AN ENHANCED FP-GROWTH BASED NEW MINING TECHNIQUE FOR VERY LARGE DATASETS IN E-COMMERCE

R. Z. Inamul Hussain[1], Dr. S. K. Srivatsa[2], M. Mohamed Rafee[3]
[1]Research Scholar, Sri Chandrashekhendra Saraswathi Viswa maha Vidyalaya University, Enathur Kanchipuram, India
[2]Retired Professor, Anna University, Chennai, India
[3]Assistant Professor Department of IT CAHCET

Abstract

Association rule mining is one of the best significant extents in data mining, which has received a great deal of consideration. The determination of association rule mining is the detection of association relationships or correlations amongst a set of items. In this paper, we present an efficient way to discovery the valid association rules among the infrequent items, which is rarely mentioned and whose importance often get ignored by other researchers. We design a new data structure; we have proposed a model of preprocessing, mining of patterns and assignments of weights to discover highly positive association rules. To find the frequently occurring sets of data in databases, we have used clustering to group similarity, parallel processing each group and used our algorithm to find the frequent itemsets from which association rules generated and finally our algorithm to mine the frequent patterns. All the frequent items sets are combined with the help of Cartesian product from which positive rules are generated. This algorithm is applied to E-Commerce website to generate recommendation to the users. This algorithm is more memory efficient and runs extremely fast on large databases. The results display that the projected model can mine association rules with high correlation which significantly improves the efficiency of the mining processes.

Keywords: Association rule mining, big data, recommendation system, frequent item sets, E-commerce

1. Introduction

1.1. Association rule mining

In order to mine beneficial information, it is essential to achieve processing on a great amount of data for which Data mining approaches are extensively used. Processed data is straight understood or exploited in order to feed into additional processes. Pattern analysis has newly attracted a significant attention to researchers and experts because it has been evident in many areas such as decision provision, market approach, financial predictions. Association Rule mining [1, 2] is one of the algorithms, used to find valuable and valuable patterns from big dataset.

In numerous cases, Pattern analysis methods are used to discovery relations that can be of interest to a specific application domain. Itemsets, graph, orders and association rules are the numerous categories of patterns that happen in databases. Selection of the technique depends not only on what should be found but also depends on the nature of input data and opinion of the data, essential to be represented in a more brief and intelligible method. Confidence and Support are the principles used to filter out the patterns [3]. While the confidence displays how frequently items happen among all records, support signifies frequency of patterns happening in the dataset.

Association rules determine all rules which gratify the user-defined minimum confidence and minimum support limits. Market Basket Analysis is a request of association rule mining. In this request, correlations between the purchased items are examined. An example of the association rule is as follows,

butter□cheese[support =10% and confident=80%]

This rule shows that 10% of consumers purchase butter and cheese together and customers whopurchase cheese also purchase cheese 80% of the time. Association rules offer useful and interesting itemsets from the dataset. Among them, frequent itemset is constant with the prospects of the researchers and they are the witnesses of recurrent phenomena. However in many applications, search for rare items are more interesting [4]. Rare items are conflicting to frequent itemsets i.e. they do not happen frequently in the dataset. They possibly challenge the beliefs of domain experts and resemble to unexpected occurrences. Rare items convey highly interesting evidence to many domains containing medicine or biology.

Single minimum support constraint model undertakes that all items have related frequency or nature in the dataset. In many real life applications, we will meet following problems [5]:

If the minimum support is set to a higher count, we cannot find those rules that contain of rare item sets.

For discovering the rule which contains both rare and frequent items, we have to set low minimum support value. But it produces a large amount of frequent patterns which are not valuable.

Association rule mining is a very vital topic in data mining. To produce associations is to find relationships or correlations between a set of items. An association rule in the form X □ Y can be understood as X→Y the items with attribute X are likely to have attribute Y". Because of its clear and informal reasonable format, association rule mining is extensively used in transaction data analysis in business decision-making procedure.

Agrawal [1] first presented the problem of deriving a unconditional association rule from transactional databases. The unique concern is about to invent relationships among various itemsets in a ``market-basket" database. One can get information of one set of items from the information of the other set of items by finding the association rules. Such data will be useful for sales drives. Since then, to find association rules has become a vital field of data mining and substantial researches have been showed on association examination on all aspect. Many algorithms for association rule mining have been proposed in the literature. One of the most vital algorithms is the advanced work presented by [2] in which the usage of the monotone a priori property is introduced to reduce the computational cost of frequent itemsets. [3] used the hash table structure to reduce the candidate space. [4] Proposed the OPUS approach for association rules.

1.2 Big data

Big data is an all-inclusive word for any group of data sets which are very large and compound [6]. The development to larger data sets is due to the extra information derivable from examination of a large set of related data, allowing associations to be found to "spot business trends, avoid diseases, combat crime and so on [7–10].

Big data needs exceptional technologies to professionally cope with large number of data within bearable elapsed times [10]. Although these systems have achieved a great deal in finding association rules, there are some problems appearing with the growing of the data. The first one is the calculating cost. While these approaches are applied in a very large dataset, it needs to examine databases much more times to form frequent item sets. The next one is the effectiveness of the exploring of the rules. With the data rising up, some rules will be missed and the correctness of the results attained will be decreased. The inspiration of this research is to discover an association rules mining algorithm which can address the faults revealed above in big data.

2. Proposed system

2.1 Pre processing

The very large data set is taken from data set repository.it is converted in to the desired format which is accepted by the software. Available data set cannot be used as input, so it is preprocessed in preprocessing null are removed, every field is given a data type as integer. Since every transaction has transaction ID. Every item is identified by item number. The software wants the input in binomial format but the input is in nominal form.so we have to convert nominal to binomial format. The first record in dataset is marked as column name to identity each column

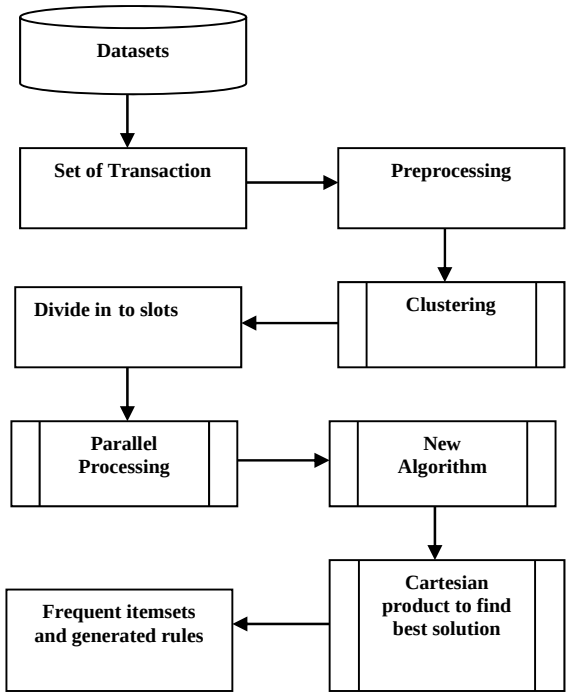

Figure 1: Architecture of Propose Work

2.1 Clustering

Clustering is used to group and divide the datasets in to several subsets. Various clustering algorithms are used we have used K-mean clustering algorithm .Grouping is done to combine items which are similar to each other. The algorithm uses the position of the center in the attributes. This is otherwise called as centroid. The output of the clustering is cluster model. Here in our project we have we have clustered according to the items like vegetables, electronic devices; cloths etc. to find interesting rules from the very large datasets or otherwise called as big data.

2.2 Partitioning and parallel processing

The datasets are now divided into partitions with the help of clustering algorithm. Every partition is given as individual input to generate frequent item sets with the help of our algorithm. Multi core programming is done so that every processor takes input separately and process independently. The output of the every processor is taken as output and given input to Cartesian product. The output is generated frequent pattern from which interesting rules are mined.

The work aims at efficient utilization of all the cores existing in the system with less time consumption and also stabilities the job among them. Full-fledged usage of system resources and load balance can be attained by perfect scheduling and providing effective parallel algorithms.

It undergoes various steps to do parallel processing. At first the entire datasets from cluster is divided in to slots. Each slot has undergone various processes to obtain the rules. They are as follows

Step 1 Set the parameters: chromosome's head length, the operation rate of each operator, the function set and the terminal set;

Step 2 Initialize the population P, set the number of slots as well as MAX and MIN, and then divides the chromosomes into these slots equally;

Step 3 Use the new algorithm on each slot and take the solution set$(OP(P(t+1),\infty))$into a kernel set of slots as the computation result in this generation;

Step 4 Find the similarities among the best individuals from each slot; combination will be operated according to Algorithm 1;

Step 5 Select the top 10% individuals depending on the fitness from the set $P(x+1)k$; and then make a

Cartesian product over them to obtain the solution set$(OP(P(x+1),\infty))$which is regarded as the outputs

of current slots.

Step 6 If the global optimal solution is found or the preset maximum number of generations is reached, end the process. Otherwise, go to Step 3.

Algorithm 1 slot Fusion

① Merge all individuals of two slots (which are supposed as slot1 and slot2, and before fusion, their

sizes are s1 and s2 respectively) to be fused into slot1; go to ②;

② Examine the similarity of slot1 so as to exclude isomorphic chromosomes to combine, and then obtain

the size *s*'1 of the modified slot1; go to ③;

③ If *s*'1 is bigger than MAX, redundant individuals will be selected out; then adjust

s'1 and go to ④; else go to ⑤;

④ If *s*'1is smaller than MIN, the new individual will be introduced randomly until the smallest size is satisfied; then adjust *s*'1 and go to ⑤;

⑤ Create slot2 randomly, and make the equation fulfill $s'2 = s1 + s2 - s'1$.

2.4 New algorithm for frequent itemsets

A new algorithm is developed to find the frequent itemsets from the large datasets. This alogirthm uses permutation to find frequent itemsets.

Drawing-Growth Algorithm. Taking the transaction database in Table 1 as an example, the mining process with Painting-Growth algorithm is as follows.

(1) The algorithm scans the database once, obtains two-item permutation sets of all transactions, and paints peak set (the peak set is a set of all different items in transaction database). Here we take the first transaction as an example.

TID	Items
1	I1,I2,I3,I4,I5
2	I2,I3,I5
3	I3,I5,I4
4	I1,I3,I4

Table 1: Transaction Database

The first transaction is {I1, I2, I3, I4, I5}.

Two-item permutation sets after scanning the first transaction are

{(I1, I2),(I1, I3),(I1, I4), (I1, I5),(I2, I2),(I3, I1), (I4, I1),(I5, I1),(I2, I3), (I2, I4),(I2, I5),(I3, I2),(I4, I2),(I5, I2),(I3, I4), (I3, I5),(I4, I3),(I5, I3), (I4, I5),(I5, I4)}.

Other transactions are similar to the first transaction. The peak set after scanning database is {I1, I2, I3, I4, I5}.

(2) After obtaining the peak set and two-item per-mutation sets of all transactions, the algorithm draws the association image according to two-item permutation sets and peak set. It links the two items looking in each two-item permutation. When the permutation seems again, the link count increases by 1. The association image is shown in Figure 2.

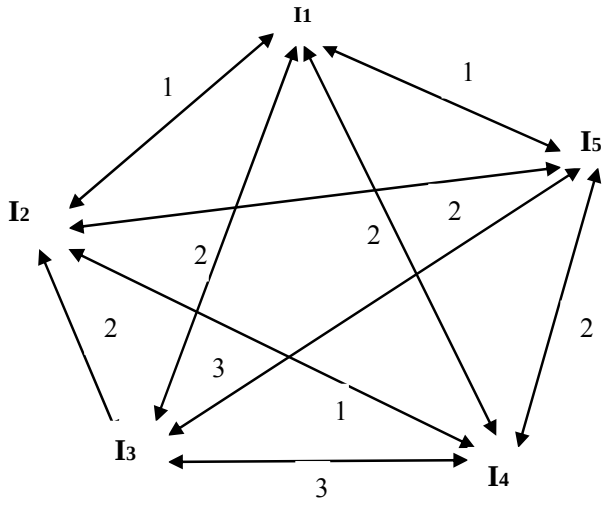

Figure 2: The association Image

(3) According to the association picture, algorithm exploits the support count to remove unfrequented associations. We can get the frequent item association sets as follows: {I1(I3:2,I4:2);I2(I3:2,I5:2); I3 (I1:2,I2:2,I4:3,I5:3); I4(I1:2,I3:3, I5:2); I5 (I2:2,I3:3,I4:2)}.

Here we take the item A as an example. I1 (I3 :2,I4:2) shows that the support count of two-item set (I1 I3) is 2 and the support count of two-item set (I1 I4) is 2. Other items are similar to item I1.

(4) According to the frequent item association sets, we can get all two-item frequent sets of this transaction database: {(I1,I3):2;(I1,I4):2;(I2,I3):2;(I2,I5):2;(I3,I4):3;(I3 ,I5):3;(I4,I5):2}.

(5) According to the frequent item association sets {I1(I3:2,I4:2);I3(I1:2,I2:2,I4:3,I5:3);I4(I1:2,I3:3,I 5:2)}, we can get a three-item frequent set {(I1,I2,I4):2}.

And according to the frequent item association sets {I1(I3:2,I4:2);I3(I1:2,I2:2,I4:3,I5:3);I4(I1:2,I3:3,I 4:2)}, we also can get a three-item frequent set {(I2,I3,I5):2}.

Similarly,according to the frequent item association sets {I3(I1:2,I2:2,I4:3,I5:3);I4(I1:2,I3:3,I5:2);I5(I2:2,I3: 3,I4:2)}, we get a three-item frequent set {(I3,I4,I5):2}.

(6) At this point, we get all frequent item sets. The algorithm pseudocode is as follows.

Algorithm 3 (Drawing-Growth).

Input. Transaction database, minimum support count: 2

Output. All frequent item sets

```
(1) HashMapping⟨Str, int⟩ hm0;
    //define a HashMappnig set hmp0
(2) List⟨String⟩ list,list0;  //define the List set list,list0
(3) List⟨String⟩ permutation();  //scan the transaction database, execute two-item arranging to each trans-action, return list
(4) draw(Graphics g)  //drawing method
(5) String[] s=null, x=null;  //define String[] s, x (6) String z, y;
(7) HashMapping⟨Str, HashMapping⟨Str, int⟩⟩ hm=null;  //define a HashMapping set hm
(8) For (int k=0; k<list. size(); i++) (9) {
(10) s = list.get(k).split(“,”);  //let list.get(k) to a String[]
(11) drawingLine(A[0].x, A[0].y, A[1].x, A[1].y);  //draw a line between A[0] and A[1]
(12) HashMappnig⟨Str, HashMapping⟨Str, int⟩⟩ count(drawLine()); //count the drawing line and return the item associations to hm
(13) }
(14) Iterate it = hmp.keySet().iterator; //define key set iterator of hm
(15) z = it.next();  //let the key in key set of hmp to z
(16) Iterator it0 = hmp.get(z). keySet(). iterator;  //define the key sets iterator in value sets of hm
(17) y = it0.next();  //let the key in key sets of value sets of hm to y
(18) if(hmp.get(z).get(y)<minsup∗N) //if the value in value sets of hm less than minimum support count
(19){it0.remove();}
//remove the unfrequented item sets
(20)List⟨String⟩combination(hmp.get(z).keySet ()); //combination the key sets in value sets based on key z of hm, return list0
(21) for(int j=0;
j<list0.size();j++)
(22) {
(23) x = list0.get(j).split(“,”);
(24)
if(count(hm.contain(z+“,”+list0.get(j))==1+x.
length)) //if the count of item sets in hm equal with
(25)
    {hmp0.put(z+“,”+list0.get(j),value)};//sa
    ve the item sets and its support count in hm0
(26) }
(27) return hmp0;//gain all frequent item sets
(28) super.drawComponents(g);
//execute drawing method
```

3.2. N Drawing-Growth Algorithm. The concept of N Drawing-Growth algorithm is related to the Drawing-Growth algorithm, but with altered implementation technique. N Drawing-Growth algorithm eliminates the drawing steps. The mining procedure of N Drawing-Growth is as follows.

(1) The procedure scans the database just the once and develops two-item permutation sets of all transactions.

(2) Then, the process counts every permutation in two-item permutation to acquire all the item sets.

(3)Later, the algorithm eliminates infrequent associations permitting to the support count and acquires frequent itemsets.

(4) Lastly, it gets entirely frequent item sets giving to the frequent item association sets. Mining finishes.

From the above procedures it can be realized that the N Drawing-Growth algorithm is the eliminating of drawing steps type of Drawing-Growth. The application approaches are different: Drawing-Growth algorithm introductions java.awt and javax.swing, applying mining through calling super.paintComponents(g); N Drawing-Growth algorithm only permits instantiation of a class in main function to appliance

2.5 Implementation of new algorithm in E-Commerce.

The newly generated algorithm is implemented in E-Commerce platform called woo commerce and created a

commercial website. Our algorithmworks as a recommendation system which gives recommendation to the buyers in a commercial websites. Using these recommendation buyers can easily purchase the related and relevant product. Our algorthim gives the best recommendation to buyers.

3. Tests and evaluation

A sequence of experiments has been completed to estimate the performance of our algorithm for association rules mining. We initially evaluate the performance of our new algorithm through a database of measurement for environmental weight. After that, we compared the performance of our new algorithm with FP-Growth and Apriori on a standard big datasets. All tests were implemented over a desktop computer with arrangements: CPU (Intel Core i5-540M processor, 3MCache, 2.53 GHz); RAM (8 GB), Operating System (Windows 7 Professional).

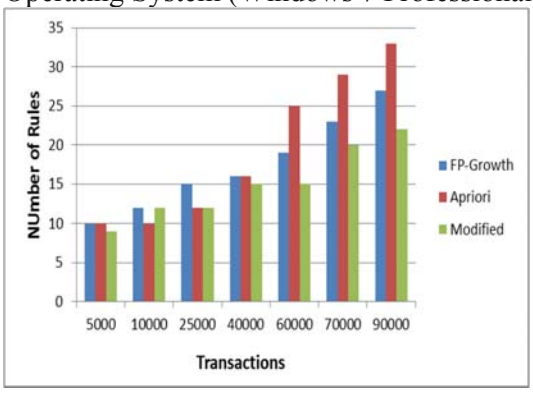

Fig 3. Number of rules generated

The database D consists of 90,000 transactions with the item attribute set *An* containing n attributes
{*a*1,*a*2,*a*3,…,*a*n}. Let min_*support*=0.1 and min_*confident*=0.5. The database is established with the interesting rules.
To evaluate the performance of three methods while dealing with big data problems, experiments were carried out separately with seven number of transaction: 5000, 10000, 25000, 40000, 60000, 70000 and 90000.

As shown in Fig. 3, our new algorithm performs better than FPGrowth and Apriori do in association rules mining problems. With the increasing number of transactions, the differences amongst our algorithm, FP-Growth and Apriori increase significantly. Our algorithm generates 21 rules for 90000 datasets where as apriori generates 34 rules and FP-Growth generates 27 rules. So we can say our new algorithm is more efficient when compared to other two algorithms.

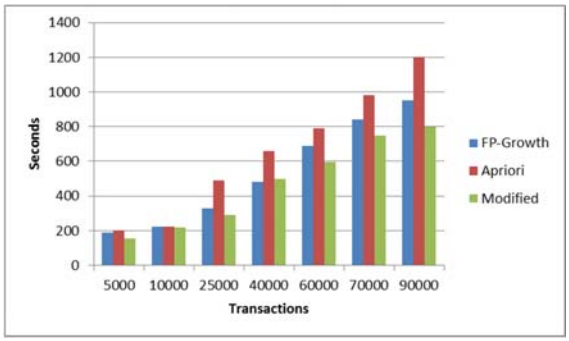

Fig 4. Time taken to generate rules

It also can be indicated that our new algorithm executes extremely faster than FP-Growth and Apriori do with the number of transactions increasing when achieving the above successes. When the transactions grows up from 10,000 to 25,000, the execute time of our algorithm increase from 200 to 300 s, while that of FP-Growth is from 200 to 485.4 s and for Apriori is from 210 to 514.3 s.

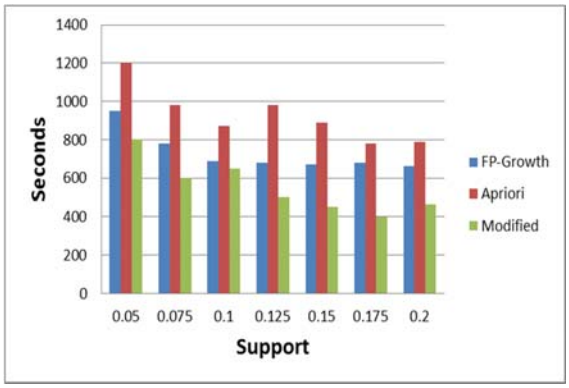

Fig 5. Time taken to generate rules for various support values

It also can be indicated that our new algorithm executes extremely faster than FP-Growth and Apriori do with the number of support values increasing when achieving the above successes. When the support grows up from 0.005 to 0.075, the execute time of our algorithm decreases from 800 to 600 s, while that of FP-Growth is from900 to 785.4 s and for Apriori is from 1200 to 985.4 s.

4. Conclusions and future work

Analyses of big data demand new approaches to fulfill the demands of applications. To overcome the disadvantages of traditional approaches, we examined the possibility and efficiency of a new

approach to solving association rules mining problems in big data. The new method combined with clustering, partitioning, parallel processing and new algorithm is designed to explore association rules in large amounts of data set. The process of our method begins with the people initialization, preprocessing and then divides individuals into each slot. Our new algorithms are applied on each slot and then some of the sub-slots are fused according to the similarity of best individuals. After that, Cartesian product process is applied in the kernel set of slots to generate improved outputs.

A series of experiments have been carried out to have an in-depth investigation on the performance of the proposed algorithm. For the big data set, ten sets of trials have first been completed to assess the efficiency of our new algorithm. The results display that the number of rules found using our algorithm can attain a high value and the execution time is fewer than the other approaches. In comparison with FP-Growth and Apriori, the number of rules obtained using our algorithm are always lesser than the other approaches. Particularly the convergence speeds of our algorithm are higher than those of FP-Growth and Apriori (e.g., with transactions of 90,000 datasets, the execution time is 800.6 s (our algorithm) vs. 985.4 s (FP-Growth) vs. 1200.3 s (Apriori)). Our algorithm is implemented in E-Commerce platform called Woo Commerce for recommendation to the users.

For future work, we will study other procedures of interestingness for rules such as Collective strength and Conviction [11]. Another exciting work is to use our algorithm on fuzzy association rules and the categorical characteristics [12, 13].

References

1. J. Hipp, U. Guntzer, G. Nakhaeizadeh, “Algorithms for Association Rule Mining-A General Survey and Comparison”, ACM SIGKDD Explorations Newsletter, vol. 2, pp. 58-64, 2000.

2. R. Agrawal, T. Imielinski, A. Swami, “Mining association rules between sets of items in large databases”, Proceedings of the 1993 ACM SIGMOD International Conference On Management Of Data, vol. 22, pp. 207-216, 1993.

3. K. Sotiris, K. Dimitris, “Association rules mining-A recent overview”, Proceedings of International Transactions on Computer Science and Engineering-GESTS, vol. 32, pp. 71-82, 2006.

4. C.S. Kanimonzhi Selvi, A. Tamilarasi, “Mining Association rules with Dynamic and Collective Support Thresholds”, International Journal of Engineering and Technology, vol. 1, pp. 427–438, 2009.

5. L. Szathmary, P. Valtchev, A. Napoli, “Finding Minimal Rare Itemsets and Rare Association Rules, Knowledge Science”, Engineering and Management-Springer, vol. 6291, pp. 16-27, 2010.

6. Lizhe,W., Ke, L., Peng, L., et al.: IK-SVD: dictionary learning for spatial big data via incremental atom update. Comput. Sci. Eng.**16**(4), 41–52 (2014)

7. Barnes, J.: Data, data, everywhere. ITS Int. **20**(1), 44–49 (2014)

8. Deng, Z., Wu, X., Wang, L., et al.: Parallel processing of dynamic continuous queries over streaming data flows. IEEE Trans. Parallel Distrib. Syst. (2014).

9. Chen,D.,Wang, L.,Wu, X., et al.:Hybrid modeling and simulation of huge crowd over a hierarchical grid architecture. Future Gener. Comput. Syst. **29**(5), 1309–1317 (2013)

10. Chen, D.,Wang, L., Zomaya, A., et al.: Parallel simulation of complex evacuation scenarios with adaptive agentmodels. IEEE Trans. Parallel Distrib. Syst. (2014).

11. Piatetsky-Shapiro, G.: Discovery, analysis and presentation of strong rules. In: Piatetsky-Shapiro, G., Frawley,W.J. (eds.) Knowledge Discovery in Databases, pp. 229–248. AAAI Press (1991)

12. Kuok, C.M., Fu, A., Wong, M.H.: Mining fuzzy association rules in databases. ACM Sigmod Rec. **27**(1), 41–46 (1998)

13. Chen, D., Li, X., Wang, L., Khan, S., Wang, J., Zeng, K., Cai, C.: Fast and scalable multi-way analysis of massive neural data. IEEE Trans. Comput. (2014).