COMP 527 - 2018 - CA Assignment 1
Data Classification
Implementing Perceptron algorithm

Assessment Information

Assignment Number	1 (of 2)
Weighting	12%
Assignment Circulated	1st February 2018
Deadline	8th March 2018, 15:00 UK Time (UTC)
Submission Mode	Electronic via Departmental submission system
Learning outcome assessed	(1) A critical awareness of current problems and research issues in data mining. (3) The ability to consistently apply knowledge concerning current data mining research issues in an original manner and produce work which is at the forefront of current developments in the sub-discipline of data mining.
Purpose of assessment	This assignment assesses the understanding of the Perceptron algorithm by implementing a binary Perceptron for text clustering.
Marking criteria	Marks for each question are indicated under the corresponding question.
Submission necessary in order to satisfy Module requirements?	No
Late Submission Penalty	Standard UoL Policy.

1 Objectives

This assignment requires you to implement the Perceptron algorithm using the Python programming language.

> **Note that *no credit will be given for implementing any other types of classification algorithms or using an existing library for classification instead of implementing it by yourself. However, you are allowed to use numpy library for accessing data structures such as numpy.array. But it is not a requirement of the assignment to use numpy. You must provide a README file describing how to run your code to re-produce your results.***

2 Text Classification using Binary Perceptron Algorithm

Download the *CA1data.zip* file from the COMP 527 web site and uncompress it. Inside, you will find two files: *train.data* and *test.data*, corresponding respectively to the train and test data to be used in this assignment. Each line in the file represents a different train/test instance. The first four values (separated by commas) are feature values for four features. The last element is the class label (class-1, class-2 or class-3).

Questions/Tasks

(1) Explain the Perceptron algorithm for the binary classification case, providing its pseudo code. **(20 marks)**

(2) Implement a binary perceptron. **(30 marks)**

(3) Use the binary perceptron to train classifiers to discriminate between (a) class 1 and class 2, (b) class 2 and class 3 and (c) class 1 and class 3. Report the train and test classification accuracies for each of the three classifiers after 20 iterations. Which pair of classes is most difficult to separate? **(20 marks)**

(4) For the classifier (a) implemented in part (3) above, which feature is the most discriminative? **(5 marks)**

(5) Extend the binary perceptron that you implemented in part (2) above to perform multi-class classification using the 1-vs-rest approach. Report the train and test classification accuracies for each of the three classes after training for 20 iterations. **(15 marks)**,

(6) Add an ℓ_2 regularisation term to your multi-class classifier implemented in question (5). Set the regularisation coefficient to 0.01, 0.1, 1.0, 10.0, 100.0 and compare the train and test classification accuracy for each of the three classes. **(10 marks)**

3 Deadline and Submission Instructions

- Submit

  (a) the source code for all your programs,

(b) a README file (plain text) describing how to compile/run your code to produce the various results required by the assignment, and

(c) a PDF file providing the answers to the questions.

Compress all of the above files into a single tar ball (tgz) file and specify the filename as *studentid.tgz* (replace “studentid” by your departmental student id). It is extremely important that you provide all the files described above and not just the source code! (If you are unable to create a tgz file then create a zip file)

- Submission is via the departmental submission system accessible from `http://intranet.csc.liv.ac.uk/cgi-bin/submit.pl?module=COMP527`