PAPER CODE NO.	EXAMINER : Dr. Danushka Bollegala
COMP527	DEPARTMENT : Computer Science Tel. No. 0151 7954283

UNIVERSITY OF LIVERPOOL

Resit Examinations 2016/17

Data Mining and Visualisation

TIME ALLOWED : Two and a Half Hours

INSTRUCTIONS TO CANDIDATES

Answer FOUR questions.

If you attempt to answer more questions than the required number of questions (in any section), the marks awarded for the excess questions answered will be discarded (starting with your lowest mark).

UNIVERSITY OF LIVERPOOL

Question 1

A. State two techniques used in Support Vector Machines to classify linearly non-separable datasets. **(2 marks)**

Kernels, slack variables.

(a) Consider the quadratic kernel given by $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y} + 1)^2$ for two vectors $\boldsymbol{x} = (x_1, x_2)^\top$ and $\boldsymbol{y} = (y_1, y_2)^\top$. Show that the quadratic kernel can be written as the inner-product between two 6-dimensional vectors, each containing information only from one of $\boldsymbol{x}$ and $\boldsymbol{y}$. **(4 marks)**

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{y}) &= (\boldsymbol{x}^\top \boldsymbol{y} + 1)^2 \\ &= (x_1 y_1 + x_2 y_2 + 1)^2 \\ &= x_1^2 y_1^2 + x_2^2 y_2^2 + 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 \\ &= (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, 1)^\top (y_1^2, y_2^2, \sqrt{2}y_1, \sqrt{2}y_2, \sqrt{2}y_1 y_2, 1). \end{aligned}$$

As shown in the last step, the quadratic kernel can be represented as the inner-product between two 6-dimensional vectors each containing terms involving only $\boldsymbol{x}$ or $\boldsymbol{y}$.

(b) Explain why a quadratic kernel might be able to learn a decision hyperplane for a non-linearly separable dataset. **(5 marks)**

As shown above, quadratic kernel is projecting 2 dimensional data to a high (6 dimensional) space. Therefore, the probability of finding a hyperplane in this high dimensional space that can linearly separate the dataset will increase. For example, in XOR non-linearity, the cross-product terms $x_1 x_2$ will be sufficient to find a hyperplane.

B. Consider three data points $\boldsymbol{x}_1 = (1, 0)$, $\boldsymbol{x}_2 = (3, -1)$, and $\boldsymbol{x}_3 = (3, 1)$. Here, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are labelled positively (+1), whereas $\boldsymbol{x}_1$ is negative (-1). Answer the following questions related to training a linear-kernel support vector machine on this dataset.

(a) Assuming that the decision hyperplane is defined by a weight vector $\boldsymbol{w} = (w_1, w_2)^\top$ and a bias term b, show that the prediction score y of a test instance $\boldsymbol{z} = (z_1, z_2)$ is given by $y = w_1 z_1 + w_2 z_2 + b$. **(3 marks)**

Prediction score is given by

$$y = \boldsymbol{w}^\top \boldsymbol{z} + b$$

. By substituting for $\boldsymbol{x}, \boldsymbol{z}$ we derive the required result.

(b) Write the equality constraints that must be satisfied by the three support vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$. **(3 marks)**

$$\begin{aligned} w_1 + b &= -1 \\ 3w_1 - w_2 + b &= 1 \\ 3w_1 + w_2 + b &= 1 \end{aligned}$$

(c) By solving the equality constraints you derived in (b), compute w_1, w_2 and b. **(3 marks)**

By solving the three simultaneous linear equations we obtain $w_1 = 1, w_2 = 0, b = -2$.

(d) Plot the decision hyperplane alongside with the support vectors. **(3 marks)**

This will be a straight line parallel to the y-axis and passing through (2,0).

(e) Predict the class label of a test data point $(4, 1)$. **(2 marks)**

$y = 4 \times 1 - 2 = 2 > 0$. *Therefore, this test instance is predicted as positive.*

Question 2 We would like to use the Perceptron algorithm to learn a linear classifier $y = \boldsymbol{w}^\top \boldsymbol{x} + b$, defined by a weight vector $\boldsymbol{w} \in \mathbb{R}^d$ and a bias $b \in \mathbb{R}$ from a training dataset consisting of four instances, $\{(t_n, \boldsymbol{x}_n)\}_{n=1}^4$. Here, $\boldsymbol{x}_1 = (-1, 0)^\top$, $\boldsymbol{x}_2 = (1, 0)^\top$, $\boldsymbol{x}_3 = (1, 1)^\top$, and $\boldsymbol{x}_4 = (-1, 1)^\top$, and the labels are $t_1 = -1$, $t_2 = +1$, $t_3 = +1$, and $t_4 = -1$. We predict an instance $\boldsymbol{x}$ as positive if $\boldsymbol{w}^\top \boldsymbol{x} + b > 0$, and negative otherwise. The initial values of the weight vector and the bias are set respectively to $\boldsymbol{w}^{(0)} = (0, 0)^\top$ and $b = 0$. We visit the training instances in the order $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$, and $\boldsymbol{x}_4$. Answer the following questions.

A. Plot the dataset in the two-dimensional space. **(2 marks)**

x_1 and x_2 will be on the x-axis and x_4 and x_4 will be parallel to this on line $y = 1$.

B. Write the perceptron update rule for a misclassified instance $(t, \boldsymbol{x})$. **(3 marks)**

$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + tx$

C. What will be the values of the weight vector and the bias after observing the instance $\boldsymbol{x}_1$. **(3 marks)**

$y_1 = \boldsymbol{w}^{(0)\top} \boldsymbol{x}_1 + b^{(0)} = 0$. *Therefore, this instance is classified correctly as negative. The weight vector and bias are not updated.* $w^{(1)} = (0, 0)^\top$, $b^{(1)} = 0$.

D. What will be values of the weight vector and the bias after observing $\boldsymbol{x}_2$. **(4 marks)**

$y_2 = 0$. *Therefore, $\boldsymbol{x}_2$ will be incorrectly classified as negative. The weight vector and the bias will be update to* $\boldsymbol{w}^{(2)} = (1, 0), b^{(2)} = 1$

E. What will be the values of the weight vector and the bias after observing $\boldsymbol{x}_3$. **(4 marks)**

$y_3 = (1, 0)^\top (1, 1) + 1 = 2 > 0$. *Therefore, $\boldsymbol{x}_3$ will be correctly classified as positive, requiring no updates.*

F. What will be the values of the weight vector and bias after observing $\boldsymbol{x}_4$. **(4 marks)**

$y_4 = (-1, 1)^\top (1, 0) + 1 = 0$. *Therefore, x_4 is correctly classified as negative, requiring no updates.*

G. If we re-assign the labels as $t_1 = -1, t_2 = +1, t_3 = -1, t_4 = 1$, show that there does not exist a weight vector $\boldsymbol{w} = (w_1, w_2)^\top$ and a bias b that can classify all four instances correctly. **(5 marks)**

First let us assume there exist such a weight vector and a bias. Then the following four inequalities must be satisfied by w_1, w_2, b.

$$
\begin{aligned}
-w_1 + b &\leq 0 && (1)\\
w_1 + b &> 0 && (2)\\
w_1 + w_2 + b &\leq 0 && (3)\\
-w_1 + w_2 + b &> 0 && (4)
\end{aligned}
$$

From (1) *and* (2) *we have $w_1 > 0$. However, from* (3) *and* (4) *we have $w_1 < 0$. There does not exist w_1 that can satisfy both constraints. Likewise, from* (1) *and* (3) *we have $2b + w_2 \leq 0$, whereas from* (2) *and* (4) *we have $w_2 + 2b > 0$. Again we cannot have w_2 and b that satisfy both those constraints. Therefore, there does not exist $\boldsymbol{w}$ and b for this linearly non-separable dataset.*

Question 3 Consider a two-dimensional dataset consisting of five instances $x_0 = (0, 0)$, $x_1 = (1, 1)$, $x_2 = (-1, 1)$, $x_3 = (-1, -1)$, and $x_4 = (1, -1)$. We would like to cluster this dataset into two clusters using the k-means clustering algorithm. Answer the following questions.

A. Write the within cluster sum of squares objective function for a set of K clusters $S_1, S_2, \ldots, S_K$. **(3 marks)**

$$\sum_{j=1}^{K} \sum_{x \in S_j} ||\boldsymbol{x} - \boldsymbol{\mu}_j||^2$$

B. Explain why it is important to randomly initialise the k-means algorithm and run for multiple times. **(4 marks)**

The within cluster sum of squares objective that is optimised by the k-means algorithm is a nonconvex function. Therefore, there is no guarantee that we will reach the minimum value on any single iteration started from an arbitrary initial set of cluster centres. The algorithm can in practice stuck in local minima. To overcome this issue, we must repeat the algorithm with different random initialisations and select the final set of clusters at convergence that correspond to the minimum value of the objective function.

C. Let us assume the two initial cluster centres to be x_1 and x_2. Find the two clusters produced by the k-means algorithm at convergence. **(3 marks)**

$\{x_2, x_3\}, \{x_0, x_1, x_4\}$

D. Compute the value of the within cluster sum of squares objective for the set of clusters obtained in **(C)**. **(2 marks)**

2.0 and 2.67. The total is 4.67

E. Let us assume the two initial cluster centres to be x_0 and x_2. Find the two clusters produced by the k-means algorithm at convergence. **(3 marks)**

$\{x_2\}, \{x_0, x_1, x_3, x_4\}$

F. Compute the value of the within cluster sum of squares objective for the set of clusters obtained in **(E)**. **(2 marks)**

0 and 5.5. The total is 5.5

G. Let us assume the two initial cluster centres to be x_4 and x_2. Find the two clusters produced by the k-means algorithm at convergence. **(3 marks)**

$\{x_2\}, \{x_0, x_1, x_3, x_4\}$

H. Compute the value of the within cluster sum of squares objective for the set of clusters obtained in **(G)**. **(2 marks)**

0 and 5.5. The total is 5.5

I. Based on your results above, which set of clusters should we select? Justify your answer. **(3 marks)**

We must select the set of clusters that correspond to the minimum value of the within cluster sum of squares objective, which is $\{x_2, x_3\}, \{x_0, x_1, x_4\}$ *(corresponding to the objective function value of 4.67)*

Question 4 Let us assume that we must develop a sentiment classifier to predict sentiment of user reviews about products for an online e-commerce portal. Answer the following questions.

A. Providing examples, explain what is meant by the term *stop word* in the context of text mining. **(3 marks)**

Stop words are non-content words such as functional words that can be safely removed from a document when representing the document using some features. For example, words such as a, an, the, is *are considered as stop words.*

B. State a benefit of removing stop words when training a classifier. **(2 marks)**

This will reduce the number of features, thereby speeding up the training, testing times, and save memory when storing the train/test instances.

C. Providing examples, explain what is meant by the term *part-of-speech tagging* in the context of text mining. **(3 marks)**

Part-of-speech (POS) tagging is the task of assigning words in a text parts of speeches such as nouns, verbs, adjectives, adverbs etc. The set of tags is pre-defined such as in the Penn POS tag set. For example, given the sentence, I like eating burgers. *it will be assigned POS tags as follows:* I/PRP like/VBP eating/VBG burgers/NNS

D. Why would it be useful to use part-of-speech tags when training a sentiment classifier? **(2 marks)**

Sentiment is often expressed using adjectives. Therefore, knowing a particular feature (unigram) is an adjective can be useful clue to the sentiment classifier.

E. Why would it be good to use bigrams in addition to unigrams when representing user reviews for training a sentiment classifier. **(4 marks)**

Negation indicators such as not *will be handled as individual features under a unigram only feature representation. This can be problematic when training a sentiment classifier because we cannot accurately detect what has been negated in the document. On the other hand, bigrams will retain the negation indicators with the target of the negation. For example,* not+great *will be represented as a single bigram, indicating a negative sentiment.*

F. If the user reviews are rated from 1 to 5 stars in a discrete ordinal scale, where higher the value more positive the sentiment, how would you assign labels to the reviews such that you can train a binary sentiment classifier? **(3 marks)**

We could discretise the ratings. For example, we can assign a negative *(-1) label to reviews rated either 1 or 2, whereas a* positive *(+1) label can be assigned to reviews rated either 4 or 5. We can ignore all reviews with rating 3 because they can be ambiguous with respect to the sentiment expressed.*

G. If our dataset contains identical copies (duplicate reviews), will it affect the performance of a naive Bayes classifier? Explain your answer. **(4 marks)**

Yes. Each feature in the duplicated reviews will be counted multiple times within each class. This will increase the likelihoods $p(w|t)$ *of words, thereby affecting the posterior probability according to the Bayes' rule. Answers without the explanation will receive only 1 mark.*

H. We would like to apply singular value decomposition to reduce the size of the feature vectors representing the reviews. Assuming that you do not have a separate validation dataset, how can you determine the optimal dimensionality for the singular value decomposition? **(4 marks)**

We can split the training dataset into two parts: a train part (80%) and a validation part (20%). We can then perform singular value decomposition-based dimensionality reduction under different dimensionalities and evaluate the sentiment classification accuracy of the trained binary sentiment classifier on the validation part of the dataset. We can then select the dimensionality that returns the best classification accuracy.

Question 5 Consider the multi-layer feedforward neural network shown in Figure 1. This neural network has 3 inputs x_1, x_2 and x_3 connected to a hidden layer consisting of two nodes h_1 and h_2. The weight of the edge connecting x_i to h_j is w_{ji}. The two hidden nodes are connected to the output node o. The weight of the edge connecting the hidden node h_i to the output node o is u_i. The activation functions at hidden and output layers is set to sigmoid function defined as follows:

$$\sigma(\theta) = \frac{1}{1 + \exp(-\theta)}$$

Moreover, squared error is used as the loss function at the output node, and is defined as,

$$E(o, t) = \frac{1}{2}(o - t)^2,$$

where t is the target output.

Answer the following questions.

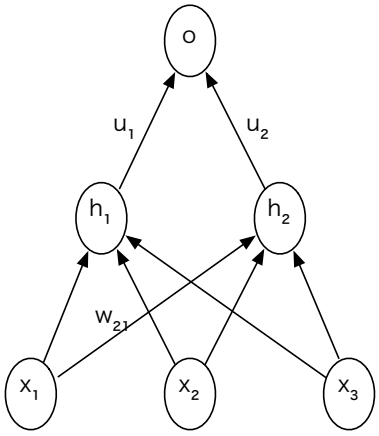

Figure 1: A multi-layer feedforward neural network.

A. State one advantage and one disadvantage of using a single layer neural network vs. multi-layer neural network. **(2 marks)**.

A single layer neural network can only classify linearly separable data, whereas a multi-layer neural network has the capability to handle non-linear data. However, the number of edge weights grow rapidly with the number of layers used in the neural network making it (a) time consuming to train, (b) require large labeled datasets, and (c) likely to overt to the train data.

B. Using the symbols defined in Figure 1, compute the activation at h_1. **(3 marks)**

$$a(h_1) = \sigma(x_1 w_{11} + x_2 w_{12} + x_3 w_{13})$$

C. Compute the gradient of the loss with respect to the output o. **(3 marks)**

$$\frac{\partial E}{\partial o} = o - t$$

D. Compute the gradient of the weight u_1 with respect to the loss. **(3 marks)**

$$\frac{\partial E}{\partial u_1} = (o - z)\sigma'(a(h_1)u_1 + a(h_2)u_2)a(h_1)$$

E. Compute the gradient of the weight w_{12} with respect to the loss. **(4 marks)**

$$\frac{\partial E}{\partial w_{12}} = (o - z)\sigma'(a(h_1)u_1 + a(h_2)u_2)\sigma'(x_1w_{11} + x_2w_{12} + x_3w_{13})x_2$$

F. Using the Stochastic Gradient Descent rule, write the update rule for w_{12}. **(4 marks)**

$$w_{12}^{(k+1)} = w_{12}^{(k)} - \eta\frac{\partial E}{\partial w_{12}}$$

*We can substitute for $\frac{\partial E}{\partial w_{12}}$ from part **(E)** above.*

G. Using the update rule derived in **(F)**, explain why we should scale the initial values of the weights such that the activation at the output node does not fall in the saturated regions of the logistic sigmoid function. **(3 marks)**

At the saturated regions of the logistic sigmoid we have either $\sigma(\theta) \to 0$ or $\sigma(\theta) \to 1$, for which the update becomes small (or zero). Therefore, the weight will get stuck in the initial values and will not get updated.

H. If we would like to learn a sparse neural network where most of the edge weights are set to zero, how can we modify the loss function. **(3 marks)**

We can add a regularisation term to the loss function. For obtaining sparse solutions, for example, we can add ℓ_1 regularisation terms for $\boldsymbol{w} = (w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23})^\top$ and $\boldsymbol{u} = (u_1, u_2)^\top$. The modified loss function will be then,

$$E = \frac{1}{2}(o - z)^2 + \lambda_1 ||\boldsymbol{w}||_1 + \lambda_2 ||\boldsymbol{u}||_1$$