COMP5318 - Machine Learning and Data Mining

Assignment 1

Due: 09 May 2016, 11:59PM

1 Data set description

The dataset is collected from the Apps Market. There are four main files:

1. training_data.csv:
   - There are 20,104 rows; each row corresponds to an app.
   - For each row, each column is separated by comma (,). The first column is the app's name, with the remaining columns containing the tf-idf values. The tf-idf values are extracted from words in the description of each app. We have done some pre-processing steps which resulted in 13,626 unique words. If a word is found in the description of an app, it has a tf-idf value (the tf-idf value is not zero). On the other hand, its tf-idf value is equal to zero if the word is not found in the description of the app. More information about tf-idf could be found in http://en.wikipedia.org/wiki/Tf%E2%80%93idf
   - In summary, data_train.txt is a matrix with dimension: 20,104 $\times$ 13,627 (remember the first column is the app's name).
2. training_desc.csv:
   - There are 20,104 rows; each row is for an app.
   - For each row, each column is separated by comma (,). The first column is the app's name and the second column contains the app's description.
3. training_labels.csv:
   - There are 20,104 rows; each row is for an app.
   - For each row, each column is separated by comma (,). The first column is the app's name and the second column is for the label.

- There are 30 unique labels in total, for example Casual, Health and Fitness, etc.

Note that it is not necessary that the same rows of two training files refer to the same app. Please use the app's name as a reference.

4. test_data.csv:

   - This is a subset of the original data set; we have split the original data set into 90% for training set and 10% for test set (per label). This file should NOT be used for training the classifier.
   - Your code must be able to read the test set, and output a file "predicted_labels.csv" in the same data-format as "training_labels.csv". Make sure the predictions (classification results for the test set) are in the same order as test inputs, i.e. the first row of "predicted_labels.csv" corresponds to the first row of "test_data.csv" and so on).
   - The score will be based on how accurate your approach is. We will collect "predicted_labels.csv" and compare it to the actual labels to get the accuracy of your approach. For further testing purposes, we may use a different test set while grading.

2 Task description

Each group consists of up to 3 students. Your task is to determine / build classifier for the given data set and write a report. The score allocation is as follows:

- Classifier: max 20 points
- Report: max 80 points

Please see section 4 for the detailed marking scheme. The report and the code are to be submitted to eLearning by the due date.

2.1 Programming languages and libraries

You are allowed to use one of the following languages:
Python, Cython, Matlab, R, C/C++ or Java.

However, all are encouraged to use Python3. Although you are allowed to use external libraries for optimization and linear algebraic calculations, you are NOT

allowed to use external libraries for basic pre-processing and classification. For instance, you are allowed to use scipy.optimize for gradient descent or scipy.linalg.svd for matrix decomposition. However, you are NOT allowed to use sklearn.svm for classification (i.e. you have to implement the classifier yourself, if required). If you have any ambiguity whether you can use a particular library or a function, please post on edstem under the "Assignment 1" thread.

2.2 Performance evaluation

We expect you to have a rigorous performance evaluation and a discussion. To provide an estimate of the performance (precision, recall, F-measure, etc.) of your classifier in the report, you can perform a 10-fold cross validation on the training set provided and average the metrics for each fold.

3 Instructions to hand in the assignment

1. Go to eLearning and upload the following files/folders compressed together as a zip file.

   (a) report (a pdf file)
   The report should include each member's details (student ID and name).

   (b) code (a folder)

   i. algorithm (a sub-folder)
   Your code (could be multiple files or a project E.g. a PyCharm project)

   ii. input (a sub-folder)
   Empty
   Although "training_data.csv", "training_desc.csv", "training_labels.csv" and 'test_data.csv" should be inside the input folder, please do not include these four files in the zip file as they are over 30 MB. We will copy these four files to the input folder when we test the code.

   iii. output (a sub-folder)
   "predicted_labels.csv" - This file must be in the output folder. We will use this file for grading.

   If you work as a group, only one student needs to submit the zip file which must be named as student ID numbers of all group members separated by underscores. E.g. "xxxxxxxx_xxxxxxxx_xxxxxxxx_.zip".

2. Your submission should include the report and the code. A plagiarism checker will be used. Clearly provide instructions on how to run your code in the appendix of the report.

3. The report must clearly show (i) details of your classifier, (ii) the results from your classifier, including precision and recall results on the training data, (iii) run-time, and (iv) hardware and software specifications of the computer that you used for performance evaluations.

4. There is no special format to follow for the report but please make it as clear as possible and similar to a research paper.

5. A penalty of MINUS 1 (one) points per each day after the due date. Maximum delay is 7 (seven) days, after that assignments will not be accepted.

6. Remember, the due date to submit them on eLearning is 09 May 2016, 11:59PM.

4 Marking scheme

Category	Criterion	Marks	Comments
Report [80]	Introduction [5] •What is the aim of the study? •Why is this study important? Methods [20] •Pre-processing (if any) •Classifier Experiments and results [25] •Accuracy •Extensive analysis Discussion [10] •Meaningful and relevant personal reflection Conclusions and future work [5] •Meaningful conclusions based on results •Meaningful future work suggested Presentation [8] •Academic style, grammatical sentences, no spelling mistakes •Good structure and layout, consistent formating •Appropriate citation and referencing Other [7] •At the discretion of the marker: for impressing the marker, excelling expectation, etc. Examples include fast code, using LaTeX, etc.		
Marks [20]	•Code runs and classifies within a feasible time •Well organized, commented and documented		
Penalties [−]	•Badly written code: [−20] •Not including instructions on how to run your code: [−30] •Late submission: [−1] for each day late		

Note: Marks for each category is indicated in square brackets. The minimum mark for the assignment will be 0 (zero).