M.Tech. Project **Final Report - (January-April 2014)**

Frequent Itemset Based Hierarchical Document Clustering and Incremental Outlier Detection for Dynamic Data Streams

Guide : Prof. Kamalakar Karlapalem

Submitted by : Vini Dixit (201205579)

*** This project is extended idea implementation of Paper - "Frequent Itemset Based Hierarchical Document Clustering Using Wikipedia as External Knowledge". The extension idea was first proposed in term paper of DWDM course in Monsoon 2013.

Index

1. Abstract

Base paper[1] majorly solves problems related to Clustering of documents with high dimensionalty as well as also dealt with semantic relationship between the words. It says, "High dimensionality is a major challenge in document clustering. Some of the recent algorithms address this problem by using frequent itemsets for clustering. But, most of these algorithms neglect the semantic relationship between the words. On the other hand there are algorithms that take care of the semantic relations between the words by making use of external knowledge contained in WordNet, Mesh, Wikipedia, etc but do not handle the high dimensionality. In this paper we present an efficient solution that addresses both these problems. We propose a hierarchical clustering algorithm using closed frequent itemsets that use Wikipedia as an external knowledge to enhance the document representation. We evaluate our methods based on F-Score on standard datasets and show our results to be better than existing approaches. "

Outliers are large deviate from others data points. They are often not the errors, and may carry important information. An outlier in present may become an important document(non-outlier) in future. So, a system which handles hierarchical documents supplied dynamically is required. This system not only

clusters these high dimensional documents by above efficient approach, but also handles the change and update of new clusters. These changes may be caused due to new documents or the already existing outliers, which are no longer outlier now. Obviously, it detects outlier too by an incremental approach.

The technique which is majorly used to add this feature is by constructing some additional datastructures which keep account of i) previously processed clusters and outliers in Associative Rule Warehouse(ARW) globally, and ii) new coming documents and its clusters locally in Transaction Sensitive Buffer(TSB). TSB is intentionally kept fixed in size to control flow of incoming requests at a time.

2. Introduction

A recent trend in clustering documents is the use of frequent itemsets. These methods handle the high dimensionality of the data by considering only the words which are frequent for clustering. A frequent itemset is a set of words which occur together frequently and are good candidates for clusters. Many algorithms in this category consider the entire set of frequent itemsets for clustering, which may lead to redundant clusters.

Most approaches performing document clustering do not consider the semantic relationship between the words. Thus if two documents talking about the same topic do that using different words (which may be synonyms), these algorithms can not find the similarity between them and may cluster them into two different clusters. A simple solution to this problem is to use if-idf scoring to find best fitting cluster for a document.

With the growth of information technology, data streams become a widespread state of data. For example, applications involving stream data abound and include network traffic monitoring, credit card fraud detection and stock market trend analysis. There will generate some outliers with the data streams coming, which are often not errors and always carry important information. Thus, it is very essential to mine outliers from these data streams.

3. Motivation

In the first phase of project, base paper was implemented on Reuters news dataset and results as well as the efficiency was coming good and expectedly. But, there were two problems - i) It was unable to find expected clusters when documents are supplied dynamically ii) Automatic outlier detection in dynamic data streams. When existing system was applied on dynamic data streams, it was giving redundant clusters and also absence of now known important clusters, which were outliers in previous pass.

A simple solution proposed is to provide some global data structures, which will keep an account of previous pass results. And also a set of rule to be applied on updated data(previous FIS and new FIS) to find out those fis which were outlier(s) before, but now due to increase in its support count wrt MDD (Match Difference Degree) rules are no longer outlier(s).

For implementing this approach, a separate system aka Incremental Outlier Detection Model(IODM) is proposed, which dynamically detects outlier and FIS in every pass. Also, it updates global databases according to rules described later.

4. Itemset-based Document Clustering problem

In this section, we formally describe the problem of itemset-based clustering of documents. In particular, we show that the dual problems of document clustering and topic detection are related very closely when seen in the context of frequent itemset mining.

Intuitively, the document clustering problem is to cluster text documents by using the idea that *similar documents share many common keywords*. Alternatively, the topic detection problem is to group related keywords together into meaningful topics using the idea that *similar keywords are present in the same documents*. Both of these problems are naturally solved by utilizing frequent itemset mining as follows.

For the first problem, keywords in documents are treated as items and the doc- uments (treated as sets of keywords) are analogous to transactions in a market- basket dataset. This forms a transaction space, that we refer to as **doc-space** as illustrated below, where the d_i are documents and w_{ij} are keywords.

$d_1 - [w_{11}, w_{21}, w_{31}, ...]$

$d_2 - [w_{12}, w_{32}, w_{42}, ...]$

....

Then, in this doc-space, frequent combinations of keywords (i.e., frequent item- sets) that are common to a group of documents convey that those documents are similar to each other and thereby help in defining clusters. e.g: if (a, b, c) is a frequent itemset of keywords, then (d_1, d_2, d_3) which are the documents that contain these keywords form a cluster.

For the second problem, documents themselves are treated as items and the keywords are analogous to transactions the set of documents that contain a keyword is the transaction for that keyword. This forms a transaction space, that we refer to as **topic space** as illustrated below, where the d_i are documents and w_i are keywords.

$w_{11} - [d_{i1}, d_{j1}, d_{k1}, ...]$

$w_{12} - [d_{i2}, d_{j2}, d_{k2}, ...]$

...

Then, in this topic-space frequent combinations of documents (i.e., frequent item- sets) that are common to a group of keywords convey that those keywords are similar to each other and thereby help in defining topics.

5. IODM Model and Rules

Match Difference Dregree(MDD)

We derive match difference degree using association rules to detect outlier transactions from a transaction data stream. The basic idea is that in a transaction T, in which some items are not observed even though they should occur in T according to the association rules. For example, an association rule $X \rightarrow Y$ with a high confidence, when X occurs in T, then Y occurs with high probability. That is, if X occurs in T, but Y does not occur, it is an indication that the transaction T is abnormal.

Rule1. Strong Match rules

An association rule is an implication of the form $X \rightarrow Y$, where X and Y both are items. T is an item set, we denote $X \rightarrow Y$ is a strong match rule for T, if and only if $X \epsilon T$, and $Y \epsilon T$.

This rule helps in updating old outliers to main clusters, which are strong now.

Rule2. Weak Match rules

An association rule is an implication of the form $X \rightarrow Y$, where X and Y both are items. T is an item set, we denote $X \rightarrow Y$ is a weak match rule for T, if and only if $X \epsilon T$, but $Y \epsilon T$.

This rules helps in identifying outliers.

Icremental Outlier Detection Model (IODM)

In this section, we briefly describe the design of IODM, a fully automated transactional outlier detecting model for transaction data streams. Then, we discuss in detail the design and imple- mentation of our IODM-algorithm and the outlier detecting process. The IODM is designed to perform the transaction data streams outlier detecting in three steps as shown in Fig. 1.

IODM first handles a large transactional data samples as the basic dataset. Standard sampling techniques are used to generate a sample dataset from the entire history of transaction database. We mine and maintain a large association rule set as the fundamental ARW using Apriori algorithm.

In the transactional outlier detecting step online, we give the TSB and the MDD_{min} . The window sliding phase is activated after the current TSB becomes full. A new incoming transaction is appended to the current sliding window, and the oldest transaction is removed from the buffer. For each transaction removed from the sliding window, we use the approach of MDD to match each item in the transaction with all association rules in the ARW and calculate the $MDD(T)$ by one scan the TSB. Then, IODM determines whether a transaction is outlier by $MDD(T)$. If $MDD(T) > MDD_{min}$, we define T is an outlier transaction. If it is not, we hold T as a normal transaction and storage it in the TSB.

Updating the ARW incrementally is the third step of the IODM model. When the transaction- sensitive buffer becomes full, we mine association rules from the transaction dataset which is collected by the TSB. Therefore, we get an association rule set. For each association rule R, if $R \epsilon ARW$, IODM will do nothing. And if $R \epsilon ARW$, we insert the R into ARW. The IODM repeats this operation for all arrived transaction and dynamically updates the ARW with the time and memory constrains. It corresponds to the performance of data stream that the latest data are always more valuable than history data. Hence, the IODM model can adapt to the evolution of transaction data streams.

IODM System Flow Diagram

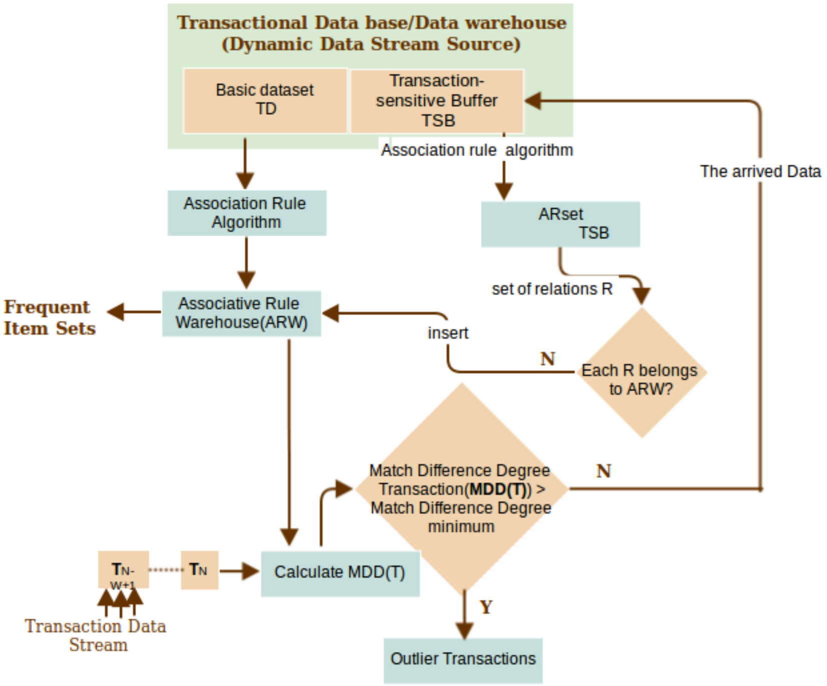
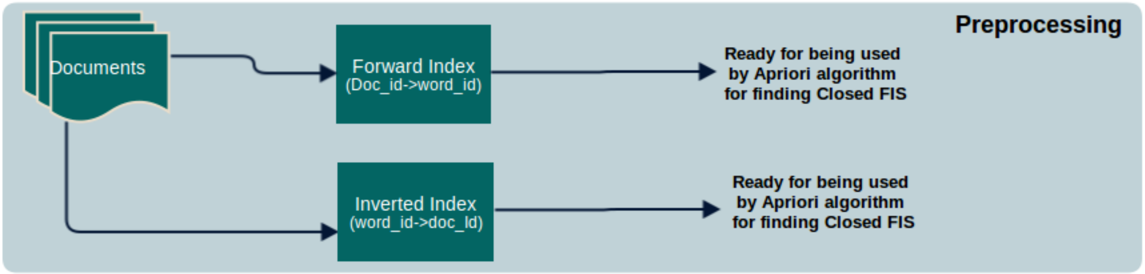

Transactional Data base/Data warehouse
(Dynamic Data Stream Source)
Basic dataset TD
Transaction-sensitive Buffer TSB
Association rule algorithm
The arrived Data
Association Rule Algorithm
ARset TSB
set of relations R
Frequent Item Sets
Associative Rule Warehouse(ARW)
insert
N
Each R belongs to ARW?
Match Difference Degree Transaction(MDD(T)) > Match Difference Degree minimum
N
TN-W+1
TN
Calculate MDD(T)
Y
Transaction Data Stream
Outlier Transactions

Fig : Flow Diagram for finding Automatic Incremental Frequent Itemsets and Outliers from Realtime Dynamic Data Streams

6. System Flow Diagram and Class Interactions


Preprocessing
Documents
Forward Index
(Doc_id->word_id)
Ready for being used by Apriori algorithm for finding Closed FIS
Inverted Index
(word_id->doc_Id)
Ready for being used by Apriori algorithm for finding Closed FIS

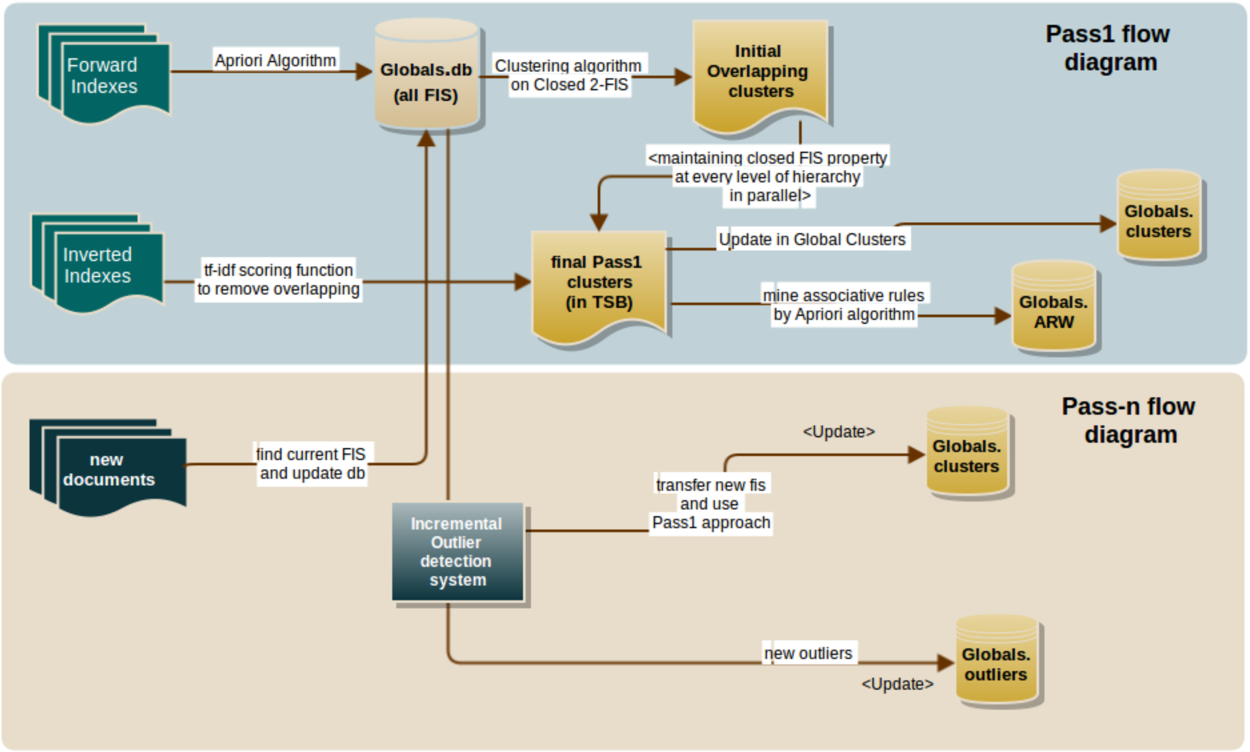
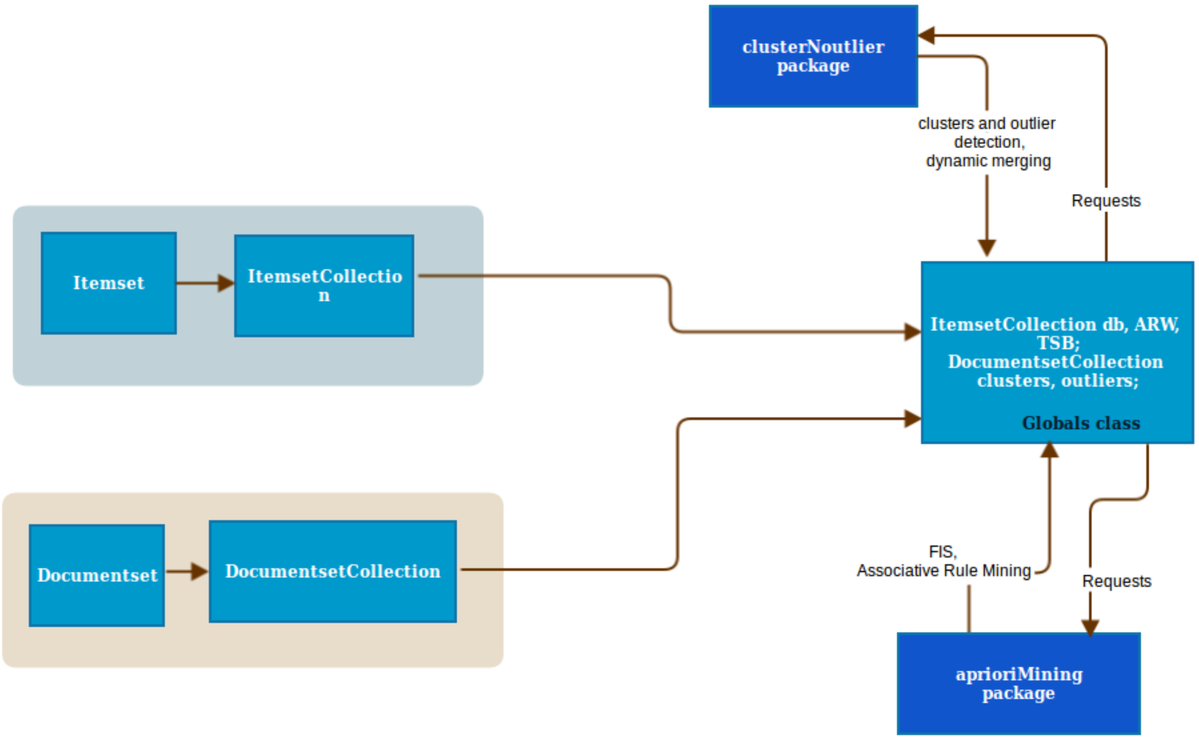

Pass1 flow diagram
Forward Indexes
Apriori Algorithm
Globals.db (all FIS)
Clustering algorithm on Closed 2-FIS
Initial Overlapping clusters
<maintaining closed FIS property at every level of hierarchy in parallel>
Globals. clusters
Update in Global Clusters
Inverted Indexes
tf-idf scoring function to remove overlapping
final Pass1 clusters (in TSB)
mine associative rules by Apriori algorithm
Globals. ARW
Pass-n flow diagram
new documents
find current FIS and update db
<Update>
Globals. clusters
transfer new fis and use Pass1 approach
Incremental Outlier detection system
new outliers
Globals. outliers
<Update>

Fig. Complete system Flow diagram

Class Interactions


clusterNoutlier package
clusters and outlier detection, dynamic merging
Requests
Itemset
ItemsetCollection
ItemsetCollection db, ARW, TSB; DocumentsetCollection clusters, outliers;
Globals class
Documentset
DocumentsetCollection
FIS, Associative Rule Mining
Requests
aprioriMining package

Class Interaction Diagram of system

7. Results, Observations and Analysis

Pattern :

[FIS] → [set of documents] ⇒ (senses/ meanings/context) → {set of documents, forming a cluster}

The documents which are found corresponding to a particular FIS, are kept in 1 cluster. The documents which are not shown in any of these clusters are taken as outliers, as they could not satisfy Strong Match degree with any of the strongly occuring itemset.

Pass 1

Initial Clusters obtained : for 1000 documents count - 4

[**46, 131**] → [5, 19, 180, 235, 241, 253, 254, 395, 425, 501, 742, 856, 874, 1057, 1067, 1069, 1094, 1185, 1257, 1295, 1377, 1388, 1396, 1405, 1570, 1590, 1640, 1674, 1777, 1792, 1901, 1907, 1970]

[**82, 117**] → [6, 69, 200, 229, 297, 303, 313, 833, 855, 1394, 1405, 1582, 1792, 1882]

[**56, 74**] → [221, 296, 307, 336, 341, 411, 475, 519, 522, 913, 926, 942, 943, 953, 999, 1008, 1112, 1154, 1421, 1503, 1553, 1560, 1724, 1931, 1943, 1967, 1976, 1982]

[**20, 46**] → [5, 57, 97, 193, 197, 235, 327, 395, 501, 516, 694, 855, 934, 1131, 1215, 1268, 1273, 1369, 1395, 1399, 1582, 1652, 1777, 1843, 1952]

After Dynamic Merging (if required): count - 3

[**82, 117**] → [6, 69, 200, 229, 297, 303, 313, 833, 855, 1394, 1405, 1582, 1792, 1882]

[**56, 74**] → [221, 296, 307, 336, 341, 411, 475, 519, 522, 913, 926, 942, 943, 953, 999, 1008, 1112, 1154, 1421, 1503, 1553, 1560, 1724, 1931, 1943, 1967, 1976, 1982]

[**20, 46, 131**] → [5, 19, 180, 235, 241, 253, 254, 395, 425, 501, 742, 856, 874, 1057, 1067, 1069, 1094, 1185, 1257, 1295, 1377, 1388, 1396, 1405, 1570, 1590, 1640, 1674, 1777, 1792, 1901, 1907, 1970]

Observation 1 : [20,46] and [46,131] FIS are having 1 item in common. So, according to rules and MDD threshold(taken as 1 item), documents of both shouls share same clusters. **This can be extended to more FIS sizes and corresponding larger MDD threshold to get improved clusters.**

Pass 2

Initial Clusters obtained : for 2000 documents, after updating in initial clusters set. count - 5

[**85, 162**] → [208, 223, 287, 323, 333, 342, 419, 676, 753, 759, 872, 888, 893, 894, 914, 991, 1306, 1347, 1385, 1392, 1409, 1499, 1553, 1579, 1605, 1656, 1675, 1697, 1742, 1772, 1796, 1827, 1911, 1926, 1929, 1936, 1951]

[**82, 117**] → [6, 69, 200, 229, 297, 303, 313, 833, 855, 1394, 1405, 1582, 1792, 1882]

[**56, 74**] → [221, 296, 307, 336, 341, 411, 475, 519, 522, 913, 926, 942, 943, 953, 999, 1008, 1112, 1154, 1421, 1503, 1560, 1724, 1931, 1943, 1967, 1976, 1982]

[**20, 46, 131**] → [5, 19, 180, 235, 241, 253, 254, 395, 425, 501, 742, 856, 874, 1057, 1067, 1069, 1094, 1185, 1257, 1295, 1377, 1388, 1396, 1405, 1570, 1590, 1640, 1674, 1777, 1792, 1901, 1907, 1970]

[159, 162] → [223, 315, 331, 334, 337, 389, 419, 730, 759, 800, 875, 888, 946, 978, 1347, 1389, 1421, 1538, 1539, 1560, 1579, 1600, 1724, 1809, 1865, 1929]

After dynamic Merging (if required): final count - 4

[85, 159, 162] → [**208**, 223, 287, 323, 333, 342, 419, 676, 753, 759, 872, 888, 893, 894, 914, 991, 1306, 1347, 1385, 1392, 1409, 1499, 1553, 1579, 1605, 1656, 1675, 1697, 1742, 1772, 1796, 1827, 1911, 1926, 1929, 1936, 1951]

[82, 117] → [6, 69, 200, 229, 297, 303, 313, 833, 855, 1394, 1405, 1582, 1792, 1882]

[56, 74] → [221, 296, 307, 336, 341, 411, 475, 519, 522, 913, 926, 942, 943, 953, 999, 1008, 1112, 1154, 1421, 1503, 1560, 1724, 1931, 1943, 1967, 1976, 1982]

[20, 46, 131] → [5, 19, 180, 235, 241, 253, 254, 395, 425, 501, 742, 856, 874, 1057, 1067, 1069, 1094, 1185, 1257, 1295, 1377, 1388, 1396, 1405, 1570, 1590, 1640, 1674, 1777, 1792, 1901, 1907, 1970]

Observation 2: These are clusters formed for 1000(from previous pass) + 1000(from current pass). And still its count is 4, instead of 7 (4 from previous + 3 current unmerged clusters), with added contextual information. Also, *so many new documents have been added*, *which were absent in previous pass*, because *they were detected as outliers*. (Example, document 208. 208¡1000, which means this document comes under pass 1 documents, but was detected as an outlier).

Pass 3

Initial Clusters obtained : for 3000 documents, after updating in initial clusters set. count - 6

[46, 131] → [5, 6, 19, 97, 180, 235, 241, 253, 254, 395, 425, 501, 694, 742, 856, 874, 1057, 1067, 1069, 1094, 1185, 1257, 1295, 1377, 1388, 1396, 1405, 1570, 1590, 1640, 1674, 1777, 1792, 1901, 1907, 1970, 2003, 2008, 2044, 2074, 2095, 2169, 2172, 2191, 2217, 2223, 2367, 2382, 2425, 2508, 2524, 2535, 2741, 2747, 2857, 2864, 3132, 3138, 3272, 3282, 3299, 3314, 3323, 3330, 3334, 3335, 3358, 3401, 3429, 3445, 3737, 3897, 3904, 3913]

[85, 159, 162] → [208, 223, 287, 323, 333, 342, 419, 676, 753, 759, 872, 888, 893, 894, 914, 991, 1306, 1347, 1385, 1392, 1409, 1499, 1553, 1579, 1605, 1656, 1675, 1697, 1742, 1772, 1796, 1827, 1911, 1926, 1929, 1936, 1951]

[82, 117] → [6, 69, 97, 200, 229, 297, 303, 313, 694, 833, 855, 1394, 1405, 1582, 1792, 1882, 2087, 2382, 2456, 2521, 3282, 3453, 3458, 3467, 3540, 3979]

[56, 74] → [221, 296, 307, 336, 341, 411, 475, 519, 522, 913, 926, 942, 943, 953, 999, 1008, 1112, 1154, 1421, 1503, 1553, 1560, 1724, 1931, 1943, 1967, 1976, 1982, 2022, 2576, 2601, 2626, 2678, 2683, 2697, 2966, 2996, 3002, 3020, 3044, 3060, 3068, 3121, 3137, 3164, 3292, 3493]

[20, 46, 131] → [5, 19, 180, 235, 241, 253, 254, 395, 425, 501, 742, 856, 874, 1057, 1067, 1069, 1094, 1185, 1257, 1295, 1377, 1388, 1396, 1405, 1570, 1590, 1640, 1674, 1777, 1792, 1901, 1907, 1970]

[20, 46] → [5, 6, 57, 97, 193, 197, 235, 327, 395, 501, 516, 694, 855, 934, 1131, 1215, 1268, 1273, 1369, 1395, 1399, 1582, 1652, 1777, 1843, 1952, 2044, 2172, 2183, 2217, 2264, 2382, 2436, 2456, 2595, 2599, 2617, 2727, 2741, 2749, 2777, 2848, 2913, 2922, 2947, 3138, 3191, 3282, 3299, 3306, 3330, 3337, 3358, 3401, 3429, 3847, 3855, 3881, 3949, 3979, 3981]

Observation 3: Due to chosing Closed Frequent ItemSet criteria for defining a cluster, a much reduced number of cluster set is coming. This number is *avoiding unnecessary duplication of documents across various clusters having similar context.*

After dynamic Merging (if required): final count - 4

[85, 159, 162] → [208, 223, 287, 323, 333, 342, 419, 676, 753, 759, 872, 888, 893, 894, 914, 991, 1306, 1347, 1385, 1392, 1409, 1499, 1553, 1579, 1605, 1656, 1675, 1697, 1742, 1772, 1796, 1827, 1911, 1926, 1929, 1936, 1951]

[82, 117] → [6, 69, 97, 200, 229, 297, 303, 313, 694, 833, 855, 1394, 1405, 1582, 1792, 1882, 2087, 2382, 2456, 2521, 3282, 3453, 3458, 3467, 3540, 3979]

[56, 74] → [221, 296, 307, 336, 341, 411, 475, 519, 522, 913, 926, 942, 943, 953, 999, 1008, 1112, 1154, 1421, 1503, 1553, 1560, 1724, 1931, 1943, 1967, 1976, 1982, 2022, 2576, 2601, 2626, 2678, 2683, 2697, 2966, 2996, 3002, 3020, 3044, 3060, 3068, 3121, 3137, 3164, 3292, 3493]

[20, 46, 131] → [5, 6, 19, 97, 180, 235, 241, 253, 254, 395, 425, 501, 694, 742, 856, 874, 1057, 1067, 1069, 1094, 1185, 1257, 1295, 1377, 1388, 1396, 1405, 1570, 1590, 1640, 1674, 1777, 1792, 1901, 1907, 1970, 2003, 2008, 2044, 2074, 2095, 2169, 2172, 2191, 2217, 2223, 2367, 2382, 2425, 2508, 2524, 2535, 2741, 2747, 2857, 2864, 3132, 3138, 3272, 3282, 3299, 3314, 3323, 3330, 3334, 3335, 3358, 3401, 3429, 3445, 3737, 3897, 3904, 3913]

Observation 4: With every pass, more context is getting added for defining a cluster, and so more documents are able get part of a cluster.

Pass 4

Initial Clusters obtained : for 4000 documents, after updating in initial clusters set. count - 6

[85, 162] → [208, 223, 287, 323, 333, 342, 419, 676, 753, 759, 872, 888, 893, 894, 914, 991, 1306, 1347, 1385, 1392, 1409, 1499, 1553, 1579, 1605, 1656, 1675, 1697, 1742, 1772, 1796, 1827, 1911, 1926, 1929, 1936, 1951, 2229, 2255, 2352, 2354, 2406, 2456, 2633, 2648, 2840, 2854, 2894, 2988, 2991, 2992, 3019, 3089, 3199, 3206, 3421, 3444, 3446, 3458, 3469, 3489, 3519, 3532, 3555, 3567, 3628, 3927, 3931, 3949]

[85, 159, 162] → [208, 223, 287, 323, 333, 342, 419, 676, 753, 759, 872, 888, 893, 894, 914, 991, 1306, 1347, 1385, 1392, 1409, 1499, 1553, 1579, 1605, 1656, 1675, 1697, 1742, 1772, 1796, 1827, 1911, 1926, 1929, 1936, 1951]

[82, 117] → [6, 69, 97, 200, 229, 297, 303, 313, 694, 833, 855, 1394, 1405, 1582, 1792, 1882, 2087, 2382, 2456, 2521, 3282, 3453, 3458, 3467, 3540, 3979]

[56, 74] → [221, 296, 307, 336, 341, 411, 475, 519, 522, 913, 926, 942, 943, 953, 999, 1008, 1112, 1154, 1421, 1503, 1553, 1560, 1724, 1931, 1943, 1967, 1976, 1982, 2022, 2576, 2601, 2626, 2678, 2683, 2697, 2966, 2996, 3002, 3020, 3044, 3060, 3068, 3121, 3137, 3164, 3292, 3493]

[20, 46, 131] → [5, 6, 19, 97, 180, 235, 241, 253, 254, 395, 425, 501, 694, 742, 856, 874, 1057, 1067, 1069, 1094, 1185, 1257, 1295, 1377, 1388, 1396, 1405, 1570, 1590, 1640, 1674, 1777, 1792, 1901, 1907, 1970, 2003, 2008, 2044, 2074, 2095, 2169, 2172, 2191, 2217, 2223, 2367, 2382, 2425, 2508, 2524, 2535, 2741, 2747, 2857, 2864, 3132, 3138, 3272, 3282, 3299, 3314, 3323, 3330, 3334, 3335, 3358, 3401, 3429, 3445, 3737, 3897, 3904, 3913]

[159, 162] → [223, 315, 331, 334, 337, 389, 419, 730, 759, 800, 875, 888, 946, 978, 1347, 1389, 1421, 1538, 1539, 1553, 1560, 1579, 1600, 1724, 1809, 1865, 1929, 2061, 2201, 2354, 2763, 2900, 3012, 3019, 3055, 3179, 3248, 3535, 3682, 3716, 3832, 3877, 3883]

After dynamic Merging (if required): final count - 4

[85, 159, 162] → [208, 223, 287, 323, 333, 342, 419, 676, 753, 759, 872, 888, 893, 894, 914, 991, 1306, 1347, 1385, 1392, 1409, 1499, 1553, 1579, 1605, 1656, 1675, 1697, 1742, 1772, 1796, 1827, 1911, 1926, 1929, 1936, 1951, 2229, 2255, 2352, 2354, 2406, 2456, 2633, 2648, 2840, 2854, 2894, 2988, 2991, 2992, 3019, 3089, 3199, 3206, 3421, 3444, 3446, 3458, 3469, 3489, 3519, 3532, 3555, 3567, 3628, 3927, 3931, 3949]

[82, 117] → [6, 69, 97, 200, 229, 297, 303, 313, 694, 833, 855, 1394, 1405, 1582, 1792, 1882, 2087, 2382, 2456, 2521, 3282, 3453, 3458, 3467, 3540, 3979]

[56, 74] → [221, 296, 307, 336, 341, 411, 475, 519, 522, 913, 926, 942, 943, 953, 999, 1008, 1112, 1154, 1421, 1503, 1553, 1560, 1724, 1931, 1943, 1967, 1976, 1982, 2022, 2576, 2601, 2626, 2678, 2683, 2697, 2966, 2996, 3002, 3020, 3044, 3060, 3068, 3121, 3137, 3164, 3292, 3493]

[20, 46, 131] → [5, 6, 19, 97, 180, 235, 241, 253, 254, 395, 425, 501, 694, 742, 856, 874, 1057, 1067, 1069, 1094, 1185, 1257, 1295, 1377, 1388, 1396, 1405, 1570, 1590, 1640, 1674, 1777, 1792, 1901, 1907, 1970, 2003, 2008, 2044, 2074, 2095, 2169, 2172, 2191, 2217, 2223, 2367, 2382, 2425, 2508, 2524, 2535, 2741, 2747, 2857, 2864, 3132, 3138, 3272, 3282, 3299, 3314, 3323, 3330, 3334, 3335, 3358, 3401, 3429, 3445, 3737, 3897, 3904, 3913]

Analysis of final result

After mapping FIS word ids of final clusters to actual words, we get

85 japan **159** - uk **162** usa

82 oilseed **117-** soybean

56- interest **74-** money-fx

20- corn **46-** grain **131-** wheat

Cluster senses thus obtained are :

1. $(japan, uk, usa)$ ⇒ showing country set
2. $(oilseed, soybean)$ ⇒ showing beans set
3. $(interest, money - fx)$ ⇒ showing interests related to money
4. $(corn, grain, wheat)$ ⇒ showing creals set

After giving a careful observation, we can see that all senses are leading a similar(inter-cluster) and disjoint(intra-cluster) meaning and context. Also, these senses are maximally occuring senses and thus are chosen as to define contexts for the documents. Other itemset were not powerful enough to show their interestingness and thus are chosen for outlier documents.

8. Conclusion

In this project a hierarchical algorithm to cluster documents using frequent itemsets on dynamic data streams is presented. This process is part of outlier detection model, which works in parallel.

On-line transaction data streams and detect those transactions that are likely to be outliers are targetted. Defining the concept of Match Closure based on the association rules with high confidence, we provide a formula called Match Difference Degree (MDD) for detecting outliers. Also, a transaction data stream outlier detecting model IODM and its strongness rules are presented. To control the dimensions of association rules, mining association rules using an incremental method.

When detecting outliers from transaction data streams on-line, we use TSB to maintain the data stream. Experiments using real-world data show that this method has sufficient detection performance in precision and running time, and can be used in detecting transaction data streams.

9. Future Work

Contextual information can be improved by using some NLP techniques. It will add similarity measure in terms of existantial similarity in meaning real data. This will help in giving better clusters for blogs like Twitter etc., where not much information is given in terms of data and its structure.

10. References

[1] Kiran G.V.R., Ravi Shankar, Vikram Pudi, "Frequent Itemset Based Hierarchical Document Clustering using Wikipedia as External Knowledge".

[2] Hasan H. Malik, John R. Kender, "High Quality efficient Hierarchical Document Clustering Using Closed Interesting Itemsets"(HCCI).

[3] Chunhua Ju, Yaolin Li, "An Incremental Outlier Detection Model for Transaction Data Streams".

[4] R. Agrawal, R. Srikant, "Fast algorithm for mining association rules".