Danmarks Tekniske Universitet

Report 3

Course:
02450

Dataset: SPAM E-mail Database (Spam)

Author(s):
Claus Michael Oest Lensbøl (s132308)
Stephan Thordal Larsen (s146907)

DTU Compute
Institut for Matematik og Computer Science

02450
Morten Mørup

1 December 2015

Contents

1 Introduction

As a continuation of the first and second report for the course 02450, *Introduction to Machine Learning and Data Mining* at DTU, this document describes the process of performing clustering, association mining and outlier detection analysis of the SPAM dataset from `http://statweb.stanford.edu/~tibs/ElemStatLearn/`.

The report will describe, compare and discuss the results of two different methods for clustering, that is *Gaussian Mixture Models* and *Hierarchical Clustering*. Both will be evaluated on the purity of the clusters they generate.

For the association mining the *Apriori* algorithm is used to find similarities between the different attributes. Outlier detection is done with *Gaussian Kernal Density*, *K-Nearest Neighbour density*, *K-Nearest Neighbour average relative density* and *5th neighbour density*.

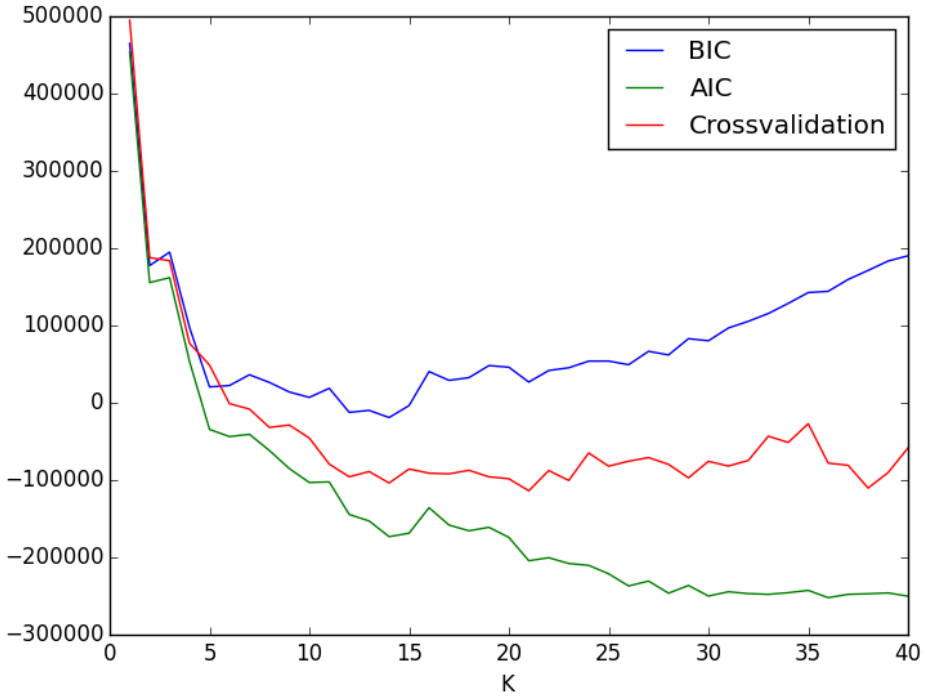


Figure 1: AIC, BIC and negative log likelihood results from running GMM with cross validation, from 0 to 40 clusters. One finds the optimal number of clusters is around 15.

2 Clustering

This section will cover clustering of the dataset using *Gaussian Mixture Models* and *Hierarchical Clustering*. The *Gaussian Mixture Model* will be run through an optimal number of clusters, which also will be used as cutoff for the *Hierarchical Clustering*. Finally the two will be briefly compared.

2.1 Gaussian Mixture Model

To find the ideal number of clusters for the Gaussian Mixture Model, cross validation was run from 1 to 40 clusters. All with full covariance, and each run with 10 different initializations, keeping the best result. The resulting *Bayesian information criterion*, *Akaike information criterion* and *negative log likelihood* can be found in Figure 1. Here one sees the ideal result, before *BIC* and *negative log likelihood* rises again, is around 15.

Running GMM with 15 clusters, one gets the clusters seen in Table 1. One finds that most of the data is located in one large cluster with low purity, and the rest is in smaller clusters with high purity.

By inspecting some of the cluster centroids, one sees what characterizes each cluster. Cluster 2 as an example has low occurrences of the words *remove*, *addresses* and *font*, but high occurrences of *you* and *address*. If one looks at a cluster mainly consisting of spam, cluster 5 as an example, one finds that most of the attributes are 0, but still has a very high count of capital letters, the characters *$, #, !* and the words *free*, *you* and *3d*.

Cluster	Spam	Valid	Purity
1	9	536	0.983
2	769	371	0.675
3	3	0	1.0
4	64	53	0.547
5	424	63	0.871
6	1	0	1.0
7	1	0	1.0
8	11	0	1.0
9	124	9	0.932
10	0	2	1.0
11	76	215	0.739
12	84	692	0.892
13	42	528	0.926
14	115	96	0.545
15	90	223	0.712

Table 1: Clusters from GMM with 15 clusters.

2.2 Hierarchical

A wide array of distance measures were tested, and all fell short on cluster purity. With cut-off set to 15 clusters, as found most effective in GMM, the data is split into many very small clusters with high purity (close to 1), and then one large cluster with a purity around 0.6. Some of the distance measures also resulted in medium sized clusters (around 100 nodes), with purity also around 0.6. This means that most of the data fits into one single cluster. This is the case with both single and complete linkage.

The chosen distance measure is *standardized euclidean* with single linkage, since it was found to have the highest purity in the small clusters, and around 0.6 in the single large cluster. The resulting clusters and purity can be found in Table 2.

The resulting dendrogram can be found in Figure 2, where one sees that most of the clusters don't have to far a distance from each other, and are therefore clustered together quite quickly.

2.3 Comparison and Evaluation

After having run both *Gaussian Mixture Models* and *Hiearchical* clustering, the *GMM* was found to perform best. *GMM* has overall lower purity in its clusters, but the clusters are larger, and can therefore better support future predictions. Although with high risk for errors in classification. The *Hierarchical* clustering split the data into many small clusters of size 1, and then one single cluster with the rest of the data, resulting in a rather useless clustering.

Cluster	Spam	Valid	Purity
1	1	0	1.0
2	1	0	1.0
3	0	3	1.0
4	1807	2778	0.606
5	0	1	1.0
6	0	1	1.0
7	1	0	1.0
8	0	1	1.0
9	0	1	1.0
10	0	1	1.0
11	0	1	1.0
12	1	0	1.0
13	1	0	1.0
14	0	1	1.0
15	1	0	1.0

Table 2: Clusters from hierarchical run with standardized euclidean distance measure and single linkage.

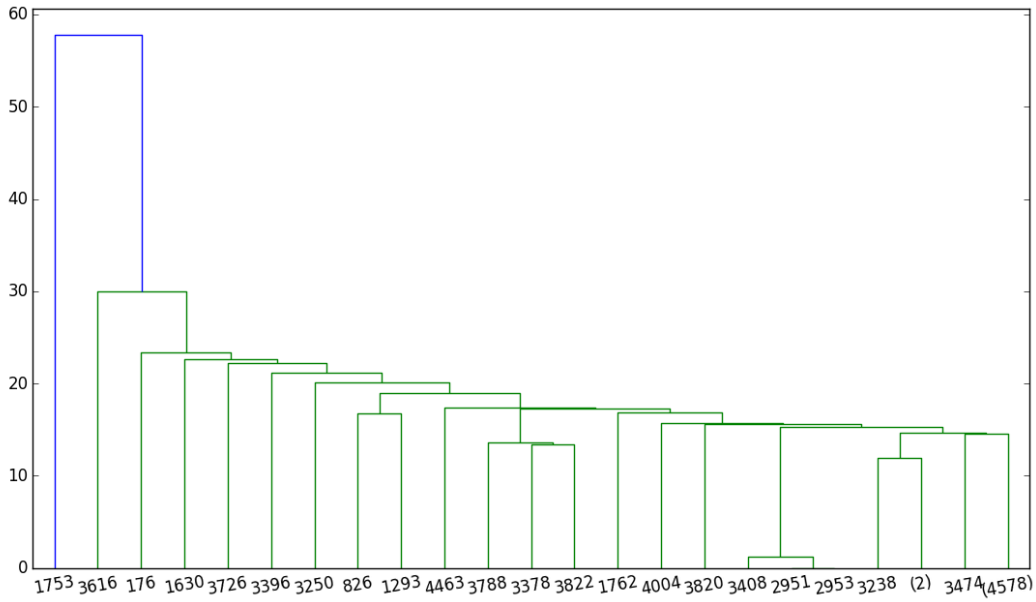


Figure 2: Dendrogram from hierarchical run with standardized euclidean distance measure and single linkage.

3 Association - Apriori

For doing association mining, the Apriori algorithm was used. The Apriori program required a specific format of binarized data and as the spam data is not binary, a binarization of this data was made.

3.1 Binarization method

One way of binarizing the data is to use the median as a pivot. Everything below the median gets set to 0 and everything above gets set to 1. This however presents a problem, as only high values will yield support and confidence, leaving low values *unnoticed*.

As we would like to be able to do associations based on low values as well, all attributes was instead binarized into two columns, one for values lower than the median and one for values higher than the median. The second column is equal to the methods described below, and the first column is a "mirror" as it yields of 1 for values lower than the median and 0 for higher values.

In the case of the spam dataset, this leaves us with 114 attributes with the suffixes *_low* and *_high*.

This method did however mostly show how the samples differ with showing low values of different attributes and not really giving a clear sign of anything else than the spread of attributes. The raw data from this is included in appendix A and is not discussed further.

To get some data on how the samples are alike, the method was changed to only look at high values (above median). To be able to spot rules that might be an indication of spam, the last attribute column denoting spam and not spam (1 and 0) is added to the matrix before the apriori algorithm is run.

3.2 Association results

Looking at the purely high values of attributes in the samples gave a better perspective on the samples. Running the apriori algorithm with a minimum support of 35%, a minimum confidence of 70%, and a max number of rules set to 4, yielded the item sets in table 3 with the *spam_indicator* underlined.

These items are mostly common words, but also has a fair bit of the capital letter assessment as part of them, along with different characters[1]. The only set containing more than one item, is made from the average run length of capital letters and th longest run of capital letters, indicating to no surprise that these occur frequently together.

The rules generated from the algorithm is shown in table 4 and have the *spam_indicator* underlined.

The most interesting of these are the ones containing the *spam_indicator* as these have a possibility of giving an indication as to if the sample is spam or not. These are filtered out in table 5 to give a better overview of these seven rules.

For the rules not containing the *spam_indicator* as a condition, a couple of conditions are made up of {char_freq_! ,word_freq_your} with one them being the only condition to

[1]The dataset only includes ({ and [as indicators. It would seem like double effort to match on both opening and closing braces.

Item	Support
capital_length_total	50%
word_freq_will	50%
word_freq_your	50%
capital_length_longest	50%
capital_length_average	50%
char_freq_!	50%
char_freq_(	50%
word_freq_you	50%
capital_length_longest capital_length_total	40%
capital_length_average capital_length_total	40%
capital_length_average capital_length_longest	40%
char_freq_! word_freq_your	40%
word_freq_all	40%
word_freq_you word_freq_your	40%
word_freq_our	40%
spam_indicator	40%

Table 3: Association items. The items consists of common words as well as indicators for capital letters and the characters ! and (.

the *spam_indicator*. Other implications of this condition are: *capital_length_longest*, *capital_length_total*, *capital_length_average* and *word_freq_you*. This could imply that this rules are indeed a good classifier for the spam.

Another set of recurring conditions are {capital_length_longest, capital_length_total} that implies *word_freq_will*, *word_freq_your*, *char_freq_!* and *capital_length_average*. None of these implications presents any obvious conclusions.

As mentioned before the attribute *spam_indicator* has a connection to the attributes *word_freq_your* and *char_freq_!* which is also apparent in table 5. As the support is somewhat lower than the amount of spam messages (40%), these attributes cannot be used as a sound indicator, but can give an indication as to if the sample is spam or not.

Implication	Condition	Support	Confidence
capital_length_longest	capital_length_average capital_length_total	30%	90%
capital_length_average	capital_length_longest capital_length_total	30%	90%
capital_length_total	capital_length_average capital_length_longest	30%	80%
capital_length_longest	char_freq_! word_freq_your	30%	80%
capital_length_total	char_freq_! word_freq_your	30%	80%
capital_length_longest	capital_length_total	40%	80%
capital_length_total	word_freq_all	30%	80%
capital_length_longest	capital_length_average	40%	80%
capital_length_average	capital_length_longest	40%	80%
word_freq_your	spam_indicator	30%	80%
word_freq_you	char_freq_! word_freq_your	30%	80%
capital_length_average	spam_indicator	30%	80%
char_freq_!	spam_indicator	30%	80%
spam_indicator	char_freq_! word_freq_your	30%	80%
capital_length_total	capital_length_longest	40%	80%
capital_length_total	word_freq_our	30%	80%
char_freq_!	word_freq_you word_freq_your	30%	70%
word_freq_will	capital_length_total	30%	70%
word_freq_will	word_freq_our	30%	70%
word_freq_your	word_freq_our	30%	70%
capital_length_total	spam_indicator	30%	70%
word_freq_your	word_freq_you	40%	70%
capital_length_longest	spam_indicator	30%	70%
word_freq_will	word_freq_you word_freq_your	20%	70%
capital_length_average	char_freq_! word_freq_your	20%	70%
word_freq_you	word_freq_your	40%	70%
word_freq_your	word_freq_all	30%	70%
word_freq_will	word_freq_all	30%	70%
char_freq_!	capital_length_average capital_length_total	30%	70%
char_freq_!	capital_length_longest capital_length_total	30%	70%
word_freq_your	char_freq_!	40%	70%
char_freq_!	word_freq_your	40%	70%
word_freq_your	capital_length_average capital_length_total	30%	70%
capital_length_total	capital_length_average	40%	70%
capital_length_average	capital_length_total	40%	70%
word_freq_your	capital_length_longest capital_length_total	30%	70%
word_freq_will	capital_length_longest capital_length_total	30%	70%
word_freq_you	spam_indicator	30%	70%

Table 4: Association rules. Non of these rules really stand out as they are common or part of the

Implication	Condition	Support	Confidence
word_freq_your	spam_indicator	30%	80%
capital_length_average	spam_indicator	30%	80%
char_freq_!	spam_indicator	30%	80%
spam_indicator	char_freq_! word_freq_your	30%	80%
capital_length_total	spam_indicator	30%	70%
capital_length_longest	spam_indicator	30%	70%
word_freq_you	spam_indicator	30%	70%

Table 5: Association rules including the spam indicator.

4 Outlier / Anomaly detection

Four methods of outlier detection was used on the data set. The four methods was performed with the data normalized as well as with the data not normalized. As some of the attributes might help identify outliers both tests were made.

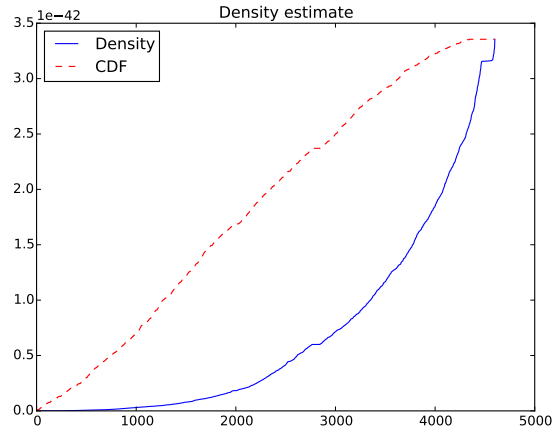
Figure 3a and 3b shows the Gaussian Kernel Density for the data.

Figure 4a and 4b shows the KNN density for the data.

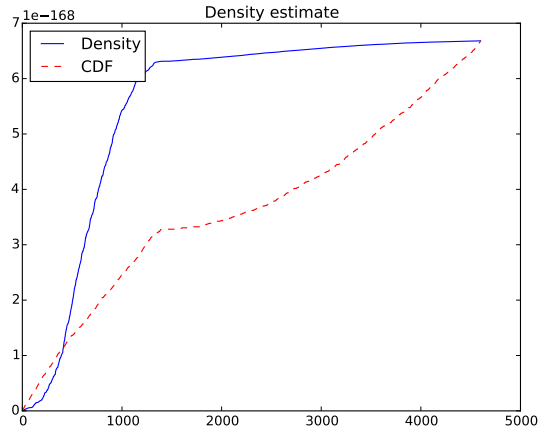
Figure 5a and 5b shows the KNN average relative density for the data.

Figure 6a and 6b shows the 5-N density for the data.

Please note that the CDF curves are made to sum to the maximum of the density rather than to 1.

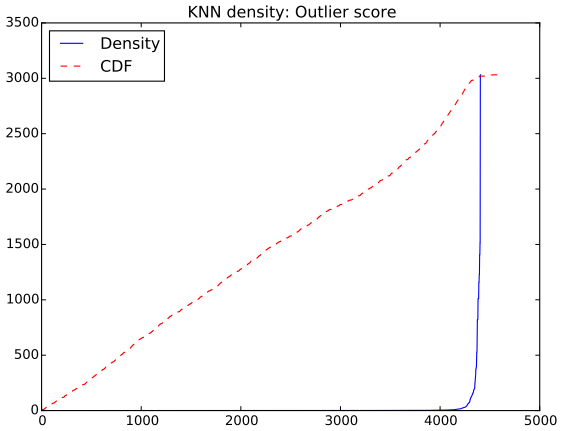


(a) Normalized before detection.

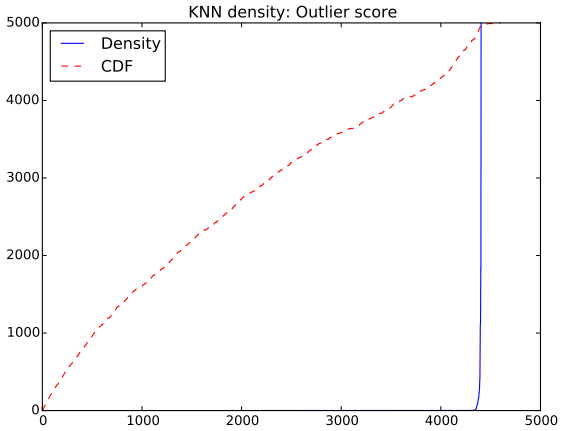


(b) Not normalized before detection.

Figure 3: Gaussian kernel density measurement.

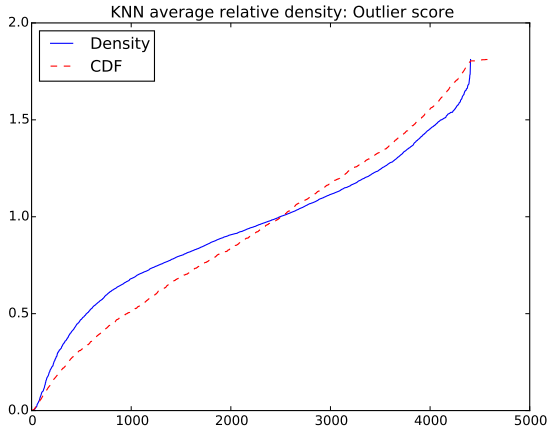


(a) Normalized before detection.

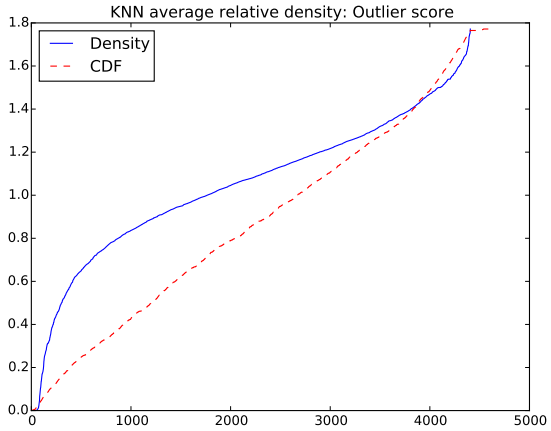


(b) Not normalized before detection.

Figure 4: KNN density.

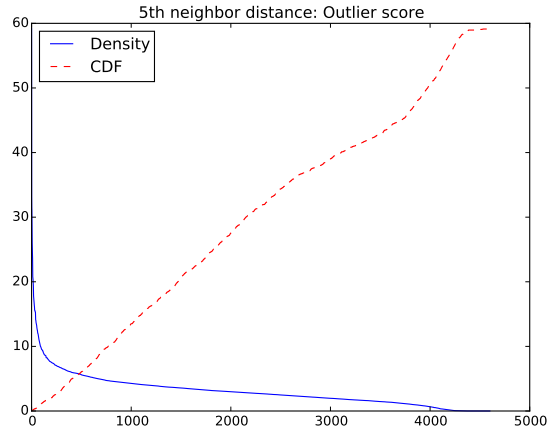


(a) Normalized before detection.

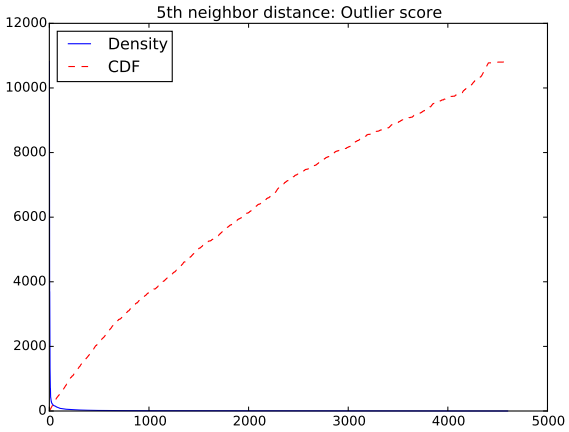


(b) Not normalized before detection.

Figure 5: KNN average relative density.



(a) Normalized before detection.



(b) Not normalized before detection.

Figure 6: 5-N Neighbor distance.

4.1 Possible outliers

For the Gaussian kernel density it seems that the "best trend" is found for the non-normalized data in figure 3b. There seems to be a collation between the low values of the density estimate, and spam mails. About 40% of the mail is supposed to be spam, and around 40% of the spam mail is accounted for before the "bend" in the curve.

For the KNN density, outliers do exist, but the numbers are far to few to carry any real data as to whether we can classify spam or not.

The same unfortunately goes for the average relative density and the 5-N distance as nothing comes close to the 40% spam we are looking to find.

5 Conclusion

Clustering was performed with 15 clusters, found as optimal amount by running cross validation on GMM. The clustering performed by GMM was found to be better than the hierarchical variant, since it spread the data more equally across the clusters. Better results from hierarchical clustering could probably be achieved by using another distance measure.

Using the Apriori algorithm for association mining gives two interesting associations, one with the *spam_indicator* as the implication. These associations are still not clear enough to use as a classification. When binarizing the data it only makes sense to look at high values of the attributes, as the dissimilarities between the samples are much higher than the similarities, and this dominates the dataset if included.

For outlier detection some outliers can using the Gaussian kernel density (without normalizing the data) account for 40% of the spam, but with the last 60% of the spam not considered outliers this would prove hard to use in production to find all the spam.

A High/low Associations

The parameters for Apriori algorithm was a minimum support of 95% and a minimum confidence of 95%. 2 rules as a maximum was allowed to limit the number of rules generated.

Below is a list of frequent item sets, note that all the items are based on low values of the attributes.

```
Item: word_freq_3d_low[Sup. 1e+02]
Item: word_freq_table_low word_freq_3d_low[Sup. 1e+02]
Item: word_freq_415_low word_freq_857_low[Sup. 1e+02]
Item: word_freq_857_low[Sup. 1e+02]
Item: word_freq_conference_low[Sup. 1e+02]
Item: word_freq_cs_low[Sup. 1e+02]
Item: word_freq_cs_low word_freq_parts_low[Sup. 1e+02]
Item: word_freq_cs_low word_freq_table_low[Sup. 1e+02]
Item: word_freq_cs_low word_freq_3d_low[Sup. 1e+02]
Item: word_freq_font_low[Sup. 1e+02]
Item: word_freq_font_low word_freq_parts_low[Sup. 1e+02]
Item: word_freq_font_low word_freq_table_low[Sup. 1e+02]
Item: word_freq_font_low word_freq_table_low word_freq_3d_low[Sup. 1e+02]
Item: word_freq_font_low word_freq_3d_low[Sup. 1e+02]
Item: word_freq_parts_low[Sup. 1e+02]
Item: word_freq_parts_low word_freq_table_low[Sup. 1e+02]
Item: word_freq_parts_low word_freq_table_low word_freq_3d_low[Sup. 1e+02]
Item: word_freq_parts_low word_freq_3d_low[Sup. 1e+02]
Item: word_freq_table_low[Sup. 1e+02]
Item: word_freq_415_low[Sup. 1e+02]
```

Below is a set of rules generated.

```
Rule: word_freq_3d_low <- [Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_857_low <- word_freq_font_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_415_low <- [Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_conference_low <- word_freq_857_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_857_low <- word_freq_conference_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_cs_low <- word_freq_857_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_857_low <- word_freq_cs_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_font_low <- word_freq_857_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_parts_low <- word_freq_857_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_table_low <- [Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_857_low <- word_freq_parts_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_table_low <- word_freq_857_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_857_low <- word_freq_table_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_3d_low <- word_freq_857_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_857_low <- word_freq_3d_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_857_low <- [Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_415_low <- word_freq_3d_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_3d_low <- word_freq_415_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_415_low <- word_freq_table_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_table_low <- word_freq_415_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_415_low <- word_freq_parts_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_parts_low <- word_freq_415_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_415_low <- word_freq_font_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_font_low <- word_freq_415_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_415_low <- word_freq_cs_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_cs_low <- word_freq_415_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_415_low <- word_freq_conference_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_conference_low <- word_freq_415_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_415_low <- word_freq_857_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_857_low <- word_freq_415_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_telnet_low <- word_freq_857_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_telnet_low <- word_freq_415_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_lab_low <- word_freq_857_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_cs_low <- word_freq_conference_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_conference_low <- word_freq_cs_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_font_low <- word_freq_conference_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_cs_low <- [Conf. 1e+02,Sup. 1e+02]
```

```
Rule: word_freq_table_low <- word_freq_3d_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_3d_low <- word_freq_table_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_parts_low <- [Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_parts_low <- word_freq_3d_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_3d_low <- word_freq_parts_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_parts_low <- word_freq_table_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_table_low <- word_freq_parts_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_font_low <- [Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_font_low <- word_freq_3d_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_3d_low <- word_freq_font_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_font_low <- word_freq_table_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_table_low <- word_freq_font_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_font_low <- word_freq_parts_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_parts_low <- word_freq_font_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_cs_low <- word_freq_3d_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_conference_low <- word_freq_font_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_3d_low <- word_freq_cs_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_cs_low <- word_freq_table_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_table_low <- word_freq_cs_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_cs_low <- word_freq_parts_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_parts_low <- word_freq_cs_low[Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_cs_low <- word_freq_font_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_font_low <- word_freq_cs_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_conference_low <- [Conf. 1e+02,Sup. 1e+02]
Rule: word_freq_conference_low <- word_freq_3d_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_3d_low <- word_freq_conference_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_conference_low <- word_freq_table_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_table_low <- word_freq_conference_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_conference_low <- word_freq_parts_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_parts_low <- word_freq_conference_low[Conf. 1e+02,Sup. 9e+01]
Rule: word_freq_lab_low <- word_freq_415_low[Conf. 1e+02,Sup. 9e+01]
```

One thing to notice from the item sets and rules generated is the complete lack of items and rules including the is *_high* suffix. This generally makes sense as the emails are more not alike than alike due to the many attributes. If the support or confidence were to have a lower threshold, items and rules would also be represented here.

In general, these rules and items are a sign of the emails being similar in *missing attribues*, but that is the most general that can be said.