School of Computing and Information Systems
The University of Melbourne
COMP90049 Introduction to Machine Learning (Semester 2, 2020)

Tutorial exercises: Week 6

1. What is **Logistic Regression**?

   (i). How is **Logistic Regression** similar to **Naive Bayes** and how is it different? In what circumstances would the former be preferable, and in what circumstances would the latter?

   (ii). What is “logistic”? What are we “regressing”?

2. Bob tries to gather information about this year's apple harvest, and ran a search. He retrieved a number of articles, but found that a large portion of the retrieved articles are about the Apple laptops and computers -- and hence irrelevant to his search. He built the following data set of 5 training instances and 1 test instance. Develop a logistic regression classifier to predict label $\hat{y} = 1$ (fruit) and $\hat{y} = 0$ (computer).

ID	*apple*	*ibm*	*lemon*	*sun*	CLASS
	TRAINING INSTANCES				
A	1	0	1	5	1 FRUIT
B	1	0	1	2	1 FRUIT
C	2	0	0	1	1 FRUIT
D	2	2	0	0	0 COMPUTER
E	1	2	1	7	0 COMPUTER
	TEST INSTANCES				
T	1	2	1	5	?

For the moment, we assume that we already have an estimate of the model parameters, i.e., the weights of the 4 features (and the bias θ_0) is $\hat{\theta} = [\theta_0, \theta_1, \theta_2, \theta_3, \theta_4] = [0.2, 0.3, -2.2, 3.3, -0.2]$.

   (i). Explain the intuition behind the model parameters, and their meaning in relation to the features

   (ii). Predict the test label.

   (iii). [OPTIONAL] Recall the conditional likelihood objective

$$\log \mathcal{L}(\theta) = -\sum_{i=1}^{n} y_i \log\big(\sigma(x_i;\theta)\big) + (1 - y_i)\log\big(1 - \sigma(x_i;\theta)\big)$$

   We want to make sure that the Loss (the negative log likelihood) our model, is lower when its prediction the correct label for test instance T, than when it’s predicting a wrong label.

   Compute the negative log-likelihood of the test instance (1) assuming that the true label y = 1 (fruit), i.e., our classifier made a mistake; and (2) assuming the true label as y = 0 (computer), i.e., our classifier predicted correctly.

3. For the model created in question 3, compute a single gradient descent update for parameter θ_1 given the training instances given above. Recall that for each feature j, we compute its weight update as

$$\theta_j \leftarrow \theta_j - \eta \sum_i (\sigma(x_i; \theta) - y_i) x_{ij}$$

Summing over all training instances i. We will compute the update for θ_j assuming the current parameters as specified above, and a learning rate $\eta = 0.1$.

4. [OPTIONAL] What is the relation between "odds" and "probability"?

5. [OPTIONAL] Why is a perceptron (which uses a **sigmoid** activation function) equivalent to *logistic regression*?

6. Consider the following training set:

(x_1, x_2)	y
(0,0)	0
(0,1)	1
(1,1)	1

With the bias value of 1, the initial weight function of $\theta = \{\theta_0, \theta_1, \theta_2\}$ = {0.2, -0.4, 0.1} and learning rate of $\eta = 0.2$.

Consider the activation function of the perceptron as the step function

$$f = \begin{cases} 1 & if\ \Sigma > 0 \\ 0 & otherwise \end{cases}$$

(i). Can the perceptron learn a perfect solution for this data set?

(ii). Draw the perceptron graph and calculate the accuracy of the perceptron on the training data before training?

(iii). Using the perceptron *learning rule* and the learning rate of $\eta = 0.2$. Train the perceptron **for one epoch**. What are the weights after the training?

(iv). [OPTIONAL] What is the accuracy of the perceptron on the training data after training for one epoch? Did the accuracy improve?