COMP 527 - 2019 - CA1 Re-sit Assignment
Data Classification
Implementing k-NN classifier

Assessment Information

Assignment Number	1 (of 2 Re-sit)
Weighting	12%
Assignment Circulated	6th July 2019
Deadline	31st July 2019, 15:00 UK Time (BST)
Submission Mode	By email to danushka@liverpool.ac.uk
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 Marking criteria	This assignment assess the understanding of k nearest neighbour classification algorithm. 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 Overall marking scheme

The resit coursework for COMP527 consists of two assignments. The contribution of each assignment towards the final mark is as follows

Assignment 1	12%
Assignment 2	13%
TOTAL	25%

2 Objectives

This assignment requires you to implement a sentiment classifier using k-nearest neighbour (kNN) algorithm using Python programming language.

> **Note that** ***no credit will be given for implementing any other types of classification algorithms or using an existing library for the kNN instead of implementing it by yourself. You must provide a README file describing how to run your code to produce the results. Programs that do not run will result in a mark of zero!***

3 Sentiment Classification using kNN

In binary sentiment classification, the goal is to classify a given user review about a product as to whether the review expresses a positive or a negative sentiment about the product. We encounter such reviews in numerous online shopping sites such as Amazon or eBay. If we can *automatically* predict the sentiment of a review, then we can group reviews into positive and negative ones and read only a subset of all the reviews.

4 Assignment

Download and the file `http://danushka.net/lect/dm/resit/resit-CA1data.zip` and decompress it. Inside, you will find four files: *train.positive*, *train.negative*, *test.positive*, and *test.negative.* These files correspond to the positive and negative train/test reviews we will be using in this assignment. Each line in each file represents a review using a set of features. We will be using both unigram and bigram (concatenated using two underscores) features to represent a review. A review is represented using a bag-of-features. Moreover, each feature is counted only once, giving a boolean valued feature representation (i.e. a set of features for each review).

Questions/Tasks

(1) Write a program to load the train/test instances (positive/negative) from the train/test files. (**10 marks**)

(2) Implement a kNN classifier and measure the classification accuracy on the test instances. Classification accuracy is defined as the percentage of the total number of correctly classified instances to the total number of test instances. (**50 marks**)

(3) Vary the value of k and evaluate the performance of your kNN classifier. Plot your results in a graph where the x-axis corresponds to the value of k and the y-axis corresponds to the classification accuracy. What trends can be observed from the graph? Briefly report your findings. (**20 marks**)

(4) For measuring the similarity for computing the neighbourhood in your kNN classifier try different similarity/distance measures such as a) cosine similarity, b) Euclidean distance, and c) Manhattan distance. Compare the performance of the kNN classifier with these three measures.

(You may use additional similarity/distance measures other than the (a), (b), and (c) listed above.) Briefly report your findings. (**10 marks**)

(5) Using different sub-samples of positive vs. negative training instances, evaluate the robustness of the kNN classifier under unbalanced training datasets. Briefly report your findings. (**10 marks**)

5 Deadline and Submission Instructions

- Deadline for submitting the first assignment is **31st July 2019, 15:00 UK time (BST)**.
- 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 and graphs for the questions [2], [3], and [4].

  Compress all of the above files into a single tar ball (tgz) file and specify the filename as *studentid.tgz*, where "studentid" is your student ID number. It is extremely important that you provide all the files described above and not just the source code!

- Submission is by Email to danushka@liverpool.ac.uk