CMPT 741 - Data Mining
Project Report

Walther Maciel - 301278740
Course Based M.Sc.

December 4, 2015

1 Introduction

For a proper introduction please refer to the assignment question document available at: www.com

2 Methodology

The Methodology for providing recommendations to which words are missing from the documents consists of three basic steps: clustering, frequent pattern mining and association rule matching.

2.a Clustering

The first step in the whole process is the partitioning of the dataset into 4 clusters. Each one of them corresponds to one of the four mini assignments used to create the dataset.

It is assumed that documents for the same assignment share a higher number of words when compared to documents from other assignments. Consequently the pattern mining inside these clusters should yield more meaningful results, allowing for a more accurate classification.

In order to accomplish a good performing clustering, the K-means method presented by Hartigan *et al.* [4] was used. As K-means tend to be very sensitive to the start condition, the algorithm was ran a hundred times and a majority vote was taken.

This process was accomplished using and R script, which easily allows for the use of various clustering algorithms using the function `kmeans`. The resulting clustering was tested against the validation software, `Accuracy.jar`, provided by the instructor, which accused an accuracy of approximately 95%

After acquiring a satisfactory clustering result, the input file `DocumentWords.txt` was split four ways, where each row of the original was copied to a file representing the cluster in which that row was determined to belong to. Each one of these files was subject to its own pattern mining and association rule matching processes.

2.b Frequent Pattern Mining

Documents who share many words are more likely to share other words. In order to be able to efficiently find these commonly shared words, we can consider the `DocumentWords.txt` as a transactional database, where each document is a transaction, and the words they contain are the items.

The task of finding these frequent item sets was accomplished with the use of an open source Python library originally written by Dagenais [3] called pymining. The algorithm used is *Recursive Elimination*, by Borgelt *et al.* [2] [1].

Each cluster file was mined with the minimum support for a frequent item set defined as 20.

2.c Association Rules Matching

The association rules was generated based on the frequent patterns found. This process was also generated by a python script using pymining [3]. The matching of the rules to the transactions, however, was written by me as a python script.

Let every association rule r be $r : A \rightarrow B$, where A is the set of items that lead to the second set of items B. Then the check to see if a rule r matches a transaction t, we verify if $(A \subseteq t) \wedge (B \cap t = \emptyset)$.

For each transaction, the matching rules are sorted in descending order of confidence. And the predicted items are selected from these rules in order, until a total of five items are selected.

Finally, the predictions for each cluster were concatenated in a single file so they could be easily read by `Validation.jar`.

3 Results

The results obtained by the aforementioned methodology were compared against a validation set of answers by a software provided by the instructor called `Validation.jar`, which outputed a score of approximately 34%. The table below lists the most important parameters passed to the various functions throughout the methodology in order to ensure the repeatability of the experiment.

Function	Parameter	Value
K-means	`n.clusters`	4
K-means	`n.random starts`	100
Pattern Mining	`min.support`	20
Association Rules	`min.support`	10
Association Rules	`min.confidence`	0.45

Final result	
Average MAP@5	0.3449425287356322

References

[1] Christian Borgelt. Keeping things simple: Finding frequent item sets by recursive elimination. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pages 66–70. ACM, 2005.

[2] Christian Borgelt. Simple algorithms for frequent item set mining. In *Advances in machine learning II*, pages 351–369. Springer, 2010.

[3] Barthelemy Dagenais. pymining. `https://github.com/bartdag/pymining`, 2010.

[4] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.