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

UNIVERSITY OF LIVERPOOL

SECOND SEMESTER EXAMINATIONS 2017/18

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, the marks awarded for the excess questions answered will be discarded (starting with your lowest mark).

Question 1 Consider the sentence
S = "I would love to go to London by train tomorrow or at least by bus next week".
Answer the following questions.

A. Given the stop word list [*to, at, by, the, or, would*], represent S as a bag-of-unigrams. **(2 marks)**

I, love, go, London, train, tomorrow, least, bus, next, week

B. Generate all possible bigrams from S without removing the stop words. How many bigrams do you get? **(2 marks)**

16

C. Generate all possible bigrams from S after removing the stop words from S. How many bigrams do you get? **(2 marks)**

9

D. State one advantage of removing stop words in text mining tasks. **(2 marks)**

Reduction of the number of features, speed up, smaller model sizes.

E. State one disadvantage of removing stop words in text mining tasks. **(2 marks)**

Removing features such as *to* or *would* will likely to loose the information about respectively purpose/direction of an action or the modality of the text.

F. What is meant by *part-of-speech* in text mining? **(2 marks)**

Part-of-speeches are nouns, verbs, adjectives, adverbs, etc. that indicate syntactic categories of words.

G. Given the unigram feature set $\{\mathrm{London, train, bus, John, week, love, Liverpool}\}$, represent S by a binary-valued feature vector $\boldsymbol{s}$ **(2 marks)**

Let us assign indexes to the features as follows: London=0, train=1, bus=2, John=3, week=4, love=5, Liverpool=6. Then $\boldsymbol{s} = (1, 1, 1, 0, 1, 1, 0)^\top$.

H. Compute the ℓ_2 norm of $\boldsymbol{s}$. **(2 marks)**

$\sqrt{5}$

I. Compute the ℓ_1 norm of $\boldsymbol{s}$. **(2 marks)**

5

J. Assume that the d-dimensional pre-trained word embedding for a word w is given by $v(w)$. Let us denote the set of unigrams computed in part (A) above by $\mathcal{V}$. Propose a method to create a d-dimensional embedding for the sentence s using $\mathcal{V}$ and the word embeddings. **(3 marks)**

One approach to do this would be to compute the centroid of the word embeddings for the words in the sentence. Specifically, $\boldsymbol{s} = \frac{1}{|\mathcal{V}|}\sum_{w \in \mathcal{V}} v(w)$

K. State a disadvantage of the sentence embedding method that you described in part (J). **(2 marks)**

The sentence embedding method described in part J ignores the ordering of the words in a sentence. Another disadvantage would be that it weighs all words equally when computing the sentence embedding.

L. Propose a method to overcome the disadvantage that you described in part (K). **(2 marks)**

We can weigh each word using inverse document frequency (IDF) (or some other salience measure) before computing the centroid to incorporate the importance of a word when representing a sentence. To make the sentence embedding order-sensitive, we can use a recurrent neural network or use bigram (or higher-order *n*-grams) embeddings rather than using unigram embeddings.

Question 2 Consider a training dataset $\mathcal{D} = \{(\boldsymbol{x}_n, t_n)\}_{n=1}^{4}$, where $\boldsymbol{x}_n \in \mathbb{R}^2$ and $t_n \in \{-1, 1\}$. Here, $\boldsymbol{x}_1 = (0,1)^\top$, $\boldsymbol{x}_2 = (-1,0)^\top$, $\boldsymbol{x}_3 = (0,-1)^\top$, and $\boldsymbol{x}_4 = (1,0)^\top$. We would like to train a binary Perceptron on $\mathcal{D}$ parametrised by the weight vector $\boldsymbol{w} = (\alpha, \beta)^\top$ and bias b. Answer the following questions.

A. Show that if the labels are $t_1 = t_2 = 1$ and $t_3 = t_4 = -1$, then $\mathcal{D}$ can be perfectly classified by the Perceptron with $\boldsymbol{w} = (-1, 1)$ and $b = 0$. **(4 marks)**

$$\begin{aligned}
\boldsymbol{w}^\top \boldsymbol{x}_1 + 0 &= 1 \geq 0 \\
\boldsymbol{w}^\top \boldsymbol{x}_2 + 0 &= 1 \geq 0 \\
\boldsymbol{w}^\top \boldsymbol{x}_3 + 0 &= -1 < 0 \\
\boldsymbol{w}^\top \boldsymbol{x}_4 + 0 &= -1 < 0
\end{aligned}$$

Therefore $\mathcal{D}$ is perfectly classified by this Perceptron.

B. Now let us relabel $\mathcal{D}$ such that $t_1 = t_4 = 1$ and $t_2 = t_3 = -1$. Initialising $\alpha = \beta = b = 0$ and visiting the training instances in $\mathcal{D}$ once in the order $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$, and $\boldsymbol{x}_4$, compute the final weight vector and the bias. **(4 marks)**

$\boldsymbol{x}_1$ is correctly classified so the weight vector does not change after observing $\boldsymbol{x}_1$. However, $\boldsymbol{x}_2$ is misclassified and the weight vector is updated to $(0,0)^\top - (-1,0)^\top = (1,0)^\top$ and the bias is updated to $b = -1$. Then, $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ are correctly classified by this weight vector and bias. Therefore, the final weight vector will be $(1, 0)$ and the bias will be $b = -1$.

C. Now let us relabel $\mathcal{D}$ such that $t_1 = t_3 = 1$ and $t_2 = t_4 = -1$. Assuming that $b = 0$, write the conditions that must be satisfied by the activation scores for each of the four points in $\mathcal{D}$, if it is to be correctly classified by $\boldsymbol{w} = (\alpha, \beta)$. **(4 marks)**

$$\begin{aligned}
(\alpha, \beta)^\top (0, 1) \geq 0 &\rightarrow \beta \geq 0 \\
(\alpha, \beta)^\top (-1, 0) < 0 &\rightarrow \alpha > 0 \\
(\alpha, \beta)^\top (0, -1) \geq 0 &\rightarrow \beta \leq 0 \\
(\alpha, \beta)^\top (1, 0) < 0 &\rightarrow \alpha < 0
\end{aligned}$$

D. Using the inequalities you wrote in part (c) show that there does not exist a Perceptron that can linearly separate $\mathcal{D}$ with a zero bias. **(2 marks)**

The second and fourth inequalities cannot be satisfied simultaneously by α. Therefore, $\boldsymbol{w} = (\alpha, \beta)$ does not exist.

E. Show that when $t_1 = t_3 = 1$ and $t_2 = t_4 = -1$, there does not exist a Perceptron even with $b \neq 0$. **(3 marks)**

The inequalities with the bias are as follows:

$$\begin{aligned} \beta + b &\geq 0 \\ -\alpha + b &< 0 \\ -\beta + b &\geq 0 \\ \alpha + b &< 0 \end{aligned}$$

From these four inequalities we have $b \geq 0$ and $b < 0$, which cannot be satisfied simultaneous by b.

F. Given a feature vector $\boldsymbol{x} = (x_1, x_2)^\top$, let us consider a kernel ψ that maps $\boldsymbol{x} \in \mathbb{R}^2$ to $\boldsymbol{x}^* \in \mathbb{R}^4$ such that $\boldsymbol{x}^* = (x_1, x_2, x_1^2, x_2^2)^\top$. Compute the projections $\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \boldsymbol{x}_3^*$, and $\boldsymbol{x}_4^*$ respectively of $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$, and $\boldsymbol{x}_4$ under ψ. **(4 marks)**

$$\begin{aligned} \boldsymbol{x}_1^* &= (0, 1, 0, 1)^\top \\ \boldsymbol{x}_2^* &= (-1, 0, 1, 0)^\top \\ \boldsymbol{x}_3^* &= (0, -1, 0, 1)^\top \\ \boldsymbol{x}_4^* &= (1, 0, 1, 1)^\top \end{aligned}$$

G. Is $\mathcal{D}^* = \{(\boldsymbol{x}_1^*, 1), (\boldsymbol{x}_2^*, -1), (\boldsymbol{x}_3^*, 1), (\boldsymbol{x}_4^*, -1)\}$ linearly separable? If yes, then give a weight vector and a bias term of a Perceptron that would correctly classify all four instances in $\mathcal{D}^*$. If no, then explain why $\mathcal{D}^*$ is not linearly separable. **(4 marks)**

$\mathcal{D}^*$ can be perfectly classified (therefore linearly separable) by $\boldsymbol{w}^* = (0, 0, 1, 0)^\top$ and $b^* = 0$.

Question 3 Consider five data points in $\mathbb{R}^2$ given by $\boldsymbol{x}_1 = (0,0)^\top$, $\boldsymbol{x}_2 = (1,0)^\top$, $\boldsymbol{x}_3 = (1,1)^\top$, $\boldsymbol{x}_4 = (0,1)^\top$, and $\boldsymbol{x}_5 = (-1,-1)^\top$. Answer the following questions about this dataset.

A. Let us assume that we clustered this dataset into two clusters $\mathcal{S}_1 = \{x_4, x_3\}$ and $\mathcal{S}_2 = \{x_1, x_2, x_5\}$. Moreover, let us represent $\mathcal{S}_1$ and $\mathcal{S}_2$ by two 2-dimensional vectors respectively $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. Write the within-cluster sum of squares objective, $f(\mathcal{S}_1, \mathcal{S}_2)$ for this clustering. **(3 marks)**

$$f(\mathcal{S}_1, \mathcal{S}_2) = ||\boldsymbol{x}_4 - \boldsymbol{\mu}_1||^2 + ||\boldsymbol{x}_3 - \boldsymbol{\mu}_1||^2 + ||\boldsymbol{x}_1 - \boldsymbol{\mu}_1||^2 + ||\boldsymbol{x}_2 - \boldsymbol{\mu}_2||^2 + ||\boldsymbol{x}_5 - \boldsymbol{\mu}_2||^2$$

B. Write $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ that would minimise $f(\mathcal{S}_1, \mathcal{S}_2)$. **(4 marks)**

$$\frac{\partial f}{\partial \boldsymbol{\mu}_1} = -2(\boldsymbol{x}_4 - \boldsymbol{\mu}_1) - 2(\boldsymbol{x}_3 - \boldsymbol{\mu}_1) = 0$$

Gives, $\boldsymbol{\mu}_1 = (\boldsymbol{x}_4 + \boldsymbol{x}_3)/2 = (0.5, 1.0)$. Likewise, we can get $\boldsymbol{\mu}_2 = (\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_5)/3 = (0, -1/3)$.

C. Assuming that we initialised $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ such that $\boldsymbol{\mu}_1 = (0.5, 0.5)^\top$ and $\boldsymbol{\mu}_2 = (-2, -2)^\top$. Following the procedure of k-means clustering, assign the data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5$ to the two clusters $\mathcal{S}_1$ and $\mathcal{S}_2$ represented respectively by $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. **(2 marks)**

$\mathcal{S}_1 = \{x_4, x_3, x_2, x_1\}, \mathcal{S}_2 = \{x_5\}$

D. Compute the next values for $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ following the assignment done in part (3). **(2 marks)**

$\boldsymbol{\mu}_1 = (0.5, 0.5)^\top$ and $\boldsymbol{\mu}_2 = (-1, -1)^\top$

E. Has the k-means clustering converged after the update in part (D)? Explain your answer. **(4 marks)**

There is a 0.5 probability that x_1 could be assigned to $\mathcal{S}_2$. If that happens, the clustering continues and has not converged. However, with 0.5 probability x_1 can get assigned to $\mathcal{S}_2$, in which case clusters do not change and has converged.

F. Consider the two clusters $\mathcal{S}_1 = \{R, R, B\}$ and $\mathcal{S}_2 = \{R, B\}$ consisting of red (R) and blue (B) colour balls. Compute the purity for this clustering. **(3 marks)**

(2+1)/5 = 0.6

G. Compute the rand index for the clustering described in part (F). **(4 marks)**

TP=1, FP=5, FN=3, and TN=3. Therefore, rand index is (1+3) / (1+3+5+3) = 0.33

H. Compute the precision, recall and F-score for the clustering described in part (F). **(3 marks)**

precision = 1/6, recall = 1/4, and F = 1/5

Question 4

A. Consider flipping a coin C and a dice D at the same time and observing the outcomes. The events corresponding to the two sides of the coin are denoted by c_1 (head) and c_2 (tail), whereas those for the dice are denoted by $d_1, d_2, d_3, d_4, d_5, d_6$ respectively for the six sides of the dice. The probability of an event e is denoted by $p(e)$. The joint observations from 40 trials are summarised in Table 1. Answer the following questions about this experiment.

	d_1	d_2	d_3	d_4	d_5	d_6
c_1	2	4	3	5	2	4
c_2	3	3	4	2	4	4

Table 1: Frequency of the events observed in the experiment.

(a) Compute $p(c_1)$, $p(c_2)$ and decide whether C is a biased coin or not. **(3 marks)**

$p(c_1) = p(c_2) = 20/40 = 0.5$. Therefore, C is an unbiased coin.

(b) Compute the mutual information between C and D. (You do not need to simplify the logarithms) **(4 marks)**

$$
\begin{aligned}
I(C, D) &= \sum_{i,j} p(c_i, d_j) \log \frac{p(c_i, d_j)}{p(c_i)p(d_j)} \\
&= \frac{2}{40} \log \frac{2*40}{20*5} + \frac{4}{40} \log \frac{4*40}{20*7} + \frac{3}{40} \log \frac{3*40}{20*7} + \frac{5}{40} \log \frac{5*40}{20*7} \\
&+ \frac{2}{40} \log \frac{2*40}{20*6} + \frac{2}{40} \log \frac{2*40}{20*6} + \frac{4}{40} \log \frac{4*40}{20*8} + \frac{3}{40} \log \frac{3*40}{20*5} \\
&+ \frac{3}{40} \log \frac{3*40}{20*7} + \frac{2}{40} \log \frac{2*40}{20*7} + \frac{4}{40} \log \frac{2*40}{20*6} + \frac{4}{40} \log \frac{4*40}{20*8}
\end{aligned}
$$

B. Consider four product reviews r_1, r_2, r_3, r_4 represented in a three dimensional feature space consisting of the unigrams *awesome*, *awful* and *burger*. The frequency of each unigram in each review is shown in Table 2 and their sentiment labels (1 and -1 respectively denote positive and negative sentiment). Answer the following questions.

Review	awesome	awful	burger	label (t)
r_1	2	0	3	1
r_2	2	1	0	1
r_3	0	2	2	-1
r_4	2	3	1	-1

Table 2: A set of four reviews represented using three attributes.

(a) Compute the marginal probabilities $p(\mathrm{awesome})$, $p(\mathrm{awful})$ and $p(\mathrm{burger})$. **(3 marks)**

$p(\mathrm{awesome}) = p(\mathrm{awful}) = p(\mathrm{burger}) = 6/18$

(b) Compute the conditional probabilities $p(\text{awesome}|t=1), p(\text{awful}|t=1)$ and $p(\text{burger}|t=1)$. **(3 marks)**

$p(\text{awesome}|t=1) = 4/8, p(\text{awful}|t=1) = 1/8$ and $p(\text{burger}|t=1) = 3/8$

(c) Compute $p(t=1)$ and $p(t=-1)$. **(2 marks)**

$p(t=1) = 1/2, p(t=-1) = 1/2$

(d) Compute $p(t=1|r_4)$ (You do not need to simplify the answer). **(4 marks)**

$$p(t=1|r_4) = \frac{p(r_4|t=1)p(t=1)}{p(r_4)}$$

$$p(r_4|t=1) = p(\text{awesome}|t=1)^2 \times p(\text{awful}|t=1)^3 \times p(\text{burger}|t=1)^1$$

$$p(r_4) = p(\text{awesome})^2 \times p(\text{awful})^3 \times p(\text{burger})^1$$

$$p(t=1|r_4) = \frac{(4/8)^2(1/8)^2(3/8)(1/2)}{(2/18)^2(3/18)^3(1/18)}$$

(e) Apply Laplace smoothing for the occurrences of attributes in reviews shown in Table 2 and compute $p(t=1|r_3)$ using the smoothed counts. (You do not need to simplify the answer) **(6 marks)**

Smoothed counts are shown in Table 3. Therefore, we have,

Review	awesome	awful	burger	label (t)
r_1	3	1	4	1
r_2	3	2	1	1
r_3	1	3	3	-1
r_4	3	4	2	-1

Table 3: Smoothed counts.

$$\begin{aligned} p(t=1|r_3) &= \frac{p(r_3|t=1)p(t=1)}{p(r_3)} \\ &= \frac{(6/16)^3(3/16)^4(5/16)^2(14/30)}{(1/30)^3(3/30)^4(3/30)^2} \end{aligned}$$

2 marks are awarded for computing the smoothed counts and another 4 marks for correctly computing $p(t=1|r_3)$. No penalties for not simplifying the answers.

Question 5 Consider the neural network with one hidden layer shown in Figure 1. Here, a two-dimensional input (represented by two features x_1, x_2) is multiplied by a weight matrix $\mathbf{W}$ and subsequently a nonlinear activation of $\tanh(\theta) = \frac{\exp(\theta)-\exp(-\theta)}{\exp(\theta)+\exp(-\theta)}$ is applied. The outputs after applying the activation at the hidden layer are z_1 and z_2, which are linearly weighted respectively by u_1 and u_2 to compute the prediction y. The output y is compared against the target output t for the instance $\boldsymbol{x} = (x_1, x_2)^\top$ to compute the loss given by

$$E(\boldsymbol{x}, t) = \frac{1}{2}(y - t)^2$$

Answer the following questions.

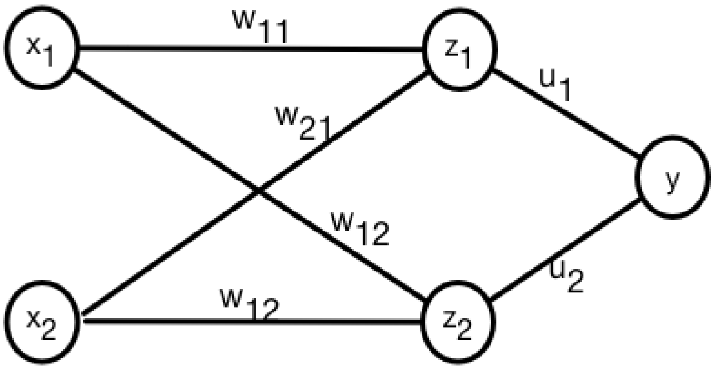

Figure 1: A neural network that takes two-dimensional feature vector and applies a tanh activation in the hidden layer. The weight connecting nodes x_i and z_j is set to w_{ij}, whereas the weight connecting node z_i to the output node y is set to u_i.

A. Express the output y in terms of z_1, u_1, z_2, u_2. **(2 marks)**

$y = z_1 u_1 + z_2 u_2$

B. Write z_1 using the input, weights in the first layer and the activation function. **(2 marks)**

$z_1 = \tanh(x_1 w_{11} + x_2 w_{21})$

C. Write the loss gradient w.r.t. y. **(2 marks)**

$\frac{\partial E}{\partial y} = (y - t)$

D. Show that

$$\frac{\partial z_1}{\partial w_{11}} = \left(1 - \tanh^2(x_1 w_{11} + x_2 w_{21})\right) x_1.$$

(4 marks)

This follows from the differentiation of $z_1 = \tanh(x_1 w_{11} + x_2 w_{21})$ w.r.t. w_{11} and applying $\tanh'(\theta) = 1 - \tanh^2(\theta)$

E. Write $\frac{\partial E}{\partial w_{11}}$, the loss gradient w.r.t. w_{11}. **(4 marks)**

$$\frac{\partial E}{\partial w_{11}} = (y - t) u_1 x_1 \left(1 - \tanh^2(x_1 w_{11} + x_2 w_{21})\right)$$

F. Derive the stochastic gradient descent update rule for w_{11}. **(3 marks)**

$w_{11}^{(k+1)} = w_{11}^{(k)} - \eta \frac{\partial E}{\partial w_{11}}$. The loss gradient w.r.t to w_{11} computed in part (E) must be substituted.

G. Explain why it would be inappropriate to initialise the weights w_{ij} in the first layer to high numerical values. **(2 marks)**

This would increase the activation score that is input to the tanh and for high (positive or negative) scores, the gradient of the activation function will be close to zero. Therefore, the weights will not be updated after the initialisation. This is called the *saturation* of the activation.

H. State a solution that you can use to reduce the overfitting in a neural network. **(2 marks)**

dropout, regularisation, early stopping, reduce the number of hidden layers.

I. Consider the ℓ_2 regularised version of the loss function given by

$$E(\boldsymbol{x}, t) = \frac{1}{2}(y - t)^2 + \lambda(w_{11}^2 + w_{12}^2 + w_{21}^2 + w_{22}^2) + \mu(u_1^2 + u_2^2),$$

where λ and μ are regularisation coefficients. Derive the update rule for w_{11} under this regularisation. **(4 marks)**

$$w_{11}^{(k+1)} = w_{11}^{(k)} - \eta\left((y - t)u_1 x_1\left(1 - \tanh^2(x_1 w_{11} + x_2 w_{21})\right)\right) + 2\lambda w_{11}$$