 0$, that will separate the data. Explain what criteria is used in the linear Support Vector Machine to choose a unique hyperplane. Explain why this is a good choice?

Basic question about SVMs.

The separating plane chosen by the SVM maximises the margin between the data and the separating plane. This is a good choice as it makes the learning machine robust to errors in the data points and thereby automatically regularises the learning machine, resulting in good generalisation performance.

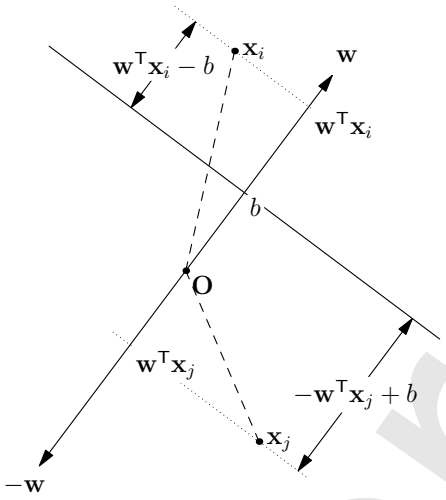
(4 marks)

(c) Show (e.g. by drawing a diagram) that the distance between the separating plane defined by $\boldsymbol{w}^\mathsf{T}\boldsymbol{x} - b = 0$ with $|\boldsymbol{w}| = 1$ and a data point $\boldsymbol{x}_i$ is equal to $\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i - b$ for points on the positive side (with respect to $\boldsymbol{w}$) of the separating plane and $-\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b$ for points on the other side. By rescaling $\boldsymbol{w}$ and b by the margin size show that the maximum margin hyper-plane can be found from the Lagrangian,

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

This is the technical part of the question. It involves a derivation given in the lectures, but requiring a lot of understanding to reproduce.

TURN OVER



The requirement that points in the data set $y_i = +1$ are above the dividing plane by a margin of at least m and those in the data set $y_i = -1$ are below the dividing plane by at least the same margin is that $y_i(\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i - b) \geq m$ for all i. If we divide through by m and define $\tilde{\boldsymbol{x}} = \boldsymbol{x}/m$ and $\tilde{b} = b/m$ then we can write these constraints as $y_i(\tilde{\boldsymbol{w}}^\mathsf{T}\boldsymbol{x}_i - \tilde{b}) \geq 1$. We note that $\|\tilde{\boldsymbol{w}}\|^2 = \|\boldsymbol{w}\|^2/m^2 = 1/m^2$. Thus minimising $\|\tilde{\boldsymbol{w}}\|^2$ is equivalent to maximising m. Finding the maximum margin hyperplane is equivalent to choosing $\tilde{\boldsymbol{w}}$ and $\tilde{b}$ to minimise $\|\tilde{\boldsymbol{w}}\|^2$ subject to the constraints $y_i(\tilde{\boldsymbol{w}}^\mathsf{T}\boldsymbol{x}_i - \tilde{b}) \geq 1$. Using α_i as Lagrange multipliers we get the Lagrangian (except for calling $\tilde{\boldsymbol{w}}$ $\boldsymbol{w}$ and $\tilde{b}$ b).

(8 marks)

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

$$\max_{\boldsymbol{\alpha}} \tfrac{1}{2}\boldsymbol{\alpha}^\mathsf{T}\mathbf{H}\boldsymbol{\alpha} + \boldsymbol{c}^\mathsf{T}\boldsymbol{\alpha},$$
$$\text{subject to the constraints,}$$
$$\alpha_i \geq 0, \quad \textstyle\sum_{j=1}^{n} \alpha_j y_j = 0.$$

Similarly a technically difficult calculation.

To solve the Lagrangian we find the extremal points

$$\nabla_{\boldsymbol{w}}\mathcal{L} = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = 0$$

Thus $\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$. Similarly

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0$$

We note that

$$\|\boldsymbol{w}\|^2 = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{w}^\mathsf{T} \boldsymbol{x}_i = \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\mathsf{T} \boldsymbol{x}_j$$

also the term proportional to b disappears because $\sum_{i=1}^{n} \alpha_i y_i = 0$, thus we are left with

$$\mathcal{L}(\boldsymbol{w}^*, b^*, \boldsymbol{\alpha}) = \tfrac{1}{2} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\mathsf{T} \boldsymbol{x}_j + \sum_{i=1}^{n} \alpha_i$$

with the original constraint $\alpha_i > 0$ (from the KKT conditions) and the new constraint $\sum_{i=1}^{n} \alpha_i y_i = 0$. This is equivalent to the question with H being a matrix with components $H_{ij} = y_i y_j \boldsymbol{x}_i^\mathsf{T} \boldsymbol{x}_j$ and $\boldsymbol{c}$ is the vector of all ones.

(8 marks)

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

Test integration of mathematical formulation and its consequences.

The support vectors are those data points $\boldsymbol{x}_i$ whose corresponding Lagrange multiplier α_i is greater than zero. For all other data points the Lagrange multipliers will be zero.

(3 marks)

END OF PAPER