Assignment 5, Lab 4

Association Rule Mining

Due : 11/27/13

Adithya Bhat(bhat.51@osu.edu)
Gaurav Ramesh(ramesh.48@osu.edu)

---

1. **Preprocessing for Mining Frequent Itemsets**

   1.1 Feature vector(<doci_d, [term_ids]>) from previous assignment, but with class ids also included as features, so new feature vector is of the form <doc_id, [term_ids],[class_ids]>

2. **Generating frequent itemsets and association rules**

   2.1 The tool we have used to find the association rules is "*Sequential Pattern Mining Framework(SPMF)*"[1]. We have used minimum **support ratio of 5% and minimum confidence of 60%.**

   2.2 The feature vector is fed into the above said application, which generates the frequent itemsets and association rules using the Apriori algorithm. The output from the tool is of the form:

```
[term_ids][class_ids] ==> [class_ids][term_ids] #Support X #Confidence Y
```

   This structure is transformed into convenient dictionary and ordered dictionary types of the form:

```
confidence => {
        rule_id => support
}
```

   2.2 Rules generated are pruned to just have rules of the form <[term_ids] => [class_ids]>, i.e all the rules that contain classes on the antecedent and items on the consequent side are removed

   2.3 The following graph is a plot of **classes vs number of documents in that class.** As you can see, most of the classes have very few documents, whereas a handful of them have huge number, so the class distribution is highly skewed. Running the Apriori algorithm once didn't give enough documents to cover all the classes, so we ran it in three stages, for classes having 0-100, 100-1000, > 1000 documents.

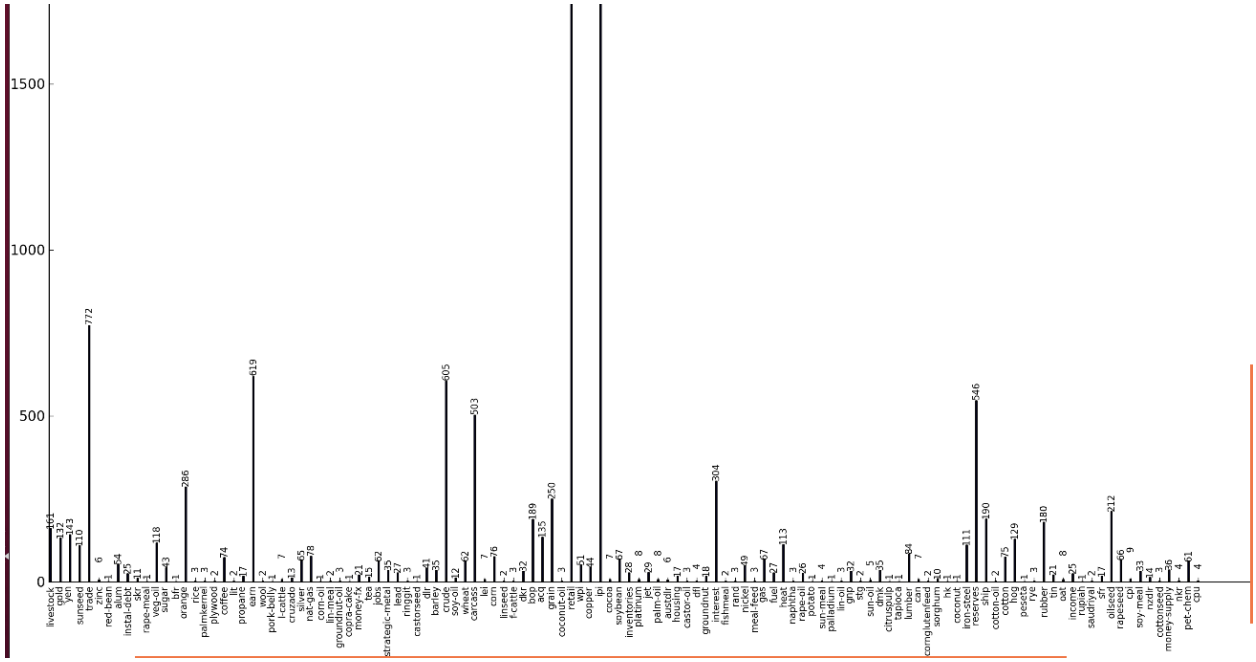

Fig 1
Class vs Number of Documents in the Class

2.4 Rules remaining from the previous step, are sorted, first by confidence, and then by support. Rules with same confidence and support are sorted in the order in which they were generated

2.5 Final rule set included rules generated from the above three runs

3. **Classification based on Association Rules**

3.1 We built the classifier based on the generated rules using *CBA-CB: Naive Algorithm*[2], according to the pseudo code given below :

```
1 R = sort(R);
2 for each rule r in R do
3      temp = NULL;
4      for each case d in D do
5             if d satisfies the conditions of r then
6                    store d.id in temp and mark r if it correctly classifies d;
7      end
8      if r is marked then
9             insert r at the end of C;
10            delete all the cases with the ids in temp from D;
11            selecting a default class for the current C;
12            compute the total number of errors of C;
13     end
14 end
15 Find the first rule p in C with the lowest total number of errors and drop all
```

the rules after p in C;
16 Add the default class associated with p to end of C, and return C (our classifier)

3.2 Accuracy of the classifier is readily obtained by the error count obtained from the previous step, as follows :

Acc = (|T| - |E|) * 100 / |T|,

where,
Acc -> accuracy of the model, in %
|T| -> number of documents considered for rule generation
|E| -> number of documents incorrectly classified

4. **Clustering and association rules generation**

4.1 Clustering of 80% of labeled documents(out of 11038) is first done using k-means algorithm, for 16 and 32 clusters

4.2 For each cluster, maximum/majority class of the cluster is assumed to be the label of the cluster

4.3 Frequent itemset generation and association rule mining is done as explained before, but this time, individually on each cluster

4.4 We have tested on the 20% of the labeled documents, with rules generated on the cluster, to which the test document belongs, based on its proximity with the cluster centroids

4.4 Accuracy of rules generated using a particular number of clusters is calculated from the total number of misclassified document count from each of the clustering exercise

5. **Visualizing output and conclusions**

5.1 Comparison of ***number of clusters vs accuracy graph*** is shown

# Clusters	**Support**	**Confidence**	**Accuracy**
8	5%	60%	65.2%
16	10%	60%	70%
150* *(without clustering)	5%	60%	62.43%

Table 1

5.2 Comparison of ***number of clusters vs time taken for rule-generation*** is shown

# Clusters	**Support**	**Confidence**	**Time Taken***
16	5%	60%	92+48 (140)
32	10%	60%	176+78 (254)
150* *(without clustering)	5%	60%	600

Table 2

*Time Taken is in seconds, clustering time + association rule generation time

References:

[1] SPMF, http://www.philippe-fournier-viger.com/spmf/
[2] Liu, Hsu, Ma. Integrating Classification and Association Rule Mining, *KDD-98, New York, Aug 27-31, 1998*