Web Science 2019 Exam

c.lioma

March 2019

This document contains three Web Science assignments. Collectively these count for 60% of the final grade. All three assignments are compulsory.

You must submit a single tar.gz or zip file that contains:

1. Your report in pdf including the latex sources or original Word document
2. The source code that you developed including any appendices describing how to run your code for each subpart of the assignment.

Do not include the datasets or any subset of them.

The tar.gz or zip file must be uploaded to Absalon before the end of day, CET, Friday 5 April. No late submissions are accepted and will count as a used attempt at completing the exam.

1 WWW as a network and challenges (10% of the final grade)

What are the challenges of processing web data? (1 page maximum)

2 Recommender systems (25% of the final grade)

You are given a small version of the MovieLens Dataset. Your task is to implement and evaluate recommender systems algorithms that can predict user movie ratings. This section provides details on how to do this.

To help you in solving this project, we strongly recommend that you keep "Collaborative Filtering Recommender Systems" by Ekstrand, Riedl and Konstan at hand. You can find it here: `http://files.grouplens.org/papers/FnT%20CF%20Recsys%20Survey.pdf`. You can also read the paper written about the dataset, which can be found here: `http://files.grouplens.org/papers/harper-tiis2015.pdf`

Download the small version of the dataset from `http://files.grouplens.org/datasets/movielens/ml-latest-small.zip`. The dataset contains 100,000 ratings and 3,600 tag applications applied to 9,000 movies by 600 users. Read the README on the zip file you downloaded so you understand the format. We expect you to implement the following tasks:

1. **Collaborative movie recommender**: The goal of this task is to implement a collaborative movie recommender using either a memory-based or a model-based approach

2. **Content-based movie recommender**: The goal of this task is to implement a content-based movie recommender by exploiting movie metadata (included in the zip file you have downloaded).

For both tasks, you should evaluate your recommender system using x-fold cross-validation. X-fold cross-validation requires that you split the dataset randomly into x folds, and use some of them for training and some of them for testing. The evaluation metric you should use in your cross-validation is the Root Mean Squared Error (RMSE). The RMSE score, commonly used in the recommender systems literature, measures accuracy by penalizing prediction errors according to:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

You must answer all of the following questions. Make sure you substantiate your writing with results from the recommender system, whenever possible.

Question 1 As discussed in class, many factors can impact the collaborative recommendations quality, including similarity measure, rating aggregation, choices of how you normalize your data, and so on. You should try different instantiations of your collaborative movie recommender and explain its variation in performance.

Question 2 As discussed in class, there are many factors that can impact the content-based recommendations quality including choices for content representation (e.g., unigrams, n-grams, concepts, named entities, latent topics), user profiling, etc. You should try different instantiations of your content-based movie recommender and explain its variation in performance.

Question 3 Now that you have implemented both recommendations approaches collaborative and content-based, compare the results between both recommender systems. Which one performed better? Could you combine both recommendations? Discuss.

Question 4 Whenever a new item/user enters the database on top of which the recommender system runs, it has no ratings/historical preference and therefore it is difficult to recommend. Discuss how a recommender system could alleviate this problem. Split your data in a way that you have items/users that were not previously seen in your training and then implement a solution that makes recommendations for them.

Question 5 The MovieLens Dataset is available in other variants that are much larger than the one used here. Does your solution scale to large datasets? Why or why not? Discuss.

3 Sentiment and data mining (25% of the final grade)

Sentiment analysis is the task of automatically determining if a given text have a positive or negative sentiment. This can be done by having an existing corpora of text which are labelled with the sentiment for each text, and training a classifier. Here you will have to train a classifier which can predict the sentiment of a text, and carry out an analysis of your classifier.

In this task we will use the IMDB sentiment dataset (Maas et al. 2011, `https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset`). This dataset consists of 50,000 movie reviews which have been labelled as either positive or negative. The dataset also consists 50,000 unlabelled movie reviews, which we do not require you to use, but you are encouraged to do so. The dataset is already split into a train and test set which you should use. You now have to solve the following tasks:

Task 1 Feature Extraction. Compute a feature representation for each review in the train and test set. For this project, we use only bigram counts as features. In order to compute the features, first tokenize the reviews, i.e. split the text into separate tokens. Then, compute a vocabulary containing all the unique bigrams (two subsequent tokens) that occur in any of the reviews. Then, for each review, compute how often any bigram in the vocabulary occurs in the review text. By the end of this step, each review should be represented by a feature vector that has n dimensions, where n is the size of the bigram vocabulary. If you want to include any additional features into your feature representations, you are encouraged to do so.

Task 2 Classification and Evaluation of Classifier Performance. Train a statistical classifier of your choice on the feature representations of reviews in the training set. You could for example use any of the classifiers in the python sklearn library. Make predictions for the feature representations of reviews in the test set. Evaluate the performance of the classifier on the test set by computing the F1-score.

Task 3 Error Analysis. Pick at least five reviews that are incorrectly classified by the classifier. Inspect these misclassified reviews and for each of them, report if you can reconstruct why the classifier made a mistake for this review.

When designing your prediction system you need to consider and describe any preprocessing steps you may wish to perform, as preprocessing of the data can affect the performance of your models. Some examples of preprocessing are

stop word removal, stemming, and text lower-casing. Take these into consideration when designing the project. Whatever design decisions you take with regards to preprocessing, cross validation, or statistical model choice justify them in the report with reasons that supports your design.