IJCSC 0973-7391 Volume 7 • Number 2 March 2016 - Sept 2016 pp. 73-79 available online at www.csjournalss.com

Performance Analysis of Data Mining Algorithms

Poonam Punia
Ph.D Research Scholar
Deptt. of Computer Applications
Singhania University, Jhunjunu (Raj.)
poonamgill25@gmail.com

Surender Jangra
Deptt. of Computer Applications
GTB College, Bhawanigarh (Sangrur),Punjab
ssjangra20@rediffmail.com

Abstract: Mining association rules are widely studied in data mining society. In this paper, we analyze the performance measure method of support–confidence framework for mining association rules, from which we implement and analyses data mining method by taking some parameter like time taken to generate frequent item set, no. of item generation, data size and varying the min support on different data set. And compare the experimental result of these algorithms. Experimental results show the generated rules and item set by. The execution time of all the algorithms is vary for different datasets with a variation in Min_Support. The running time of different frequent item set mining algorithms depends a lot on the structure of the data set.

Keywords: Data Mining, Frequent Items Data Set, Apriori Algorithm.

1. Introduction

Data mining is a process of discovering previously unknown and useful information from large databases. The most widely used data mining technologies include association rules discovery, clustering, classification, and sequential pattern mining Among them, the most popular technology is association rules discovery, which is mining the possibility of simultaneous occurrence of items, and then building relationships among them in databases. Association rules mining can be divided into two parts: find all frequent item sets, and generate reliable association rules straightforward from all frequent item sets. Because frequent itemsets mining is the most time-consuming procedure, it plays an essential role in mining association rules. The algorithms developed for mining frequent item sets can be classified into two types: the first is the candidates item sets generation approach, such as Apriori algorithm called Apriori-like; another aspect is a method without candidate item sets-generation approach, such as FP-growth algorithm called FP-growth-like.

2. Frequent Item set Mining

The task of frequent item set mining was first introduced by Agrawal in 1993. A frequent item set is a set of items that appears at least in a pre-specified number of transactions. Frequent item sets are typically used to generate association rules. The task of frequent item set mining is defined as follows:

Let I be a set of items. A set { X i } k = □ I is called an item set, or a k-item set, if it contains k items. A transaction over I is a couple T = (tid, I) where tid is the transaction identifier and I is an item set. A transaction T = (tid, I) is said to support an item set X , if X □ I .A transaction database D over I is a set of transactions over I. The support of an item set X in D is the number of transactions in D that supports X:

$$\text{Support}(X, D) = \{\text{tid} \mid (\text{tid}, I) \square D, X \square I\}$$

The frequency of an item set X in D is the probability of X occurring in a transaction T □ D :

$$\text{Frequency}(X, D) = P(X) = \text{Support}(X, D)$$

Note that |D| = support ({}, D). An item set is called frequent if its support is no less than a given absolute minimal support threshold abs, with $0 \leq \sigma abs \leq |D|$. The frequent item sets discovered does not reflect the impact of any other factor except frequency of the presence or absence of an item.

3. Association Rule Mining

Since its introduction in 1993 by Agrawal , the task of association rule mining has received a great deal of attention. Today the mining of such rules is still one of the most popular pattern-discovery methods in Knowledge Discovery and Data mining (KDD) . Association rule mining is a popular data mining technique because of its wide application in marketing and retail communities as well as other more diverse fields Association rule mining is a method of finding relationships of the form X□Y amongst item sets that occur together in a database where X and Y are disjoint itemsets.Support and confidence measures serve as the basis for customary techniques in association rule mining. The support and confidence are predefined by users to drop the rules that are not so interesting or useful. The association rule indicates that the transactions that contain X tend to also contain Y.

Suppose the support of an item is 0.1%, it means only 0.1 percent of the transaction contain purchasing of this item [4]. The task of mining association rules is defined as follows:

Let IS ={i1,i2,i3,…,im} a set of items and TDI ={t1,t2,t3, …,tn}be a set of transaction data items,

where ti = {ISi1, ISi2, ISi3, …, ISip}, p ≤ m and IS ij □ti , if X □ I with k = |X| is called a k-item set or simply an item set. An expression, where X, Y are item sets and X ∩Y = φ holds is called an association rule X □ Y.

The measure of number of transactions T supporting an item set X with respect to TDI is

termed as the Support of an item set.

$$\text{Support}(X) = \{T \square \text{TDI} \mid X \square T\}\text{TDI}$$

The ratio of the number of transactions that hold X □Y to the number of transactions that hold X is said to be the confidence of an association rule X □ Y

$$\text{Conf}(X \rightarrow Y) = \text{Support}(X \square Y)\ \text{Support}(X)$$

In this paper, we have presented a comprehensive survey of the algorithms and techniques available for frequent item set mining and association rule mining. The algorithms with the incorporation of economic utility factors have also been presented. A comparative study has been performed through the thorough assessment of the results of the algorithms and techniques on the basis of parameters utilized. The execution time and no of item generation and time taken with the minimum threshold for mining frequent item sets were the chief factors deliberated during the comparison.

4. Apriori Algorithm:

It is by far the most important data mining algorithms for mining frequent item sets and associations. It opened new doors and created new modalities to mine the data. Since its inception, many scholars have improved and optimized the Apriori algorithm and have presented new Apriori-like algorithms. Apriori uses a breadth-first search strategy to count the support of item sets and uses a candidate generation function which exploits the downward closure property of support. The Apriori algorithm search for large item sets during its initial database passes and uses its results as the seed for discovering other large data sets during the subsequent passes. The Apriori algorithm is based on the property of ant monotone that is if a set cannot pass a test, all its supersets fails the same test as well or in other words all nonempty subset of a frequent item set must also be frequent. Key terms in Apriori algorithm are:-

Frequent Itemset: All the set of items whose support is greater than the user defined support then such item sets are called frequent item sets. For example suppose T be transaction data base and S be the user defined minimum support. An item set X □ A is said to be frequent item set in T with respect to S if s(X) T=>=S. In other words the set item which has minimum support is called frequent item sets.

Apriori property: Any subset of a frequent item set must be frequent (downward closure property) or any superset of an infrequent item set must be infrequent (Upward closure property).

Join Operation: To find LK, a set of candidate K- item sets is generated by joining LK-1 with itself.

Prune Operation: Any (K-1)-item sets that is not frequent can not be a subset of a frequent K-item sets. Prune step helps to avoid heavy computation.

5. FP-Growth Algorithm FP-Growth Method:

Construction of FP-Tree

a. First create a root of tree labeled with "Null".

b. Scan database D second time as we scanned first time it to create 1-itemset and the L (L is sorted order of 1-itemset according to descending support count.)

c. The items in each transaction are processed in L order.

d. A branch is created for each transaction with item having their support count separated by colon.

e. Whenever the same node is encountered in another transaction, we just increment the support count of common node or Prefix.

f. To facilitate tree traversal, an item header table is built so that each item points to its occurrence in tree via a chain of node links.

g. Now the problem of mining frequent patterns in database is transformed to that of mining the FP-Tree.

6. ECLAT Algorithm:

Both Apriori and FP_Growth methods mine frequent patterns from a set of transactions in horizontal format. (TID: Itemset). While data can also be presented in the (Itemset: TID) format this format is known as vertical format. Eclat can mine the frequent itemset in the vertical data format. The vertical data format of Database D) as in case of Apriori) can be represented as in the table below.

Items Set	TID
A	{100, 103, 105, 107 }
B	{102, 104 }
C	{101, 102, 103, 104, 109 }
D	{ 100, 101, 102, 103,106, 107, 109}
E	{105, 106, 108 }
F	{101, 102, 103, 104, 105, 106, 108, 109 }

Vertical Data Format

The main difference between the Eclat and Apriori is that how they traverse the prefix tree in order to find the support of an itemset. Apriori traverse the prefix tree in breadth first order. In breadth first order it first checks item set of size 1, then item sets of size 2 and so on. Apriori determines the support of item set either by traversing for a transaction all subset of currently processed size by incrementing the corresponding

item sets counters or by checking for each candidate item set which transaction it is contained in.

7. ReLim Algorithm

Recursive elimination algorithm process the transaction directly without the prefix tree. This algorithm is strongly inspired from FP-Growth algorithm. FP-Growth algorithm is based on the prefix tree representation of dataset, which saves a large amount of memory for storing the transaction. Relim algorithm deletes all items from transaction that contains least frequent items, delete these items from transaction.

8. Experimental Results and Analysis

The results of all the algorithms discussed in the previous chapter are taken on the three different datasets of different size (different number of items and different no of transactions). This data is available at

http://fimi.cs.helsinki.fi/data/. The support factor is changed while taking the results. These datasets are given as follows.

S.No.	Dataset Name	Dataset Size	No. of Transactions	No. of Items	Name used in Results
1.	Kosrak	31.4 MB	990004	41270	11.txt
2.	T40I10D100K	14.8 MB	10002	942	12.txt
3.	T10I4D100K	3.93 MB	10004	870	13.txt

Summary of Results using Time and Min_Support Factor:- Dataset: Kosarak

Support	Apriori (t1)	Eclat (t2)	FP-Growth (t3)	Relim (t4)
0.5	2.11	2.88	2.56	1.98
1	0.92	0.56	0.39	0.73
1.5	0.23	0.27	0.14	0.42
2	0.11	0.13	0.03	0.19

Dataset: Kosarak Dataset: T40I10D100K

Support	Apriori (t1)	Eclat (t2)	FP-Growth (t3)	Relim (t4)
0.5	209.36	86.44	136.19	294.92
1	15.91	37.08	49.13	40.31
1.5	3.77	27.33	32.25	12.58
2	1.98	23.03	24.92	8.47

Dataset: T40I10D100K Dataset: - T10I4D100K

Support	Apriori (t1)	Eclat (t2)	FP-Growth (t3)	Relim (t4)
0.5	0.58	4.13	1.06	0.53
1	0.23	2.34	0.77	0.39
1.5	0.06	1.44	0.39	0.27
2	0.05	0.55	0.19	0.17

On the basis of execution time and Min_support we shows that Relim has better running time then all the three algorithms followed by Apriori, Fp-Growth, Eclat when support is low (0.5) but with the increasing support (1, 1.5, 2) Fp-Growth performs well followed by Eclat, Relim and Apriori. at support (0.5) that is Eclat has better running time followed by Fp-Growth, Apriori and Relim but with the increasing support Apriori performs better then others.

Summary of Results using Min_Support and no. of Frequent Itemsets Generated:-

Support		No. of Frequent Itemsets Generated		
Apriori	Eclat	FP-Growth	Relim	
0.5	1618	1618	1618	1618
1	383	383	383	383
1.5	189	189	189	189
2	121	121	121	121

Dataset: Kosarak

Dataset: T40I10D100K

Support			No. of Frequent Itemsets Generated	
Apriori	Eclat		FP-Growth	Relim
0.5	1282470	1282470	1282470	1282470
1	63671	63671	63671	63671
1.5	6509	6509	6509	6509
2	2289	2289	2289	2289

Support		No. of Frequent Itemsets Generated		
Apriori	Eclat	FP-Growth	Relim	
0.5	1068	1068	1068	1068
1	870	870	870	870
1.5	237	237	237	237
2	155	155	155	155

Dataset: - T10I4D100K

All the above three table shows that in each dataset the same no. of Frequent Itemsets are generated with respect to same support. For Example in dataset T10I4D100K the no. of frequent itemsets generated by all the algorithms is 1068 with a support 0.5

9. Conclusion

It is clear from the above results that all the algorithms will generate the same number of frequent itemsets with respect to a specific Min_Support on a given dataset. The execution time of all the algorithms in variable for different datasets with a variation in Min_Support. The running time of different frequent item set mining algorithms depends a lot on the structure of the data set. General statements are therefore difficult. On many standard benchmark datasets Apriori is outperformed by FP-growth, but not on all. Apriori can perform well if the data set is sparse and the average transaction size is small, in particular, if there are no long transactions, but is not necessarily.

Sparseness alone is not enough to perfectly predict the performance. So the performance of these algorithms depends a lot on datasets (structure of dataset).

References

[1] Xindong WU . " Data Mining : artificial intelligence in data analysis". Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004 PP.7.

[2] Han, Jiawei and Camber, Micheline. Data Mining : Concept and Techniques. San Franciso CA, USA, Morgen Kufmann Publishers, 2001.

[3] R.Evans, and D.Fisher, "Overcoming Process Delays with Decision Tree Induction", IEEE Expert, Vol.9, No.1, 1994, pp. 62-66.

[4] http://www.crisp-dm.org/CRISPWP-0800.pdf

[5] Introduction to Data Mining and Knowledge Discovery Third Edition by Two Crows Corporation

[6] Awan, M.S.K., Awais, M.M, "Data Mining - Redefining the Boundaries" IEEE/ACS International Conference on Computer Systems and Applications, Amman, 2007, pp. 416-423

[7] Qi Luo "Knowledge Discovery and Data Mining", Work shop on Knowledge Discovery and Data Mining, 2008, Adelaide, SA , pp.3-5

[8] Lobur M., Stekh Yu., Kernytskyy A, Sardieh F.M.E. "Some trends in Knowledge Discovery and Data Mining" International Conference on Perspective Technologies and Methods in MEMS Design, 2008. EMSTECH 2008, 21-24 May 2008 PP. 95 – 97

[9] Tian Lan; Runtong Zhang; Hong Dai "A New Frame of Knowledge Discovery" First International Workshop on Knowledge Discovery and Data Mining, WKDD 2008, 23-24 Jan. 2008, Page(s):607 – 611

[10] Yi Peng; Gang Kou; Yong Shi; Zhengxin Chen, "A Systemic Framework for the Field of Data Mining and Knowledge Discovery", ICDM Workshops 2006. Sixth IEEE International Conference Dec. 2006 pp 395 – 399

[11] Fu, Yongjlan "Data Mining: Tasks, techniques and applications" IEEE Potentials, (1997), pp.18-20.

[12] Lam N.S. "Discovering Association Rules in Data Mining" Department of Computer Science, University of Illinois at Urbana-Champaign [Online]. Available: www.raymond-lam.com/CS411.doc

[13] Agrawal R., Srikant R. "Fast Algorithm for Mining Association Rules", Proceedings of the 20th VLDB Conference Santiago, Chile, 1994 , pp. 487-499.

[14] Wojciechowski M., Galecki K., Gawronek K. "Concurrent Processing of Frequent Itemset Queries Using FP-Growth Algorithm" [Online]. Available: http://www.cs.put.poznan.pl/mwojciechowski/papers/admkd05a.pdf

[15] Thieme S. L., "Algorithmic Features of Eclat", [Online]. Available: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-126/schmidtthieme.pdf

[16] Borgelt C. "Efficient Implementations of Apriori and Eclat", 2003 [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?

[17] Pei, Jian^Han, Jiawei^Lu, Hongjun^Nishio, Shojiro^Tang, Shiwei^Yang, Dongqing , "H-Mine: Fast and space-preserving frequent pattern mining in large databases" IIE Transactions , June, 2007, pp. 593-605

[18] C. Borgelt, "Keeping things simple: finding frequent item sets by recursive elimination" International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, 2005, pp: 66-70