Artificial Life and Robotics (2018) 23:420–427
https://doi.org/10.1007/s10015-018-0437-y

ORIGINAL ARTICLE

Association rule mining algorithms on high-dimensional datasets

Dongmei Ai[1] · Hongfei Pan[1] · Xiaoxin Li[1] · Yingxin Gao[1] · Di He[2]

Received: 30 March 2018 / Accepted: 9 May 2018 / Published online: 30 May 2018

Abstract

The science of bioinformatics has been accelerating at a fast pace, introducing more features and handling bigger volumes. However, these swift changes have, at the same time, posed challenges to data mining applications, in particular efficient association rule mining. Many data mining algorithms for high-dimensional datasets have been put forward, but the sheer numbers of these algorithms with varying features and application scenarios have complicated making suitable choices. Therefore, we present a general survey of multiple association rule mining algorithms applicable to high-dimensional datasets. The main characteristics and relative merits of these algorithms are explained, as well, pointing out areas for improvement and optimization strategies that might be better adapted to high-dimensional datasets, according to previous studies. Generally speaking, association rule mining algorithms that merge diverse optimization methods with advanced computer techniques can better balance scalability and interpretability.

Keywords Data mining algorithms · Association rule mining · High-dimensional datasets · Frequent itemset mining

1 Introduction

Association rules mining (ARM), an important branch of data mining, has been extensively used in many areas since Agrawal first introduced it in 1993 [1]. In general, ARM can be seen as a method aimed at discovering groups of items that co-occur with high frequency. In contrast to other data mining methods involved with statistical models, ARM can extract possible relationships between variables from huge datasets with little prior knowledge according to co-occurrence features. When applied to biomedical data, ARM can obtain rules that provide a better understanding of biological associations among different covariates or between covariates and response variables. Bioinformatics techniques have been developing with increased speed and so have several high-throughput biotechnologies, such as genomic microarray and Next Generation Sequencing (NGS). High-dimensional data are often encountered in areas such as medicine, where DNA microarray technology can produce a large number of measurements at once, high-dimensional data are the data with anywhere from a few dozen to many thousands of dimensions. A dataset is a collection of data. Most commonly a dataset corresponds to the contents of a single database table, or a single statistical data matrix. It follows that a concern among researchers has been the efficient and effective discovery of latent information underlying huge amounts of data. As a possible solution to this problem, ARM has been extensively applied in this field. A typical application of ARM on such high-throughput datasets is gene association analysis (GAA) [2, 3], in which the goal is to exploit the relationships among different genes based on corresponding expression levels.

Data from these high-throughput techniques often share in common the feature of high dimensionality. For example, gene expression data typically take the form of an $N \times M$

✉ Dongmei Ai
aidongmei@sina.com

Hongfei Pan
18813128340@163.com

Xiaoxin Li
851482181@qq.com

Yingxin Gao
gaoyingxin16@outlook.com

Di He
15810251026@163.com

[1] School of Mathematics and Physics, University of Science and Technology Beijing, No.30, Xueyuan Road, Beijing 100083, China

[2] Computer Science, City University of New York Graduate School and University Center, CUNY, 365 Fifth Avenue, New York 10016, USA

matrix, where each row of the matrix represents a sample, and each column corresponds to the expression level of a certain gene. The number of genes in a given study can be in the thousands, while the number of specimens is generally dozens or hundreds.

Such high dimensionality is also true for other kinds of biomedical datasets, e.g., Operational Taxonomic Unit (OTU) abundance datasets that have different levels of extra environmental factors in metagenomics analysis [4], as well as multiple datasets, including mRNA/miRNA expression data and Copy Number Variations (CNV) data from The Cancer Genome Atlas (TCGA) project [5].

Based on the high dimensionality of such datasets, the use of traditional methods of association rules mining directly applied to these datasets could result in unsatisfactory performance [6]. To improve performance brought by high-dimensional datasets, multiple specialized algorithms have been proposed in the last decade.

2 Mining algorithms on high-dimensional datasets

2.1 Basic association rule mining algorithms

Apriori, the first ARM algorithm, was proposed by Agrawal [7], and it successfully reduced the search space size with a downward closure Apriori property that says a k-itemset is frequent only if all of its subsets are frequent. The Apriori algorithm is characterized by a feature called candidate generation whereby $(k + 1)$-candidate-itemsets are generated iteratively by combining any two frequent k-itemsets which share a common prefix of length (k-1). Further computation of the supports of each candidate itemset is then performed to determine if the candidate is frequent or not. Finally, the algorithm terminates if no more frequent itemsets can be generated.

Based on the standard Apriori algorithm, several improved variations were proposed. The performance-enhancing strategies include the hashing technique [8], partitioning technique [9], sampling approach [10], dynamic counting [11], and incremental mining [12]. As previous studies demonstrated, these Apriori-based approaches achieved good performance when the dataset was sparse and the patterns discovered were of short length. However, such methods suffer nontrivial costs caused by generating huge numbers of candidate itemsets and extra scans over the datasets for support computation.

We then saw the emergence of new algorithms like FP (frequent pattern)-growth without candidate itemsets [13]. First, an FP-tree, which retains information associating the itemsets, is constructed according to the frequency of 1-itemset. Next, patterns of different lengths are generated by concatenating suffix patterns, starting from frequent 1-pattern to least frequent itemsets, with frequent patterns from a conditional FP-tree, which is a subtree consisting of the set of prefix-paths in the FP-tree co-occurring with the suffix, recursively. In other words, this method only involves frequent patter $(k + 1)$-itemset n growth instead of Apriori-like generation-and-test. In this sense, then, it applies a partitioning-based divide-and-conquer strategy, and efficiency studies demonstrated that this method has substantially reduced search time. Subsequently, multiple algorithms were proposed as extensions to the FP-growth approach, such as generating frequent itemset in a depth-first manner [14], mining with devised hyperstructure [15], pattern-growth mining by traversing FP-tree-like structure in both directions (top-down and bottom-up) [16], and pattern-growth mining with tree structure in an array-based implementation form [17, 18]. Recursive searches on the tree could result in enormous costs in certain cases; nevertheless, such methods did lay a solid foundation for the further application of tree structure in association rule mining algorithms.

Apriori and FP-growth both adopted a horizontal format for mining frequent itemsets. In contrast, Zaki proposed an Equivalence CLASS Transformation (Eclat) algorithm employing a vertical data format [19]. Eclat also utilized Apriori's candidate generation property of $(k + 1)$-itemset candidates. In Eclat, however, the support computation of candidate can be done by just intersecting the sample id sets of the corresponding frequent $(k + 1)$-itemset. More simply stated, the support of any itemset can be obtained directly from the vertical sampleID without any further computation. Thus, additional scanning of the original dataset can be saved, again reducing the cost of search time.

Generally speaking, finding all frequent itemsets of a specific dataset can be regarded as a process consisting of search space traversal, itemset support computation, and search path pruning. A common strategy of traversing the search space includes breadth-first search (BFS). For example, in Apriori, the frequent $(k + 1)$-itemset is not generated until all frequent $(k + 1)$-itemsets have been discovered. Another common strategy is depth-first search (DFS). For example, in FP-growth, longer frequent patterns are generated recursively until no more can be done. Common strategies for support computation include counting (e.g., Apriori, FP-growth) and intersection (e.g., Eclat). In sum, the methodologies adopted by the three basic association rule mining algorithms described (Apriori [7], FP-growth [13] and Eclat [19]) serve as landmarks for the development of association rules mining and constitute the basis for subsequent association algorithms.

2.2 Maximal frequent itemset mining and frequent closed itemset mining

According to the Apriori Property, it is obvious that (*k*-2) subsets (except itself and φ) of a certain frequent *k*-itemset are also frequent. Such characteristic will result in massive unnecessary redundancy of frequent itemsets. To limit the redundancy, two alternative concepts were advanced, namely maximal frequent itemset mining and frequent closed itemset mining [20].

Many algorithms were developed to mine these two categories of itemsets. For example, MaxMiner, the very first study on maximal frequent itemset mining, was proposed in 1998 [21]. Based on Apriori, MaxMiner adopted a breadth-first search (BFS) strategy and reduced the search space by both superset and subset frequency pruning. As another efficient maximal frequent itemset mining method, MAFIA improved support counting efficiency by adopting vertical bitmaps to compress the transaction id list [22].

For frequent closed itemset mining, numerous methods have been proposed since 1999, when A-Close, an Apriori-based frequent closed itemset mining approach, was reported [23]. CLOSET explored frequent closed itemset mining based on FP-tree structure [24]. Another typical frequent closed itemset mining approach is CHARM, which adopted a hybrid search strategy, known as the diffsets technique (compact form of *tID* list information), and a hash-based "non-close item disposal" approach to enhance both computation and memory efficiency [25]. In addition, AFOPT-close presented a method which can adaptively use three different structures, including array, AFOPT-tree (FP-tree like) and buckets, to represent conditional databases according to their respective densities [26]. To integrate previous effective approaches and some newly developed techniques, CLOSET+ was proposed [27]. After thorough performance studies on diverse datasets, CLOSET+ was considered as one of the most efficient methods at the time.

Previous studies have shown that algorithms of these two categories are usually more efficient against previous iterations. However, maximal frequent itemset has a critical defect in that the supports of its subitemsets may be different from its own. This would, in turn, result in extra scans over the dataset for support computation and its ultimate unfitness for rule extraction. Frequent closed itemset mining does not encounter such problems, essentially because all subsets of a certain frequent closed itemset must have precisely the same support as that of the frequent closed itemset. Furthermore, frequent closed itemsets can be regarded as a compressed form of the complete frequent itemsets without information loss. Based on these features and properties, we can conclude that frequent closed itemset mining is more likely to play a vital role in the development of association rules mining.

In summary, frequent closed itemsets can provide analytical power equivalent to that of complete frequent itemsets, but with much smaller size. Substantial approaches have verified the higher efficiency and better interpretability obtained by frequent closed itemset mining. However, most of the above-mentioned algorithms adopt the column-enumeration strategy. Therefore, when applying such approaches over high-dimensional datasets, the search space will tend to expand exponentially, according to the feature size, thus making computational cost prohibitive. Therefore, it is easy to see why most of the algorithms discussed thus far cannot be applied to high-dimensional datasets, again underscoring the need to develop algorithms applicable to high-dimensional datasets to keep pace with advancements in sequencing and computer technology.

2.3 Algorithms applicable to high-dimensional datasets

From previous sections, we can see that the applicability of the above-mentioned algorithms to high-dimensional datasets is limited. In this section, we will discuss approaches better able to meet this challenge.

Among them, approaches incorporating frequent closed itemset mining and row enumeration can serve as a possible solution. This idea was first explored in 2003 [28]. Based on data in vertical format, CARPENTER constructs a row-enumeration tree and adopts a depth-first search (DFS) strategy to traverse it. Additionally, several pruning strategies are employed during the search process to cut off the branches incapable of generating frequent closed itemsets. Previous study [28] has shown that CARPENTER gained better performance, compared to its rivals as CHARM and CLOSE+, when applied to high-dimensional microarray datasets [27].

Adopting similar strategies, other methods were developed. For instance, RERII is like CARPENTER, but instead of searching frequent itemsets from the whole original datasets, RERII explored frequent closed itemsets in the opposite direction, starting from the nodes that represent the complete rowsets [29]. This strategy has the potential to enhance overall performance by reducing the cost of searching short rowsets and *1*-item rowsets.

To make CARPENTER more adaptable to more complex datasets, COBBLER integrated the strategies of both CARPENTER and CLOSE+ [30]. Accordingly, COBBLER can dynamically switch between row-enumeration and column-enumeration to meet estimated cost conditions. Its efficiency has been verified in experiments over datasets with high dimensionality and a relatively large number of rows.

TD-CLOSE adopts a top-down row-enumeration search strategy that enables the support of a stronger pruning power against the bottom-up style adopted by CARPENTER [31]. To guarantee closeness during the mining process, an additional closeness checking method was included in TD-CLOSE. Moreover, in 2009, an improved version of TD-CLOSE, called TTD-CLOSE, was proposed [32]. With optimized pruning strategy and data structure, TTD-CLOSE obtained better performance than the original TD-CLOSE.

To extend the applications of frequent pattern mining, new classification methods based on ARM over high-dimensional datasets, such as FARMER and TOPKRGS, emerged [33, 34]. With additional class information attached to the original datasets, both algorithms can extract classification rules in the form of $X \Rightarrow C$, where C is a class label, and X is a set of items. Based on previous analysis, it can be concluded that a rule extracted from frequent closed itemset, consisting of k items in total, also implies the existence of other 2^k-2 rules. To reduce rule redundancy, these two algorithms only extract "interesting" rule groups instead of all rules. Specifically, FARMER adopted the concept of a rule group that consists of a unique upper bound rule and a set of lower bound rules for clustering the complete results of rules, while TOPKRGS just selects the most significant top-k covering rule groups. In addition, FARMER reinforced interestingness measures with Chi square in addition to support and confidence, while TOPKRGS adopts a prefix tree structure to speed up the frequency computation and utilizes a dynamic minimum confidence generation strategy to better-fit different datasets.

To enhance mining efficiency, HDminer employs effective search space partitioning and pruning strategies. HDminer gradually narrows down the search space by pruning off the false-valued cells based on the space partition tree instead of accumulating the true-valued cells like the FP-tree- or enumeration tree-based methods. Owing to fewer false-valued cells compared to true-valued cells, HDminer works much more efficiently than the FP-tree- or enumeration tree-based methods [35]. HDMiner shows superiority, especially on synthetic data and dense microarray data.

To summarize, previous studies have verified the relatively high efficiency of row-enumeration algorithms for mining frequent closed itemset over high-dimensional datasets. However, with the advancement of biomedical data acquisition techniques, the volume of data has grown larger and larger, and the row size of a certain dataset may, therefore, become as large as the column size. In this case, methods such as COBBLER or TD-CLOSE, as described above, may still have trouble handling such large datasets. Consequently, instead of sequential algorithms, increased attention has focused on parallel and distributed algorithms. Actually, parallel association rule mining algorithms were proposed quite early in the 1990s [36, 37]. However, since the effectiveness of these algorithms was challenged by complicated strategies of workload balance, fault tolerance and data distribution, as well as interconnection costs and limited computer hardware capacity at that time, extensive application of such algorithms was suppressed. On the other hand, based on the exuberance over cloud computing and distributed computing techniques, parallelized association rule mining algorithms were revived with the opportunity to show their power. Specifically, as the most recognized large-scale data analysis technique, Hadoop has been broadly utilized in modern biomedical studies [38, 39]. Characterized by mapper and reducer functions [40], the Hadoop MapReduce framework is especially good at processing gigabytes, or even terabytes, of data. Moreover, by hiding details of underlying controls, Hadoop can enable users to just concentrate on algorithm design. All of these features make Hadoop a novel promising candidate to propel the development of ARM in the "big data" era.

Typical examples of adapting Apriori on Hadoop MapReduce include SPC (Single Pass Counting), FPC (Fixed Pass Combined-Counting) and DPC (Dynamic Pass Counting) [41]. These algorithms share common procedures of distributing data to different mappers and parallel counting supports, but differ in candidate generation. Typically, SPC generates frequent itemsets of only a single length after one phase of MapReduce, but FPC and DPC generate frequent itemsets with different lengths after phases. In addition, as the names suggest, and are fixed parameters in FPC, while in DPC, they are dynamically determined by the number of generated candidate itemsets at each MapReduce phase. Other Hadoop-based Apriori algorithms, which work in a similar manner, but different forms of implementation, were also proposed [42, 43].

To solve the problem that the traditional association rules mining algorithm has been unable to meet the mining needs of large amount of data in the aspect of efficiency and scalability, take FP-Growth as an example, the algorithm of the parallelization was realized in based on Hadoop framework and Map Reduce model. It can be better to meet the requirements of big data mining and efficiently mine frequent item sets and association rules from large dataset [44]. MRFP-Growth (MapReduce Frequent Pattern Growth) is also implementing to solve the problem of discovering frequent patterns with massive datasets. The efficiency and performance of this method have been increased compared with other mining algorithms [45]. Also, implementations of Eclat on MapReduce were proposed, such as Dist-Eclat, focusing on speed acceleration, and its optimized version BigFIM which adopts hybrid approaches incorporating both Apriori and Eclat, thus making BigFIM better suited to very large datasets [46]. Experiments on real datasets have proven their scalability. An MapReduce algorithm for mining closed frequent itemsets was implemented, as well [47].

Another noteworthy approach is PARMA [48]. It applies parallel mining algorithms to randomly selected subsets of the original large dataset. Owing to its random sample property, its mined results can be considered as the approximation of the exact results according to the whole dataset. The approximation quality was verified by both both statistical analysis and real-time application.

Based on CARPENTER, a new algorithm called PaMPa-HD was developed. This algorithm adopts the depth-first search process, as well, but the process is broken up into independent subprocesses to which a centralized version of CARPENTER is applied so that it can autonomously evaluate subtrees of the search space. Then the final closed itemsets of each subprocess can be extracted in order to compute the whole closed itemset result [49]. Since the subprocesses are independent, they can be executed in parallel by means of a distributed computing platform such as Hadoop.

To achieve compressed storage and avoid building conditional pattern bases, FiDoop was brought forward. FiDoop utilizes a frequent itemset ultrametric tree. In FiDoop, the mappers independently decompose itemsets, the reducers perform combination operations by constructing small ultrametric trees, and the actual mining of these trees is performed separately, which can speed up the mining performance for high-dimensional datasets analysis [50]. Extensive experiments using real-world celestial spectral data indicate that FiDoop is scalable and efficient.

In addition to the huge computation cost, it is typical for the size of derived patterns from high-dimensional datasets to be enormous. Such growth of derived patterns makes their effective use difficult. Therefore, a new methodology aimed at mining approximate or representative patterns, instead of full-scale patterns, appeared as a solution. The most typical approach, known as Pattern-Fusion, is based on a novel concept called core pattern. Pattern-Fusion is able to discover approximate colossal patterns, i.e., the colossal pattern mining algorithm based on pattern fusion improve seed pattern selection method, which is select pattern that the distant big, rather than random seed pattern [51].

Recently, the Graphics Processor Units (GPU) has emerged as one of the most used parallel hardware to solve large scientific complex problems. An approach benefits from the massively parallel power of GPU by using a large number of threads to evaluate association rule mining was proposed [52]. Then a new algorithm called MWBSO-MEGPU was proposed. This method combine both GPU and cluster computing to improve a Bees Swarm Optimization (BSO) metaheuristic. Several tests have been carried out to evaluate this approach. The results reveal that MWBSO-MEGPU outperforms the HPC-based ARM approaches in terms of speed up when exploring Webdocs instance [53].

3 Discussion

All algorithms reviewed as applicable approaches for high-dimensional datasets are summarized below in Table 1.

The performance evaluation of association rule mining algorithms raises two major concerns. The first is scalability, which refers to the ability of an algorithm to handle a

Table 1 Overall compilation of association rule mining algorithms on high-dimensional datasets

Methods	Category	Feature	Reference
CARPENTER	Row-enumeration closed pattern	Bottom-up	[28]
RERII		Top-down	[29]
COBBLER		Hybrid of CARPENTER&CLOSE+	[30]
TD-CLOSE		Top-down	[31]
TTD-CLOSE		Top-down	[32]
FARMER	Classification rules	Rule group	[33]
TOPARGS	Row-enumeration closed pattern	TopK-rules	[34]
HDMiner	Space partition tree	Search space partition	[35]
SPC/FPC/DPC	Hadoop-based	Apriori	[41]
PFP		FP-growth	[44]
MRFP		FP-growth	[45]
Dist-Eclat		Eclat	[46]
BigFIM		Hybrid of Apriori and Eclat	[46]
An Improved Algorithms		Closed pattern	[47]
PARMA		Approximate pattern	[48]
PaMPa-HD		Sub-process	[49]
Fidoop		Frequent items ultrametric tree	[50]
Pattern-Fusion	Colossal pattern mining	Pattern fusion	[51]
Bioarm-Gpu-Ga	GPU-based	Bio-inspired	[52]
MWBSO-MEGPU		Bio-inspired	[53]

large amount of data in a suitably efficient way. The other is interpretability, or the capacity to translate the results to real-world issues, such as biological meaning.

With respect to scalability, our primary focus in this paper, numerous approaches to efficiently process high-dimensional datasets have been proposed. with advantage of advanced computer technology and seemingly unlimited cloud computing resources, as well as "big data" processing techniques, parallelized association rules mining might be the most promising candidate to lead further development of association in the new "big data" era. More efforts should primarily concentrate on a more appropriate data distribution model, more efficient mining methods that take better advantage of the key-value feature and a more reliable load balance scheme. Furthermore, the idea adopted by Pattern-Fusion whereby approximate or representative patterns are extracted instead of full-scale patterns is a promising methodology to address the high-dimensionality problem. By employing such methodology, the mining process realizes a cost savings by identifying shorter patterns. It also yields a much smaller size of result sets, consisting of longer patterns preferred in practical use. For example, in gene expression analysis, longer patterns are usually more favorable. Still, approximation quality needs to be guaranteed to avoid major latent information loss, which may involve more statistical analysis and theoretical proof.

Interpretability is another critical issue in biomedical research. Typically, incorporating previously known biological knowledge with association rule mining algorithms is seen as providing better biologically meaningful results [6]; however, in this review, we have suggested that taking too much knowledge into account might lower the ability of ARM to obtain undiscovered rules because the algorithm would tend to fit the biological knowledge more. Additionally, such approach may increase the cost and, in turn, reduce scalability. Instead, toolkits that include fuzzy set theory, genetic algorithms, ant colony algorithms, and particle swarm optimization and other heuristic algorithms can be utilized to optimize association rule mining algorithms for better interpretability. Such optimization methods have been mainly used over quantitative data without a typical pre-discretization procedure. For example, fuzzy set theory can be used to generate fuzzy association rules with more practical meanings, genetic algorithms to dynamically specify appropriate support threshold and ant colony algorithms to reduce the scale of the result rules [54, 55]. As also suggested in this review, association rule mining algorithms that merge diverse optimization methods might, when combined with improved computer techniques, provide researchers with tools of more practical value.

4 Conclusion

Generally speaking, ARM has been widely utilized in bioinformatics studies. ARM can be used to identify the most relevant covariates in a certain biological process and thus construct the underlying intrinsic latent network. When applied over high-dimensional datasets, many older methods cannot manage the issue of high dimensionality. Many of the most recent methods have been proposed to address this problem, each with its merits and faults, but no perfect solution has been achieved. For better usage in this area, new algorithms that can better balance scalability and interpretability are still in demand.

Acknowledgements This research is partially supported by National Natural Science Foundation of China (61370131).We thank David Martin for editorial assistance in English language.

References

1. Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. Acm Sigmod Rec 22(2):207–216
2. Creighton C, Hanash S (2003) Mining gene expression databases for association rules. Bioinformatics 19(1):79–86
3. Liu YC, Cheng CP, Tseng VS (2011) Discovering relational-based association rules with multiple minimum supports on microarray datasets. Bioinformatics 27(22):3142–3148
4. Kunin V, Copeland A, Lapidus A et al (2008) A bioinformatician's guide to metagenomics. Microbiol Mol Biol Rev 72(4):557–578
5. Network CGA (2012) Comprehensive molecular characterization of human colon and rectal cancer. Nature 487(7407):330–337
6. Alves R, Rodriguez-Baena DS, Aguilar-Ruiz JS (2010) Gene association analysis: a survey of frequent pattern mining from gene expression data. Brief Bioinform 11(2):210–224
7. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceeding 20th international conference on very large data bases, VLDB, pp 487–499
8. Park JS, Chen M-S, Yu PS (1995) An effective hash-based algorithm for mining association rules. Acm Sigmod Rec 24(2):175–186
9. Savasere A, Omiecinski ER, Navathe SB (1995) An efficient algorithm for mining association rules in large databases. In: International conference on very large data bases, pp 432–444
10. Toivonen H (1996) Sampling large databases for association rules. VLDB, pp 134–145
11. Brin S, Motwani R, Ullman JD et al (1997) Dynamic itemset counting and implication rules for market basket data. Proc Sigmod 26(2):255–264
12. Cheung DW, Wong CYHan J, Ng VT (1996) Maintenance of discovered association rules in large databases: An incremental

updating technique. In: Proceedings of the twelfth international conference on data engineering, pp 106–114

13. Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: Proceeding of the 2000 ACM SIGMOD international conference on management of data, pp 1–12
14. Agarwal RC, Aggarwal CC, Prasad V (2001) A tree projection algorithm for generation of frequent item sets. J Parallel Distrib Comput 61(3):350–371
15. Pei J, Han J, Lu H et al (2007) H-Mine: Fastand space-preserving frequent pattern mining in large databases. IIE Trans 39(6):593–605
16. Liu J, Pan Y, Wang K et al.(2002) Mining frequent item sets by opportunistic projection. In: Proceedings of the eighth ACM Sigkdd international conference on knowledge discovery and data mining, pp 229–238
17. Grahne G, Zhu J (2003) Efficiently using prefix-trees in mining frequent itemsets. In: Proceeding IEEE ICSM workshop on frequent itemset mining implementations
18. Grahne G, Zhu J (2005) Fast algorithms for frequent itemset mining using fp-trees. IEEE Trans Knowl Data Eng 17(10):1347–1362
19. Zaki MJ (2000) Scalable algorithms for association mining. IEEE Trans Knowl Data Eng 12(3):372–390
20. Han J, Cheng H, Xin D et al (2007) Frequent pattern mining: current status and future directions. Data Min Knowl Disc 15(1):55–86
21. Bayardo RJ Jr (1998) Efficiently mining long patterns from databases. ACM Sigmod Int Conf Manag Data 27(2):85–93
22. Burdick D, Calimlim M, Gehrke J (2001) MAFIA: a maximal frequent itemset algorithm for transactional databases. In: International conference on data engineering, pp 443–452
23. Pasquier N, Bastide Y, Taouil R et al (1999) Discovering frequent closed itemsets for association rules. Lect Notes Comput Sci 1540:398–416
24. Pei J, Han J, Mao R (2000) CLOSET: an efficient algorithm for mining frequent closed itemsets. ACM SIGMOD workshop on research issues in data mining and knowledge discovery, pp 21–30
25. Zaki MJ, Hsiao C-J (2002) CHARM: an efficient algorithm for closed itemset mining. In: Proceedings of the 2002 SIAM international conference on data mining, pp 457–473
26. Liu G, Lu H, Yu JX et al.(2003) AFOPT: an efficient implementation of pattern growth approach. In: Proceeding of the Icdm Workshop
27. Wang J, Han J, Pei J (2003) ClOSET+: Searching for the best strategies for mining frequent closed itemsets. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, pp 236–245
28. Pan F, Cong G, Tung AK et al.(2003) Carpenter: Finding closed patterns in long biological datasets. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, pp 637–642
29. Cong G, Tan K-L, Tung AK et al.(2004) Mining frequent closed patterns in microarray data. In: 2004 ICDM'04 fourth IEEE international conference on data mining, pp 363–366
30. Pan F, Tung AK, Cong G et al.(2004) COBBLER: combining column and row enumeration for closed pattern discovery. In: Proceedings 16th international conference on scientific and statistical database management, pp 21–30
31. Liu H, Han J, Xin D et al.(2006) Mining frequent patterns from very high dimensional data: a top-down row enumeration approach. In: Proceedings of the 2006 SIAM international conference on data mining, pp 282–293
32. Liu H, Wang X, He J et al (2009) Top-down mining of frequent closed patterns from very high dimensional data. Inf Sci 179(7):899–924
33. Cong G, Tung AK, Xu X et al (2004) FARMER: finding interesting rule groups in microarray datasets. In: Proceedings of the 2004 ACM SIGMOD international conference on management of data, pp 143–154
34. Cong G, Tan K-L, Tung AK et al (2005) Mining top-k covering rule groups for gene expression data. In: Proceedings of the 2005 ACM SIGMOD international conference on management of data, pp 670-681
35. Xu J, Ji S (2014) HDminer: efficient mining of high dimensional frequent closed patterns from dense data. In: 2014 IEEE international conference on data mining workshop, pp 1061–1067
36. Agrawal R, Shafer JC (1996) Parallel mining of association rules. IEEE Trans Knowl Data Eng 8(6):962–969
37. Zaki MJ (1999) Parallel and distributed association mining: a survey. IEEE Concurr 7(4):14–25
38. Ferraro Petrillo U, Roscigno G, Cattaneo G, Giancarlo R (2017) FASTdoop: a versatile and efficient library for the input of FASTA and FASTQ files for MapReduce Hadoop bioinformatics applications. Bioinformatics 33(10):1575–1577
39. O'Driscoll A, Daugelaite J, Sleator RD (2013) 'Big data', Hadoop and cloud computing in genomics. J Biomed Inf 46(5):774–781
40. Dean J, Ghemawat S (2010) MapReduce: a flexible data processing tool. Commun ACM 53(1):72–77
41. Lin M-Y, Lee P-Y, Hsueh S-C (2012) Apriori-based frequent itemset mining algorithms on MapReduce. In: Proceedings of the 6th international conference on ubiquitous information management and communication, p 76
42. Li N, Zeng L, He Q et al.(2012) Parallel implementation of apriori algorithm based on mapreduce. In: IEEE 13th ACIS international conference on software engineering, artificial intelligence, networking and parallel and distributed computing, pp 236–241
43. Kovacs F, Illés J (2013) Frequent itemset mining on hadoop. In: Computational cybernetics (ICCC), 2013 IEEE 9th international conference on IEEE, pp 241–245
44. Fu C, Wang X, Zhang L, Qiao L (2018) Mining algorithm for association rules in big data based on Hadoop. In: AIP conference proceedings. AIP Publishing: 040035
45. Al-Hamodi AA, Lu S (2016) MRFP: discovery frequent patterns using MapReduce frequent pattern growth. In: Network and information systems for computers (ICNISC), 2016 international conference on. IEEE: 298–301
46. Moens S, Aksehirli E, Goethals B (2013) Frequent itemset mining for big data. In: Big Data, 2013 IEEE international conference on IEEE: pp 111–118
47. Gonen Y, Gudes E (2016) An improved mapreduce algorithm for mining closed frequent itemsets. In: Software science, technology and engineering (SWSTE), IEEE international conference on: 2016. IEEE: 77–83
48. Riondato M, DeBrabant JA, Fonseca R et al (2012) PARMA: a parallel randomized algorithm for approximate association rules mining in MapReduce. In: Proceedings of the 21st ACM international conference on Information and knowledge management, pp 85–94
49. Apiletti D, Baralis E, Cerquitelli T et al (2015) PaMPa-HD: a Parallel MapReduce-based frequent pattern miner for high-dimensional data. In: IEEE international conference on data mining workshop, pp 839–846
50. Xun Y, Zhang J, Qin X (2016) Fidoop: Parallel mining of frequent itemsets using mapreduce. IEEE Trans Syst Man Cybern Syst 46(3):313–325
51. Wang Z. (2014) The colossal pattern mining algorithm based on pattern fusion., Tianjin Polytechnic University, Tianjin
52. Djenouri Y, Bendjoudi A, Djenouri D, Comuzzi M (2017) GPU-based bio-inspired model for solving association rules mining problem. In: Parallel, distributed and network-based processing (PDP), 2017 25th Euromicro International Conference on. IEEE: 262–269
53. Djenouri Y, Djenouri D, Habbas Z (2018) Intelligent mapping between GPU and cluster computing for discovering big association rules. Appl Soft Comput 65:387–399
54. Mangalampalli A, Pudi V (2013) FAR-HD:A fast and efficient algorithm for mining fuzzy association rules in large high-dimensional

datasets. Parallel implementation of apriori algorithm based on mapreduce Fuzzy Systems, pp 1–6

55. Martínez-Ballesteros M, Bacardit J, Troncoso A, Riquelme JC (2015) Enhancing the scalability of a genetic algorithm to discover quantitative association rules in large-scale datasets. Integr Comput Aided Eng 22(1):21–39