DATA MINING PROJECT REPORT

Vishal Vijayvargiya Thesis Based

Overview

To solve the task of predicting words for documents, the approach followed involves clustering the documents into 4 clusters and then finding frequent patterns in those clusters to predict words in documents belonging to corresponding cluster. Benefit of this approach lies in the fact that we could generate high accuracy clusters (especially using Collaboration.txt, explained in later half) and then use frequent patterns on this clusters to get list of words which we could use for prediction. Rest of the document explains the process and results for two major tasks towards prediction.

Step 1 (Clustering)

We notice that total number of documents is 301 and total number vocabulary words is 7208. If we perform an analysis on the frequency of words in the documents, one will notice that many of the words (more than half) occur in very small number of documents, i.e. number of words which occur only in 1 or 2 documents is very high. Also there are large number of words which occur in more than 100 documents.

Analysis: We know that each cluster will have approximately same size (need not be exactly same) anywhere between 65-85. The words which are occurring in less number of documents say (0-40) and the words which are occurring in large number of documents say (100+), will not be helpful in clustering, in fact they could degrade the cluster quality. Also we know that there are some keywords in each assignment set which will occur almost in all documents belonging to that assignment set (and possibly less number of times in essays belonging to other assignment set). Thus a good cluster could be achieved by picking right words for unsupervised learning. *We pick the words which occur in at least 60 documents and no more than 90 documents.* After examining various combination of upper bound and lower bound 60-90 provided best results. Size of clusters is also a good predictor of quality of clusters. With 60-90 bound, we got clusters each with size within range of 23%-27%

Preprocessing

We preprocess the data to feed it into clustering algorithms. We create a file with a row corresponding to a particular document and for all the words which qualified for 60-90 range we add a column if that word is present in this document or not by using nominal values: “Yes” or “No”. We pass this file to WEKA for clustering.

Experiments and Results

Clustering algorithm used: **EM** Clustering (*reason: because it works good for categorical/nominal data)*

Tool used: **WEKA, Eclipse** (java code for preprocessing)

Cluster size created by EM and improved using information of Collaboration.txt: Cluster 0 (**80**), Cluster 1 (**73**), Cluster 2 (**73**), Cluster 3 (**75**)

Number of attributes(words) used: **53**

Accuracy obtained on 20% ground truth validation: **98.3%**

Analysis

Formed clusters have high accuracy mainly obtained by pruning few attributes and using the information from Collaboration.txt (how it is used is explained in last section)

Step 2 (Predictions)

The clusters obtained from previous step are used for prediction purpose. We create files for each cluster having documents belonging to that cluster and words corresponding to that document. These 4 files thus created are then passed to SPMF tool in which we run FP Growth frequent pattern mining algorithm on each file. Output is the frequent patterns in the cluster which we sort later on the basis of support(using Microsoft Excel). We create a java program which then for each document goes through the single length patterns(words) in corresponding cluster and checks if this word is present in document or not. If it is not present we output it as missing word. We do this until we have 5 such words.

Experiments and Results

Algorithm used: **FP Growth Frequent Pattern Mining** (*reason: because it is fast)*

Tool used: **SPMF** (open source data mining library), **Eclipse** (java code for preprocessing and printing to output file), **Microsoft Excel** (manipulation on data like sorting, replacing etc)

Minimum support threshold: **0.08**

Accuracy obtained on 20% ground truth validation: **32.9%**

Analysis

Frequent patterns provide reasonable way to predict words, since if a particular word is occurring many times in other documents of a cluster, it should then have high chances of appearing in the document under consideration. To verify the results the 20% ground truth validation program was used. Cross validation was used so that the results don't overfit the test set. Cross validation helps in ensuring that the results are not biased towards the test data. Also the accuracy obtained which might look less is infact due to the MAP technique which is very harsh if the output ordering has mistakes. Also frequent patterns provided better results than association rule mining. Association rules (with at least 40% confidence and 20% support) gave prediction results of around 28.9%. This may be because of considering only length 2 association rules.

Role of Collaboration.txt in this project (Thesis Based)

As mentioned in overview that the approach used involves clustering and then using those clusters for prediction. Collaboration.txt plays a major role in the whole process. It is used in clustering step to improve the accuracy of clusters we get.

How?

File contains information about students who collaborated together to submit assignment. Interesting facts which we could derive from this file are:

1) File helped in making predictions of cluster sizes (by using student and document info) and thus verify if we were getting correct clusters or not.
2) No student would have made multiple submissions for same assignment. Thus every cluster (which actually represents a mini-assignment) formed should contain no more than one document belonging to a student. Thus if doc-x and doc-y belong to student 's' and appear in same cluster, then at least one of them is assigned to wrong cluster.
3) Once we know which documents could possibly be wrong, we could assign them to correct clusters. This process could be done manually or using a java program. For example student 536 had 2 documents in cluster 2, so was the case with student 706. We see that we could resolve it as below:

536: Cluster 2 error		706: Cluster 2 error	
536,980,6741 -> Cluster 3		706,224,6890 -> Cluster 1	
536,876,1603 -> Cluster 2		706,985,7605 -> Cluster 2	
536,652,2449 -> Cluster 1		908,706,9441 -> Cluster 3	
536,706,5813 -> Cluster 2	(Corrected -> Cluster 0)	536,706,5813 -> Cluster 2	(Corrected -> Cluster 0)

Improvement after using collaboration.txt

Clusters accuracy were tested on the 20% validation data which were provided. Following accuracy were noted:

Accuracy of clusters without using collaboration.txt: **96.2%**

Accuracy of clusters after using collaboration.txt: **98.3%**

Overall improvement to prediction results:

Accuracy of prediction without using collaboration.txt: **31.1%**

Accuracy of prediction after using collaboration.txt: **32.9%**

References

1) Course slides.
2) Michael Steinbach, George Karypis, Vipin Kumar, *A Comparison of Document Clustering Techniques,* technical report, University of Minnesota.