ISSN:0975-9646

A Mining Algorithm to Generate the Candidate Pattern for Authorship Attribution for Filtering Spam Mail

Khongbantabam Susila Devi [#1] , Dr. R. Ravi [*2]

[1]*Research Scholar, Department of Information & Communication Engineering*
Anna University, Chennai, Tamil Nadu, India

[2]*Professor, Department of Computer Science & Engineering*
Francis Xavier Engineering College, Tirunelveli
Tamil Nadu, India

***Abstract* - Cyber criminals have been using internet for sharing of large amount of illegal activities universally in anonymous manner, making the forensic investigator to difficult to trace the identity of authors. The pattern decomposition algorithm significantly reduced the size of the data sets on each pass making it more efficient to mine all frequent patterns in large data sets, but it needs the support of data sets pattern to satisfy the given requirement of minimum support. Here we proposed a new data mining algorithm called Max-miner which uses the heuristic bottom up search to identify the frequent patterns as early as possible. It increases 40% of the performance compared with the aprior and decomposition algorithm and provides the high pattern matching score and counts of the frequent item sets.**

Keywords: Pattern Matching, Decomposition Algorithm, Data Mining, Max-miner Algorithm

I. INTRODUCTION

An Authorship attribution is a problem related to verifying the author of an undisclosed or disputed text if there is a closed set of candidate authors. All the approaches in authorship attribution problem are based on the fact that the author individually impact on his or her writing in a unique and recognizable style. Stylometry is the field that deals with defining and analyzing important text features that can provide as an author fingerprints. So far many of them discuss about the text features have been considered, some of them exploits text surface and take into account a regular word length or vocabulary richness while there are more complex ones dealing with text semantics or syntax trees. From the machine learning point of view, an authorship attribution problem is considered as a classification task; here the text of known authorship is assigned to one of the author from given set of candidate sets. Aprior algorithm includes the phases for finding the frequent item sets called patterns. A pattern is a set of frequent item sets appearing together along with number of databases records meeting a user defined threshold. The aprio algorithm works on a bottom up search approach by specifying the single frequent item sets. This algorithm has the expontial complexity by restricting for discovering of short patterns. The max-miner algorithm can extracts only the maximal frequent item sets, where an item is maximal if it has no superset of the frequent item sets. Max-miner is to run in time approximately linear in the number of maximal frequent item sets and the corresponding size of the database, regardless of the size of the longest frequent item sets. The Max-miner approach is strict bottom-up traversals of the search space; rather attempts to look ahead in order to quickly identify the long frequent pattern. Max-miner provides the output implicitly and consequently represents all frequent item sets and shows the performance improvements more than double times than decomposition and aprior algorithm on same data sets and provides flexibility compared with other algorithms.

II. RELATED WORK

Several new algorithms have been proposed to find the frequent pattern in data sets. The apriori algorithm is performing by employing a bottom up search [1] . This algorithm developed the candidate set to limit the minimum pattern counting to those patterns which support the less minimum requirement. At each pattern, the algorithm determines the candidates are frequent by counting their result. Due to combinatory explosion, this leads to poor performance when the pattern size is large. Here the complete set of rules cannot be extracted without the support information of the subsets of maximal frequent sets. Other techniques use the sampling method to select the random subsets of data sets to measures the candidate sets and then test those sets to identify the patterns [2,3] . The enhance Aprior along with the hashing algorithm can identify some candidates that change infrequent candidate sets if check against the database [6]. It uses the hashing algorithm to rewrite a smaller database after each pass to reduce the overhead of the subsequent pass as like the aprior, it considered the frequent itemsets. In the sampling method that it is possible that some frequent patterns are not included in candidate sets, thus the algorithm may not find all frequent patterns. Mostly the accuracy of this approach is highly depending upon the sampling techniques and characteristics of data. Lin and Kedem have proposed an algorithm called pincer- search for mining long maximal item sets [5]. It attempts to identify the long patterns

throughout the search like Max-miner. Here the difference between these algorithms is Max-miner is simple and polynomial candidate generation produced by the heuristics, while the pincer- search uses NP-hard reduction phase to provide no long candidate item sets contains any infrequent item sets.

III. APIRIOR ALGORITHM

In this**,** the first pass of the algorithm, it counts the occurrence of items to determine the large 1-itemsets. For successive pass, as K pass consist of two phases. First the large item sets L_{k-1} is found in the (K-1) pass is used to generate the candidate item sets C_k using the aprior generation function. After the first pass, next it scanned the databases and the candidate item sets C_k is counted. Due to the fast counting, it efficiently determine the candidate C_k that contains in the given transaction t.

Algorithm:

1. Take L1 be the large 1- item sets.
2. If K= 2, then L_{K-1} is not equal to phi , then increment the item sets K^{++} and start the process
3. If the candidate item sets C k is equal to the apriori generation (L_{k-1}) , then creates the new candidate item sets
4. For all transaction t which belongs to which belongs to database continue the process.
5. If item sets transaction C t is equal to subset of (C_k' t) , then the candidate item sets belongs to the candidate transaction and increment the candidate count (c.count^{++}).
6. End the process.
7. If L k is equal to the candidate c which is belongs to the candidate item sets Ck of the maximum support count is greater than equal to the minimum support.
8. Finally generate the set of maximal frequent item sets L_k , U_k

IV. PATTERN DECOMPOSITION

This paper introduced a new innovative algorithm which uses the Modified pattern decomposition to mine the frequent pattern. The Modified pattern decomposition algorithm provides three significant progresses. First one by decomposing transaction into short item sets, it has probable to link the regular pattern together, thus significantly reduced the data sets in each pass. Secondly no need to generate candidate sets, since the reduced data sets does not contain any infrequent pattern found before. Lastly using the reduced data sets saves the time for counting the occurrence of pattern.

Algorithm:

1. Take s, q_k be the frequent and infrequent item sets.
2. Decompose the frequent item set s and infrequent k-item sets of s
3. If k item sets is equal to 1
4. Remove the item in q_k from frequent item sets s.
5. Otherwise build the order frequency tree r.
6. Perform the calculation on the tree by calling the quick split.
7. Then stored the result back to quick split.
8. Map the quick split back to the item sets.

V. MAXIMAL PATTERN MATCHING CONCEPT

Basic frequent mining pattern mainly mines a large number of frequent patterns. It is mainly used to reduce the number of pattern generated.

A. Introduction to Max-miner

A data set is a group of transaction that is set over finite item domains. Transaction can be represents as the things that as a group of text written by one author or different authors writing styles or the characteristics of a person as described by his or her way of representation of text writing. A group of items is more succinctly called an item sets and frequent item sets is one that contains in the number of transaction above or equal to the minimum supports specified by the user. An item set with items will be more sufficiently referred to as a-item set and the support item sets is represented as k-item sets. The Max-miner is described by Rymon's generic set enumeration tree search framework, here the particular ordering impose on the item domains affects the parent/ child relationship in the set-enumeration tree but not its completeness, it describes the lexical ordering of the items and later provides the optimization of item sets which dramatically improves the performance by heuristically ordering the items and dynamically reordering them on a per-node basis. The set-enumeration trees are not data structures, it's like the hash tree and mainly uses to illustrate the sets of item sets are to be completely enumerated in a search problem. Max-miner purely works on breadth-first search of the set-enumeration tree in order to limit the number of passes made over the data.

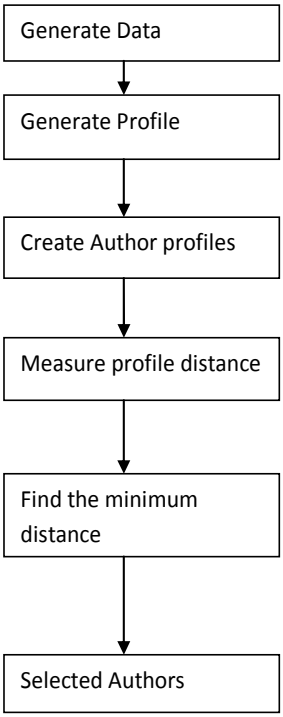

Fig1: Authorship Identification using Max-miner Algorithm

B. Characteristic of Max-Miner

Here we provide the pseudo code description of Max-miner and followed by the reason behind the description of item sets ordering policies. Implementation details describing

the how the max-miner can be use in same data uses for aprior and decomposition algorithm.

C. Max-Miner Pseudo code

The pseudo code description of Max-miner is given below in the figure. The body accepts the data sets and the minimum support specified by the users. The while loop implements a breadth first search of the set-enumeration tree that maintains every frequent item set occur so long as it is potentially maximal. The function Gen-Initial-Group works on the initial scan over the data set to identify the domain and nut the search at the second level of the tree. Here the superset frequency based pruning is implemented by expanding the sub-nodes of a candidate g if h (g) *U* t (g) is infrequent. Another instance of superset frequency pruning is any candidate group g is pruned if h (g) *U* t (g) is a subset of already known to be infrequent item set I.

1. Take the Max-miner data sets T
2. Returns the set of Maximal frequent item sets present in the transaction T
3. Set the candidate group set C is assigned to non empty sets.
4. Set the item sets F is assigned general initial groups (T, C)
5. While C is non empty then do
6. Scan T to the support of all the candidate groups in C.
7. For each group g which is belongs to the candidate C such that h(g) U t (g) ,then do
8. Frequent item sets is assigned if frequent item sets is pruned along with the superset frequency pruning
9. Set the candidate groups C new is assigned to non empty set
10. For each group g is belongs to the candidate set C when the sub nodes of candidate is infrequent then do
11. Then the frequent item set is assigned if the sub-node of candidate g, C new is the union of frequent item sets.
12. A new candidate set (C new) is assigned to the all candidate set C.
13. Then remove F from the frequent item sets with the proper super sets in F
14. Remove any candidate C from the group

Gen-Initial Groups (Data sets T, Set of candidate Group C)

1. C is passed by reference and returns the candidate Groups
2. Then the return value of the function is the frequent 1-item sets.
3. Scan the data sets T to obtained F1, the set of frequent 1-itemsets.
4. Impose an ordering on the items in F1.
5. For each item sets I in F1 other than the greatest item then do
6. Let g be the new candidate with the head h(g) is equal to set of item sets and the tail t(g) is equal to item j which follows i in the ordering
7. Then the union of candidate set and candidate group is assigned to the candidate group g.
8. Return the item sets containing the greatest item

Generating sub- nodes (Candidate Group g, Set of Candidate Groups C)

1. C is passed by referenced and returns the sub-nodes of g
2. Then the return value of the function is the frequent item sets.
3. Remove any item i from tail of the candidate group t(g) if head of the candidate group h(g) is the subset of item i is infrequent
4. Then reorder the items in tail of the candidate group t(g)
5. For each item i belongs to the tail of the candidate group t(g) other than the greatest do
6. Let g be the new candidate h(g) is equal to h(g) with the union of the subset of I and t(g) is equal to the another item j in which j belongs to t (g) and j follows i in t(g).
7. If the set of the candidate group C is the union of candidate group g then assign each group into the set of the candidate group C.
8. Return head of the candidate set h (g) is the union of the greatest set of item sets m otherwise write head of the candidate group h (g) if tail of the candidate group t(g) is empty.

VI. DESCRIPTION FOR IMPLEMENTATION

Max-miner uses the same data structure that uses in aprior and pattern decomposition algorithm for efficiently computing the item sets supports. The basic data structure use by the aprior and decomposition algorithm is that hash the tree to the index of the candidate item sets. Max-miner uses the hash tree to index only the head of the each candidate groups. For each transaction in data sets, Max-miner needs the hash tree too quickly look up all the candidate groups whose head appears in the transaction. When each candidate group g is identified, it span downwards to its tail item one by one, incrementing the support of h (g) along with the subset of the item sets i if the tail i item is present in the transaction. If each tail item is present in the transaction, then the support of head of the candidate group h (g) union along with the tail of the candidate group t (g) is also incremented. This implementation strongly faster than the individually sorting of each item sets within the hash tree. Hash tree also used in this implementation of Max-miner for efficiently identifying the subset of frequent item sets in F and C

Author	S1	S2	S3	Number of messages
XA	23	12	14	49
LF	7	9	16	32
FR	9	7	8	24
YZ	14	9	12	35
WA	16	8	9	33
DE	8	7	7	22
Grand Total of the messages				195

S1= Number of messages in colleges
S2= Number of messages under research activities
S3= Number of messages under personal interest

Table 1: English Data sets

Author	Total Number of Messages
ZZ	12
REW	14
GBS	18
NMX	13
TRD	17
FDR	18
Grand Total of Messages	92

Table 2: Arabic Data sets

Da ta sets	Measures	Aprior Algorithm			Decomposition Algorithm			Max-Miner Algorithm		
		SM	SM+SF	SM+SF+CF	SM	SM+SF	SM+SF+CF	SM	SM+SF	SM+SF+CF
News Groups	Avg. Accuracy	87.98	92.85	93.34	89.27	95.43	96.27	97.89	98.18	99.12
	Avg. Precision	86.87	91.97	93.67	89.18	95.17	96.78	97.65	98.54	99.56
	Avg. Recall	85.54	90.67	92.98	87.39	92.89	93.65	96.87	98.97	98.75
Email	Avg. Accuracy	80.23	83.65	88.23	90.22	93.52	95.64	98.11	97.23	99.23
	Avg. Precision	79.15	86.32	89.11	91.45	94.23	94.33	97.89	96.65	98.15
	Avg. Recall	78.34	85.53	88.74	91.97	94.89	95.74	98.76	96.45	99.57
Arab BBS	Avg. Accuracy	64.98	78.65	82.33	68.32	84.46	90.67	94.46	96.01	97.89
	Avg. Precision	65.13	77.43	83.45	69.22	84.89	91.47	94.98	95.34	98.02
	Avg. Recall	66.35	78.21	83.98	69.78	84.56	91.56	95.23	96.68	98.32

SM= Style Markers, SF= Structural Features, CF= Content- specific Features

Table 3**:** Performance Comparison of A prior, Decomposition and Max-miner Algorithm in terms of Precession, Accuracy and Recall

A. Results and Analysis

Based on the three data sets prepared we conducted the experiments according to the design. The results are presented in the comparison table and the discussions about the techniques are given in the following sub-sections.

B. Techniques comparison

We observed that the Max-miner, Decomposition algorithm achieved the better performance than the aprior algorithm in terms of precision, accuracy and recall with respect to all the three data sets. For example, in structural markers on the email data sets the aprior, decomposition and Max-miner achieved the accuracy of 80.23%, 90.22% and 98.11% respectively. However the performance difference between aprior , decomposition algorithm is nearly 10% difference. Then the results are generally consistent with the previous studies associated with the association rule mining algorithms, Decomposition algorithms and performance will be better in the Max-miner algorithm.

C. Arab Data Set Performance

We observed that there is significant drop in the prediction performance measures for the Arab data sets compared with the English data sets. For using style markers only aprior achieved the average accuracy of 87.98%, 80.23% for the English newsgroups and email data sets, while the Arab data sets achieved an average accuracy of 64.98%. The reason is that only 76 Arab style markers are used in our experiments, which is significantly fewer than the 205 style markers used in the English data sets. Here, we also observed that when the structural features are added to all the three algorithms achieved relatively high precession, accuracy and recall from 64.98% to 97.89% for the Arab data sets. Considering the significant language difference, this approach can improved the better performance but problem in the online message identity tracing appears in the multilingual context.

VII. CONCLUSIONS

In our experiments it identifies the author of internet news groups and email with reasonable high accuracy. Here the Max-miner can mine the maximal frequent item from the large data sets and uses the different new techniques for reducing the space for the item sets based on pruning for superset frequency. The Max-miner provides the substantial performance improvement compared with the aprior, decomposition algorithm for large data sets as well as the short data sets.

REFERENCES

1. R. Agrawal and R. Srikant, Fast algorithms for mining association rules, In VLDB, pp. 487-499, 1994.
2. Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo, Efficient algorithms for discovering association rules, In Usama M. Fayyad and Ramasamy Uthurusamy, editors, Proc. of the AAAI Workshop on Knowledge Discovery in Databases, pp. 181-192, Seattle, Washington, July 1994.

3. H. Toivonen, Sampling Large Databases for Association Rules, In Proceedings of the 22nd International Conference on Very Large Data Bases, Bombay, India, September 1996.
4. R .Rymon, Search through Systematic Set Enumeration, In Proc. of Third Int'l Conf. on Principles of Knowledge Representation and Reasoning, 539-550, 1992.
5. D.I. Lin, and Z.M. Kedem, Pincer-Search: A New Algorithm for Discovering the Maximum Frequent Set, In Proc. of the Sixth European Conf. on Extending Database Technology, to apper, 1998.
6. J. S. Park, M. S. Chen and P. S. Yu, An Effective Hash Based Algorithm for Mining Association Rules, In Proc. of the ACM-SIGMOD Conf. on Management of Data, 175-186, 1996.
7. Q. Zou, W. Chu, D. Johnson, H. Chiu, A Pattern Decomposition (PD) Algorithm for Finding All Frequent Patterns in Large Datasets. Proc. of the IEEE International Conference on Data Mining, San Jose, California, November 2001.
8. S. Wu and W. Member, Fast text searching allowing errors, Communication of ACM, 35(10): 83-91, October 1992.
9. A. Savasere, E. Omiecinski, and S. Navathe , An Efficient Algorithm for Mining Association Rules in Large Databases, In Proceedings of the 21st VLDB Conference,1995.