Open access Journal International Journal of Emerging Trends in Science and Technology

DOI: http://dx.doi.org/10.18535/ijetst/v3i01.12

An Efficient Algorithm to Mine Non Redundant Top K Association Rules

Authors

Amardeep Kumar[1], Arvind Upadhyay[2]

[1]M.E Scholar Dept of Computer Science and Engineering, IES IPS Academy Indore M.P

Email: *akamardeep11@gmail.com*

[2]Associate Professor IES IPS Academy Indore M.P

Email: *upadhyayarvind10@gmail.com*

Abstract

Association rule mining is employed the foremost well-liked fiction within the field of analysis of knowledge mining. This paper presents a survey of some commonest techniques, that square measure often used for mining association rules from a knowledge set.

Association mining may be a cardinal and advantageous researched data processing proficiency. However, looking on the choice of the arguments (the minimum support and minimum confidence), current algorithms will become terribly slow associated generate an exceptional large quantity of results or generate none or too few results, eliding helpful data, as a results of in apply users have restricted resources for analyzing the results and thus square measure usually only fascinated by discovering a particular amount of results, and fine standardization the parameters is time overwhelming. To handle this disadvantage, we tend to tend to propose associate formula to mine the top-k association rules, where k is that the variability of association rules to be found and is ready by the user. The formula utilizes a replacement approach for generating association rules named rule growths and includes several optimizations experimental results show that the formula has marvelous performance and quantify ability that it's associate advantageous completely different to classical association rule mining algorithms once the user would like to manage the number of rules generated.

Keywords: *association rule mining, top-k rules, rule enlargement, support supporting.*

Introduction

In data mining, association rule learning is a wide spread and well researched technique for locating interesting relations between variables in massive databases.

It is meant to spot robust rules discovered in data bases using different measures of power. Based on the conception of robust rules,[1] introduced association rules for locating regularities between merchandise in large-scale dealing knowledge noted by point of sale (POS) systems in super markets.

For example, the rule found within the sales knowledge of a food market would indicate that if a client buys onions and potatoes along, hearses probably going to additionally get hamburger meat. Such information are often used because the basis for decisions regarding marketing activities like, e.g., promotional evaluation or product placements. In addition to the above example from market basket analysis association rules are used these days in several application areas as well as web usage mining, bioinformatics and intrusion detection. As against sequence mining, association rule learning generally doesn't take into account the order of things either inside a transaction or across transactions.

Actual process work as follows. First we need to clean and integrate the databases. Since the data source may come from different databases, which may have some in consistence and duplications, we must clean the data source by removing those noises or make some compromises. Suppose we have two different databases, different words are used to refer the same thing in their schema. When we try to unite the two

sources we can only choose one of them, if we know that they indicate the same thing. And also real world data end to be incomplete and noisy due to the manual input mistakes. The integrated data sources can best or edina data base, data ware house or other repositories. As not all the data in the database are related tour mining task, the second process is to select task related data from the integrated resources and transform them in to a format that is ready to be mined.

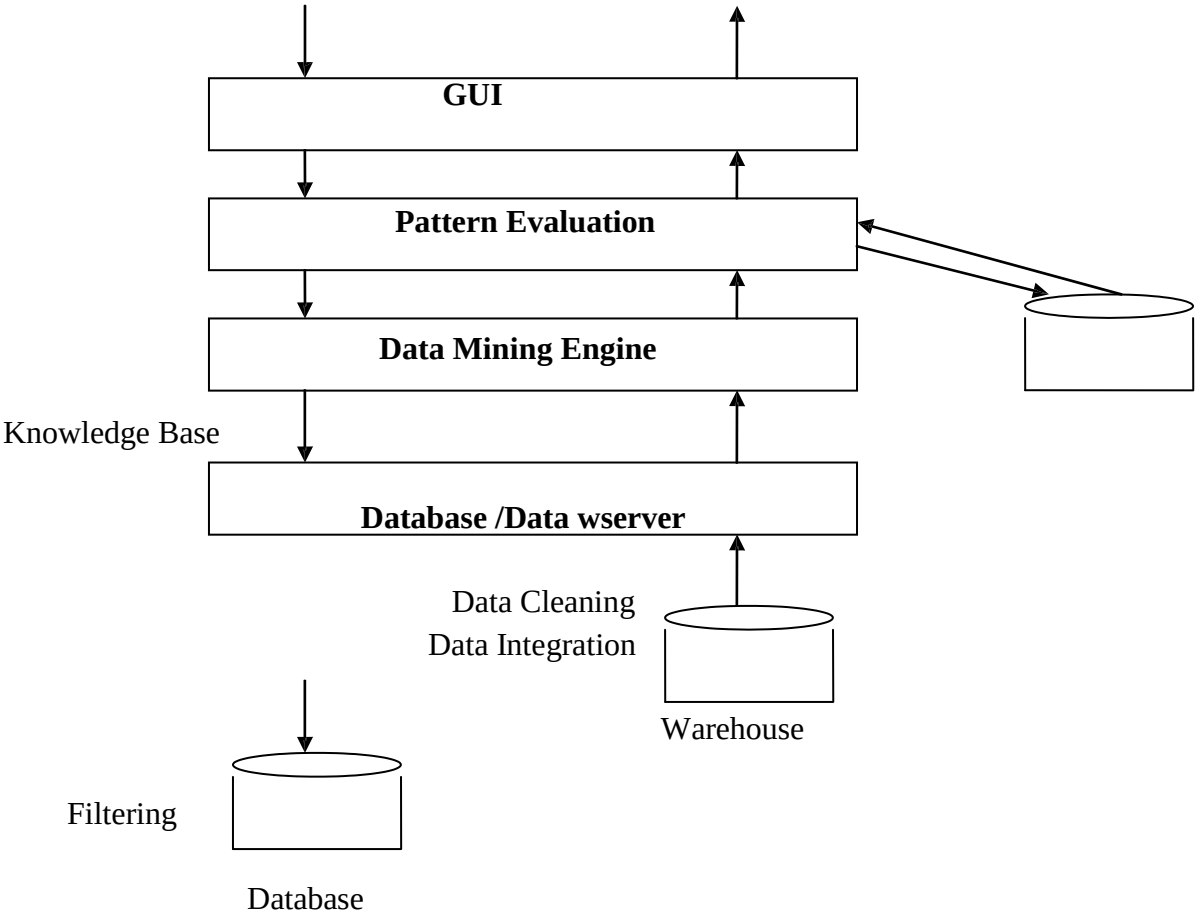


Fig1 Knowledge Discovery in Data base processes

Data mining is generally thought of as the process of finding hidden, nontrivial and previously unknown information in large collection of data. Association rule mining is an essential component of data mining. Basic objective of finding association rules is to find all co-occurrence relationship called associations.

Most of the research efforts in the scope of association rules have been oriented to simplify the rules et and to improve performance of algorithm. But these are not the only problems that can be found and when rules are generated.

Related Work Review

As result in old approaches like Apriori, Charm, FP-tree [2,3,4,5,6] they produce all possible rules that can be satisfied by user gives min sup and min conf.

In old mining approaches user have limited control they can't access limited rules. Just think about a scenario where user will mine rule from 1000000 transactions and he find 10000 rules how they can predict best value.

A user having no apriori Knowledge of the database has only a 0.08% chance of selecting amins up value that will make him satisfied. And if value is too high, not enough rules will be generated, and if value is too low it used approach generate too many rule and decrease performance.

In practice users have limited resources (time and storages pace) for analyzing the results and thus are often only interested in discovering a certain

amount of rules, and fine tuning the parameters is time- consuming.

It requires two inputs that are user specified minimum supporters hold (MST) & minimum confidence threshold (MCT). But it is not mention how the user chooses number of rules to be mine.

In current algorithms [7] user can't choose how many rules they want to mine.

Depending on the choice of the thresholds, current algorithms can become very slow and generate an extremely large amount of results orgeneraten one or too few results, omitting valuable information.

Output of existing algorithm produces duplicate rules.

As production of larger ules on existing algorithm like Apriori we can't take algorithm output as input to another program or system.

Implementation of existing algorithm is too costly and required many additional resources.

Most of algorithms implementation is not possible because it required high speed processor (dual core or high) and RAM (min 4 GB).

Mined rules can be arranged in order of min conf.

Existing data mining algorithm work on system buffer so large memory size is required to hold output as well as input data.

Problem Formulation

Let S be the database of exchanges and J = {J1,..., Jn} be itemset. An exchange T incorporates one or more than one things in J. An affiliation tenet has the structure X → Y, where Y and X are non-vacant arrangements of things (i.e. Y and X are subsets of J) such that X ∩ Y = Null. An arrangement of things is called an itemset, while X is known as the forerunner. The backing of a thing (or itemset) x is the rate of exchanges from S in which that thing or item set happens in the database The certainty or quality c for an affiliation principle X → Y is the proportion of the quantity of exchanges that contain X or Y to the quantity of exchanges that contain X. The issue of mining affiliation tenets is to discover all affiliation rules in a database having a backing no not exactly a client characterized limit minsup and a certainty no not exactly a client characterized edge minconf. For instance, Figure demonstrates an exchange database (left) and the affiliation guidelines found for minsup = 0.5 and minconf =0.5 (right).

Table 3.1 Transition Table example

ID	Transaction
T1	{a, b, c, e, f, g}
T2	{a, b, c, d, e, f}
T3	{a, b, e, f}
T4	{b, f, a, g}
T5	{b, f, g , c, d}
T6	{b, f. g, a, e}
T7	{b, f, g, a}
T8	{b, c, f, g}
T9	{b, d, f, g}
T10	{b, c, d, e, f, g}
T11	{b ,d, e, f}
T12	{b, f, g, c ,d}
T13	{b ,f, g ,a ,e}
T14	{a, b ,c, d, e, f}
T15	{b, f, g, a}

Table 3.2 Output of Apriori Algorithm

ID	Rules	Support	Confidence
R1	{a}=>{b}	0.75	1
R2	{a}=>{c, e, f}	0.5	0.6
R3	{a}=>{e, f}	0.75	1
R4	{a}=>{f}	0.8	1
R5	{a} =>{c, e}	0.4	0.5
R6	{a}=>{b, f}	0.6	0.75
R7	{a} =>{c, f}	0.5	0.79
….	…..	…..	…..

Problem Statement

In investigations of old mining systems we address an issue that they create more than client particular prerequisite and their start to finish way to deal with create principles require more opportunity to think about standards and produce last result. What's more, Final result contain more than some restricted pre-characterize particular results in this manner we can't address the outcome for another module of a project to pick an outcome and procedure the yield furthermore some yield will be repetitive. Consider a circumstance on business sector wicker container issue, in the event that we have more than 100000 results and client need to show one and only decide that backing and certainty is equivalent to or more noteworthy than give values then all old calculations can show close around 1000 principles and can't stop execution until all guidelines are not found. In this manner every single past calculation are not a decent answer for choice emotionally supportive network or future expectation.

Drawbacks of Existing Mining Method

1. As result in old methodologies like Apriori, Charm, FP-tree they create every single conceivable standard that can be fulfilled by client gives minsup and minconf.
2. In old mining methodologies client have constrained control they can't get to restricted tenets. Simply consider a situation where client will mine tenet from 1000000 exchange sand he discover 10000 principles how they can foresee best esteem.
3. A client having no from the earlier information of the database has just a 0.08 % possibility of selecting a minsup esteem that will make him fulfilled. What's more, if quality is too high, insufficient principles will be produced, and if worth is too low it utilized methodology create an excess of tenet and lessening execution.
4. By and by clients have restricted assets (time and storage room) for breaking down the outcomes and subsequently are regularly just inspired by finding a sure measure of guidelines, and calibrating the parameters is tedious.
5. It requires two inputs that are client indicated least bolster edge and least certainty limit. In any case, it is not say how the client picks number of standards to be mine.
6. In current calculations client can't pick what number of guidelines they need to mine.
7. Depending on the decision of the edges, current calculations can turn out to be moderate and create a to a great degree extensive measure of results or produce none or excessively few results, excluding important data.

8. Output of existing calculation produces copy rules.
9. As generation of substantial standards on existing calculation like Apriori we can't take calculation yield as info to another project or framework.
10. Implementation of existing calculation is too immoderate and required numerous extra assets.
11. Most of calculations usage is impractical in light of the fact that it required fast processor (double center or high) and RAM (min 4 GB).
12. Mined tenet output be orchestrated all together of minconf.
13. Existing information mining calculation deal with framework support so expansive memory size is required to hold yield and in addition data information.

Proposed Work

A vital issue that has not been tended to by every one of the calculations is the means by which the client ought to pick the edges to create a fancied measure of guidelines. This issue is essential in light of the fact that

– Users have restricted assets for breaking down the outcomes.

– Fine tuning the parameters is tedious.

If the threshold is set too high, the algorithm generates too few results, omitting valuable information. If threshold is set too low, it can generate an extremely large amount of results, and algorithm can become very slow.

Our planned answer uses 2 parameters 1st is 'k' i.e is that the variety of rules to be generated and second is mininmum confidence (minconf). Some connected works have used the term "top-k association rules mining". However they're applied to mining streams or mining non-standard rules. [30].

Algorithm the TopKN Rules

The outline of the proposed algorithm is as follows:

Step 1: Start

Step 2: Read the following as input

- Transaction Database S
- Parameter K
- Minimum confidence

Step 3: variables used in the algorithm are as follows:

- N- used to store the top k association rules
- E- used to store the rules to be expanded right or left
- Minimum support

Step 4: Initially

- set N = Null
- E = Null
- Minimum Support = 1

Step 4: The transaction data base is scanned one time and the Tids (Transaction ids of each item I is stored in a variable called Tids(i)

Step 5: for every pair of items X & Y

Where Tids(X) >= minimum support & Tids(Y) >= minimum support
Set Tids(X => Y) = Null &
Set Tids(Y =>X) = Null

Step 6: for each Tids which belongs to the Tids (X) ∩ Tids (Y)

If X occurs before Y in the Tid's then Tid's(X => Y) = Tids (X => Y) U {s} OR

If Y occurs before X in Tids then Tids(Y => X) = Tids (Y => X) U {s}

END FOR

Step 7: If │Tids (X => Y) │/ │Total sequences│> = minimum support

then

Confidence (X => Y) = │Tids(X=>Y) │/│Tids(X) │ If

Confidence (X=>Y) >= minimum confidence then

If │N│< K then

If there exist a rule r2 in N which is similar to the currently generated rule r1& whose support is also similar to the support of r1 then the rule r1 is not added to N

Otherwise this rule r1 is added to the N.

It is also added to E: E = E U r1.

If │N│ >= K then

Remove a rule s from N whose support is equal to the present minimum support Set minimum support = lowest support of rules in N
END IF END FOR

Step 8: while E is non empty Do

Select arule rwith the highest support in E
Perform the left expansion
Perform right expansion
Remove r from E
END WHILE

Results of Evaluation

We have executed the novel technique for refined association tenet mining in Java and performed investigates a PC with a double center processor running windows XP and 2 GB of free RAM. On the premise of different parameters we assess the outcomes in this area .Experiments are carried on genuine manufactured datasets ordinarily utilized as a part of the affiliation standard mining writing, to be specific Retail, Mushrooms, Chess ,Connect and so on. Table 5.1 condenses the attributes of a portion of the datasets.

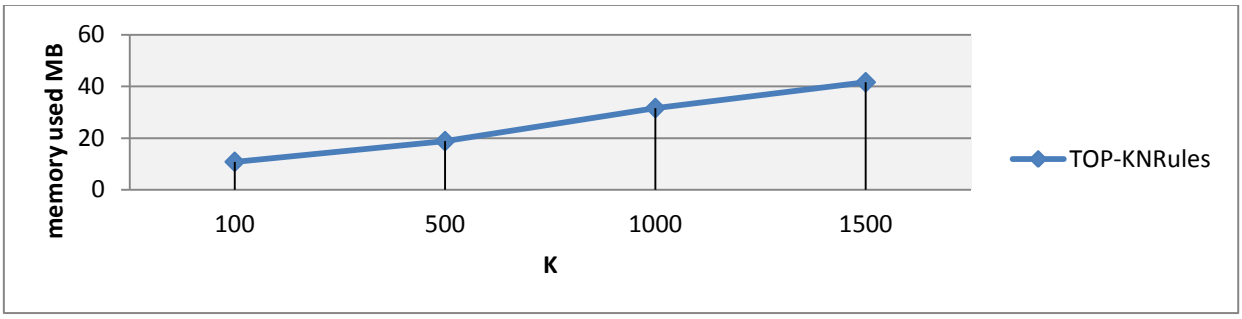
Table 4.1 Datasets Characteristics.

Data Set	No of Transaction	No of distinct items	Average transaction
Chess	3196	75	37
Mushrooms	8416	128	23
Accidents	340000	468	49
t25i10d10k	10000	129	43
Retail	88163	16470	10

Impact of the k parameter: When we ran top positioned rules with minconf = 0.7 on each dataset and shifted the parameter k from 100 to 1500 to assess its impact on the aggregate execution time and the memory use of the calculation. Execution time is communicated in seconds or milliseconds and the most extreme memory use is communicated in megabytes. Our perception is that the execution time and the greatest memory use is sensible for all datasets, we can see that the calculation execution time becomes straightly with k, and that the memory utilization gradually increments.

Table 4.2: Shows runtime and memory used with varying k value on different datasets

Datasets	Execution Time sec			Maximum Memory Usage MB		
Rule k	100	1000	1500	100	1000	1500
Chess	0.265	3.151	4.836	7.349	10.447	12.16
Mushrooms	2.013	38.69	64.55	11.52	69.83	113.5
T25i10d10k	2.496	147.5	188.8	20.67	121.31	167.9
Accidents	5.507	73.63	121.5	45.24	139.04	217.4



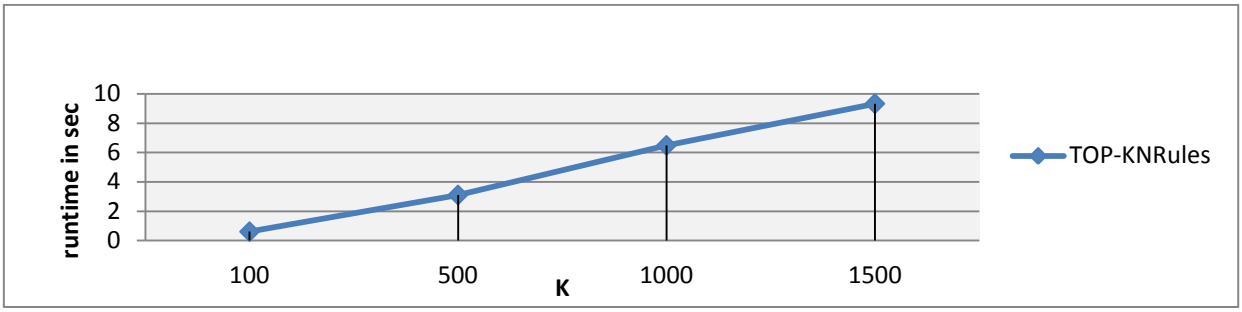


Figure 4.1: Detail results of varying K for the Mushrooms datasets

Impact of the minconf parameter: We then ran the same dataset however changed the minconf parameter to watch its effect on the execution time and the memory use. Table 5.2 demonstrates the outcome acquired for minconf=0.3, 0.5and 0.7 for k=2000, for a retail information set. Our perception is that the memory necessity and execution time increments when the minconf parameter increments.

Table 4.3: Results for K=2000 and minconf=0.3, 0.5, and 0.7

Datasets	Execution Time sec			Maximum Memory Usage MB		
	min conf =0.3	min conf =0.5	min conf =0.7	min conf =0.3	min conf =.05	min conf =.07
Chess	6.676	6.672	6.736	12.89	13.26	13.90
Mushrooms	50.685	66.16	95.846	65.354	104.52	180.165
t25i10d10k	27.971	33.573	37.581	18.392	38.67	68.52
Accidents	80.075	81.604	93.585	217.70	236.20	247.46

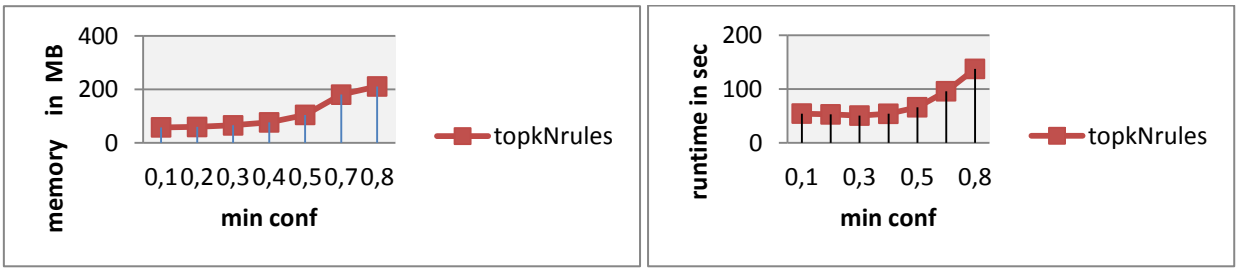


Figure4.2: Detail results of varying minconf for the Mushrooms datasets

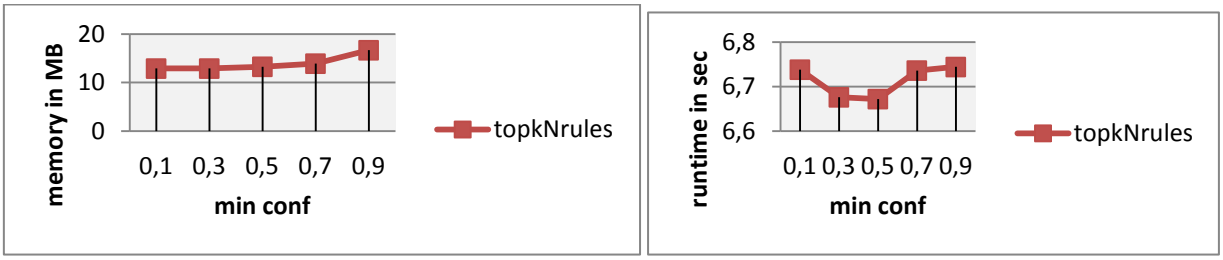


Figure4.3: Detail results of varying minconf for the Chess datasets.

Execution Comparison: Next to assess the advantage of utilizing top k positioned rules; we contrasted its execution and past calculation FP-tree calculation. Since the FP-tree calculation and our new proposed calculation are not intended for the same errand i.e. mining all affiliation rules versus mining the top k rules, it is hard to think about them.

To give a correlation of their exhibitions, we ran top k rules calculation on the distinctive dataset with minconf =0.7 and k=100, 500, 1000… 1500. We then ran FP-Tree development with minsup equivalents to the most minimal backing of guidelines found by top k rules, for every k and every information set. We assessed the outcome utilizing retail information's.

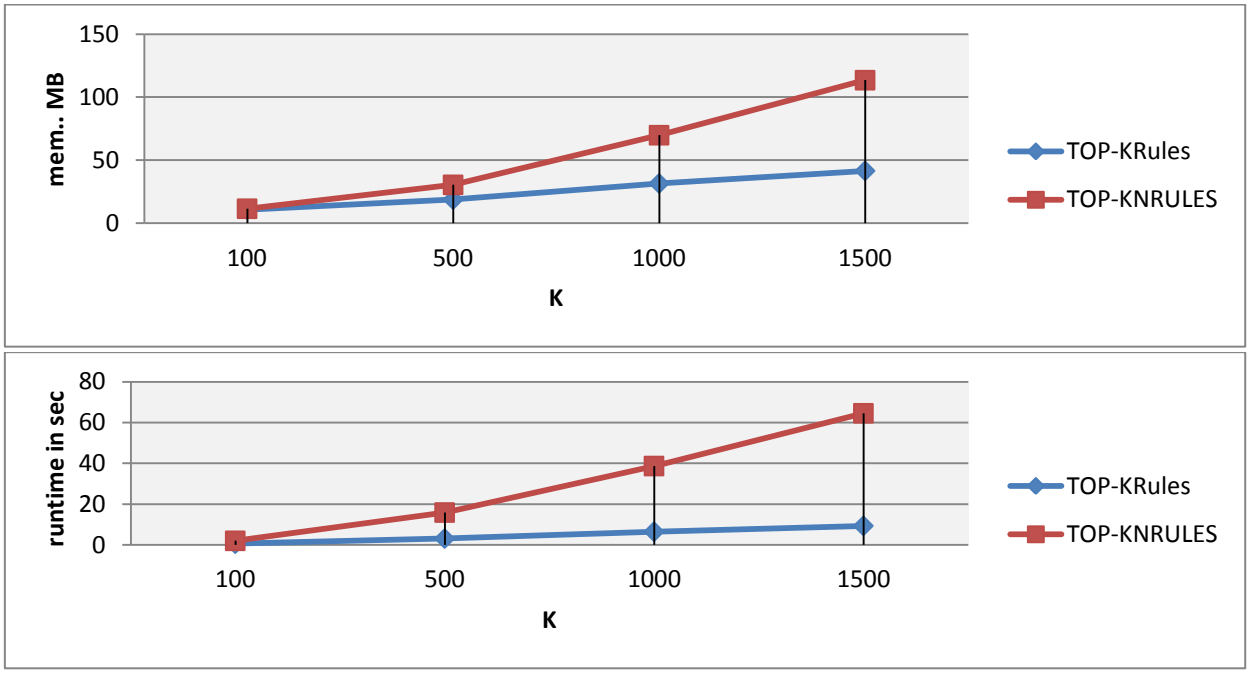


Figure 4.4: Performance comparison for optimal minsup values for Mushrooms.

Figure4.4: shows examination between the old and the new calculation on the premise of execution time and memory utilization. The conclusion from this assessment is that for an ideal decision of parameters, our calculation is just about as quick as past calculation, likewise as k builds the hole between the two calculations increments. In the event that the parameters are not picked ideally, FP-tree development calculation can run much slower than top k rules, or to produce excessively few or an excess of results. This assessment unmistakably shows the advantages of utilizing top positioned affiliation principle mining.

Table 4.4: Shows the comparison b/w old and new algorithm for k=2000 rules.

K	Min confidence	Old(TopKRules)	New(TopKNRules)
2000	0.1	2042	2000
2000	0.3	2042	2000
2000	0.5	2006	2000
2000	0.7	2031	2000
2000	0.8	2107	2000

Size of standards found

In conclusion, we explored what is the normal size of the top-KN rules on the grounds that one may expect that the guidelines might contain couple of things. This is not what we watched. For Chess, Accidents, T25I10D10K and Mushrooms, k=2000 and minconf=0.7, the normal number of things by tenets for the main 2000 principles is separately 4.32, 4.55, 5.87 and 5.38, and the standard deviation is individually 0.932, 0.92, 1.51 and 1.30, with the biggest tenets having eight things.

Conclusion

When the info set is simply too massive, the final association rule mining rule will generate a particularly great amount of rules. It takes plenty of execution time and additionally consumes vast memory. In another case the association rules mining rule might generate rules with redundant information set row. During this specific case loss valuable data and user can't opt for what number rules they need to show. To beat these higher than mentioned issues, we tend to project a completely unique rule for mining prime hierarchal information from any normal information set. Testing proposed rule on a regular information set. This information set is obtainable underneath general public license (GNU).

References

1. Agrawal, R., Imielinski, T., and Swami, A. N: Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, P. Buneman and S. Jajodia, Eds. Washington, D.C.,1993, pp207-216.
2. Agrawal, R. and Srikant, R.: Fast algorithms for mining association rules. In Proc. 20th International Conference Very Large Data Bases, VLDB, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, 1994,pp 487-499.
3. Agrawal, R. and Srikant, R.: Mining sequential patterns. In Eleventh International Conference on Data Engineering, P. S. Yu and A. S. P. Chen, Eds. IEEE Computer Society Press, Taipei, Taiwan, 1995, pp 3-
4. Bayardo R., Agrawal, R., and Gunopulos D: Constraint-based rule miningin large, densedatabases. 1999.
5. Berkhin, P.: Survey of clustering data mining techniques. Tech. rep., Accrue Software, SanJose, CA, 2002.
6. Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. : Dynamic itemset counting and implication rules Formarket basket data. In SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, USA, J. Peckham, Ed. ACM Press, May13-15, 1997, pp 255- 264.
7. Chen, M.-S, Han, J, and Yu, P. S.: Data mining: an overview from a database perspective. IEEE Trans. On Knowledge and Data Engineering 8, 1996,pp 866-883.
8. Das A., Ng, W.K., and Woon Y.K. : Rapid association rule mining. In Proceedings of the tenth international conference on Information and knowledge management. ACM Press, 2001, pp 474-481.
9. Duda, R. and Hart.: Pattern Classification and Scene Analysis. Wiley & Sons, Inc, 1973.
10. Garofalakis, M. N., Rastogi, R., and Shim, K. : SPIRIT: Sequential pattern mining with regular expression constraints. In The VLDB Journal. 1999,pp 223-234.
11. Han, J and Kamber M.: Data Mining Concepts and Techniques Morgan Kanufmann. 2000.
12. Han, J, Koperski K, and Stefanovic N.: GeoMiner-a system prototype for spatial datamining, 1997, pp 553- 556.
13. Han, J. and Pei, J.: Mining frequent patterns by pattern-growth: methodology and implications. ACM SIGKDD Explorations Newsletter 2, 2,2000, pp 14-20.
14. Han , J., Pei, J., and Yin, Y.: Mining frequent patterns without candidate generation. In 2000 ACM SIGMOD Intl. Conference on Management of Data, W. Chen, J. Naughton, and P. A. Bernstein, Eds. ACM Press, 2000, pp 1-12.
15. Han, J.: Mining knowledge at multiple concept levels. In CIKM. 1995,pp 19-24. Han J. and Fu Y.: Discovery of Multiple Level Association Rules from Large Databases. In Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95),

ZÄurich, Switzerland, September 1995, pp 420-431.

16. James, M.: Classification Algorithms. Wiley& Sons, Inc, 1985.
17. Koperski, K. and Han, J.: Data mining methods for the analysis of large geographic databases, 1996. Klemettinen, M., Mannila, H., et.al : Finding interesting rules from large sets of discovered association rules. In Third International Conference on Information and Knowledge Management (CIKM'94), N.
18. R. Adam, B. K. Bhargava, and Y. Yesha, Eds. ACM Press, 1994, pp 401-407.
19. Murthy, S. K.: Automatic construction of decision trees from data- A multi-disciplinary survey. Data Mining and Knowledge Discovery 2,4, 1998,pp 345-389.
20. Ng, R. T., Lakshmanan, L. V. S., et.al. Exploratory mining and pruning optimizations of constrained associations rules.1998, p 13-24.
21. Park, J. S., Chen, M.-S., and Yu, P. S.: An effective hash based algorithm for mining association rules. In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, M. J. Carey and D. A. Schneider, Eds. San Jose, California, 1995, pp 175-186.
22. Pazzani M.J.: Knowledge discovery from data IEEE Intelligent System, Vol. 15, Issue 12, March – April 2000, pp 10-13.
23. Pei, J. and Han, J. : Can we push more constraints into frequent pattern mining. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press, 2000, pp 350- 354.
24. Psaila, G. and Lanzi, P. L.: Hierarchy-based mining of association rules in data warehouses. In Proceedings of the 2000 ACM symposium on Applied computing2000. ACM Press, 2000, pp 307-312.
25. Savesere, A., Omiecinski, E., and Navathe, S.: An efficient algorithm for mining association rules in large databases. In Proceedings of 20th International Conference on VLDB1995.
26. Smythe and Goodman. : An information theoretic approach to rule induction from databases. In IEEE Transactions on Knowledge and Data Engineering, IEEE Computer SocietyPress 1992.
27. Srikant, R. and Agrawal, R.: Mining quantitative association rules in large relational tables In Proceedings of the 1996 ACM SIGMOD international conference on Management of data. ACM Press, 1996, pp1-12.
28. Srikant, R., Vu, Q., and Agrawal, R.:Mining association rules with item constraints. In *Proc. 3rd Int. Conf. Knowledge Discovery and DataMining, KDD*, D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, Eds. AAAIPress, 1997, pp 67-73
29. Webb and S. Zhang,: k-Optimal-Rule-Discovery, Data Mining and Knowledge Discovery, vol. 10, no. 1, 2005,pp. 39-79.
30. Y. You, J. Zhang, Z. Yang and G. Liu:Mining Top-k Fault Tolerant Association Rules by Redundant Pattern Disambiguation in Data Streams, Proc. 2010 Intern. Conf. Intelligent Computing and Cognitive Informatics, March 2010, IEEE Press, pp. 470-473.
31. Zaki M.J.: Scalable Algorithms for Association Mining. IEEE Transaction on Knowledge and Data Engineering, Vol. 12, No. 3, May – June 2000, pp 372-390.