https://www.sworldjournal.com/index.php/swj/article/view/swj06-01-012
DOI: 10.30888/2663-5712.2020-06-01-012

УДК 004.032.24

RESEARCH OF THE EFFICIENCY OF THE APRIORI GROUP ALGORITHMS ON DIFFERENT DATABASE SIZES

Polovynka Olha / Половинка О.Л.
ORCID: 0000-0002-0575-4587
Dmitrieva Olga / Дмитрієва О.А.
d.t.s., prof.
ORCID: 0000-0001-8921-8433
Donetsk National Technical University, Pokrovsk, Shibankova square, 2, Ukraine

Abstract. *The results of the work analysis of associative rule search algorithms for handling Big data are presented in the paper. The most famous modifica-tions of Apriori algorithms for finding associative rules are considered. The re-sults of the research of the efficiency of linear algorithms Apriori and Apriori-TID on the data of different volumes using computers with standard RAM vol-ume are presented. The feasibility of using the AIS algorithm and its parallel implementation on the Hadoop MapReduce framework is substantiated. A comparison of the linear implementation of the AprioriTID algorithm and the parallel implementation of the AIS algorithm is provided. The conclusions about the effectiveness of using a parallel algorithm to solve a formulated prob-lem was given.*

Keywords: *associative rules search algorithms, Apriori, AIS, linear implemen-tation, parallel implementation, Hadoop, MapReduce.*

Introduce

Recently, there has been a steady increase in digital information, the efficient processing of which can significantly increase the profits of the enterprise. Therefore, methods of solving problems related to the need to process large data sets and search for hidden patterns are in demand.

For analysis of big data Neural networks, decision trees, fuzzy models, regression and cluster analysis methods are often used. Usually these methods are used to process structured data presented as arrays containing the values of the attributes and the output parameters of data sampling. At the same time, the processing of unstructured data, in which each storage unit can be represented by a finite number of attributes, is impossible by such methods [1].

To process large arrays of unstructured data and solve these problems, it is advisable to use associative rule search methods. It is allow to reveal in the available data new regularities of a kind of "if - that". Based on the identified dependencies, it is possible to synthesize rules bases that are understandable to experts in the applied fields.

Currently, a large number of associative rules search algorithms and their modifications are presented in scientific literature. But each of them has its own peculiarities and is used for solving a certain class of tasks. There is no universal algorithm.

In addition, linear search methods for associations are not effective, because in the exponential growth of unstructured data, the time required to process large amounts of information is unacceptably long.

Therefore, the analysis of existing developments and their modification in order to select an effective algorithm for finding associative rules in terms of providing a

given level of support and reliability of the result required to process big data without being tied to the scope of this data, is an urgent task. This task have practical and scientific value.

Big Data gathering, storage and processing are increasingly being addressed by re-searchers. Associative rule search algorithms are used in many fields of activity. Initially, they was used to find typical shopping patterns in supermarkets, but now associative rules are used to restore images [2], identify relationships between habitats of biological species, cultivate folk resources, analyze the activities of public organizations, in medical diagnostics [3] and even in education [4].

The search for associative rules for processing large arrays of unstructured data and their linear implementations is given attention in [5, 6].

In [7 - 10], the possibility of implementing a priori algorithms on the MapReduce framework is considered and a scheme for implementing the algorithm on the specified platform is given.

The purpose of the work is to develop and optimize associative rules search algorithms for processing large volumes of transactional data and to obtain performance metrics for distributed processing with existing software.

1. Research of linear implementations efficiency

Basic algorithms for finding associations and basic principles for their formation are described in [6]. It is noted that the main evaluation characteristics of the algorithm are the notion of support and confidence.

The most common association search algorithm is the Apriori algorithm, which is a modification of the AIS proposed in 1993 by R. Agrawal [11]. There are a number of modifications to apriori group algorithms that reduce the search time for associative rules. The main difference between existing of apriori group algorithms lies in the different candidate generation strategies.

To determine the peculiarities of the functioning of linear a priori algorithms on large data sets the implementation of Apriori and AprioriTID algorithms from the GitHab site [12, 13], whose pseudocodes are given in [14], were used in the work.

The experiments were performed on transactions data, which were presented as a table with poorly structured data. The peculiarity of such data is their versatility, that is, similar tables can be obtained from many fields of activity. The main thing is the fact of the transaction and the indefinite number of parameters in the row. This data can be obtained not only from the sales network (pharmaceutical campaign sales), but also in the analysis of user requests on the sites, the process of random signal generation, the work of the call center and more.

At the first stage of the study, the initial table was transformed to form a structured data file suitable for processing by a priori algorithms.

The next step was to study the AprioriTid algorithm [13], the results of which are given in Table 1, and in Fig. 3.

Table 1 shows that the algorithm is characterized by an acceptable implementation time, even for a large number of transactions. However, when working with large amounts of data (20 000 000 transactions), there was a shortage of RAM, which made further processing impossible [15].

Table 1

The results of the AprioriTid algorithm

№	*Number of items*	*Run time of the algorithm, h*
1	100000	00:00:01
2	500000	00:00:05
3	1000000	00:00:28
4	2000000	00:01:32
5	4000000	00:03:22
6	8000000	00:05:04
7	14000000	00:10:19
8	20000000	00:13:20
9	26000000	MemoryError

The third group of experiments was carried out using a linear implementation of the classical Apriori algorithm [12]. It also ran out of memory, but at much less database volumes than AprioriTid. This situation is due to the fact that both algorithms load the entire volume of transactions into the computer memory.

In order to remedy these shortcomings, a series of experiments were conducted on a modified version of the classic Apriori algorithm, which does not require loading transactions into the computer memory. The results of the work are given in Table 3. According to the results of testing, it was determined that the proposed implementation does not require a large amount of RAM, which is an advantage, but it is rather slow even using relatively small amounts of data.

Despite the fact that the original version of the Apriori algorithm (without modifications) does not oblige the developer to download the entire set of transactions into the computer memory, this solution can significantly speed up the algorithm and is used in most existing developments. However, the experiments showed that the time results of the algorithms on data volumes that exceed several million transactions are unacceptable. In addition, with low support indicators, and in the presence of a large number of unique elements in transactions, the number of combinations mainly in the second iteration is too large to fit in the RAM of ordinary computers. At the same time, with this implementation of the classical Apriori algorithm, it does not depend on the amount of RAM, an excessive execution time of the algorithm is obtained.

In order to eliminate certain shortcomings, in the work modifications of a priori algorithms aimed at the possibility of parallel implementations was proposed.

2. Parallel implementation on the MapReduce framework

In the field of parallel data processing, the most popular tool is the MapReduce software model offered by Google Inc. It implies an explicit division of the data processing algorithm into two functions: map and reduce, copies of which can be executed in parallel at multiple computing nodes. This allows for efficient parallel processing of data from a single source with high scaling. It was this technology that was chosen to implement the parallel model of the associative rule search algorithm. Principles of the MapReduce framework have been considered in the works of many re-searchers, including [12].

The Hadoop cluster was deployed for the experiment. Amazon Web Services (AWS) cloud-based service was selected to host the results.

When trying to implement classic Apriori algorithm as well as AprioriTID to work on large volumes of data at the existing facilities, a message about RAM error, because these algorithms are extremely sensitive to the amount of computers memory.

As a result of the analysis, it was decided to use the AIS algorithm, which does not require loading of transactions into the computer memory, and therefore does not depend on its volume. The algorithm repeatedly runs through the database. The resulting set consists of unique sets of elements. It changes (expands) each pass.

In this case, the support of the found sets of elements is measured on each pass. These sets, which are called candidate sets, are derived from the tuples in the database and the element sets that are recorded in the resulting set.

A counter is created for each set of elements; it saves the number of transactions in which the corresponding set of elements has appeared. The counter is reset to zero when a new set is created.

Initially, the frontier set consists of only one element, which is empty. At the end of the run, support for a candidate set is compared to a given minimum support (min-support) to determine if it is the largest set of elements. At the same time, it is deter-mined whether this set of elements should be added to the frontier set for the next pass. The algorithm terminates when the frontier set becomes empty. The element counter support is stored when the itemset is added to a large/frontier set [13].

Let T - the specified transaction database; L - sets of result elements for which the support level exceeds the preset minimum: count(c)/dbsize>minsupport; t is a tuple, that is, a unique set of elements; C - validity, the need to include an element in the set (provided that previously there was no element in the set); F is the result set; f is the current set of elements.

The operation of the algorithm can be schematically depicted as follows (Fig. 1).

A feature of the AIS algorithm is that the candidates of multiple sets are generated and counted while scanning the database. Each transaction is checked for the presence of large sets, which were detected in the previous stage.

These features make the algorithm work more slowly, but allow you not to use the excess amount of computer memory, that is important in the conditions of the experiment.

The process of operation of the AIS algorithm implemented on the Hadoop MapReduce framework is presented in fig. 2.

In order to simplify the analysis of the operation of parallel and non-parallel implementations of a priori algorithms, it was decided to limit itself to three iterations. The main reason for this decision was the initial adaptation to the planned implementation of the algorithm in the Hadoop MapReduce framework. Implementing algorithms on MapReduce cannot work iteratively, so a separate task run must be performed for each iteration.

Comparative characteristics of the linear implementation of the AprioriTID algorithm and the parallel implementation of the AIS algorithm are given in Table 2

and Fig. 3.

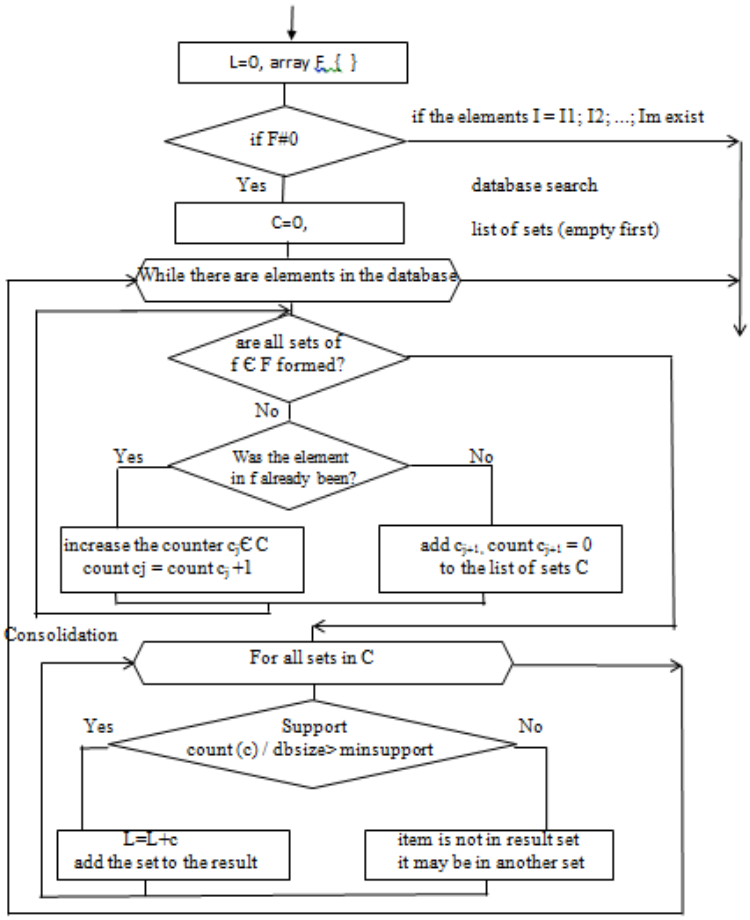

Fig. 1. Scheme of the AIS algorithm

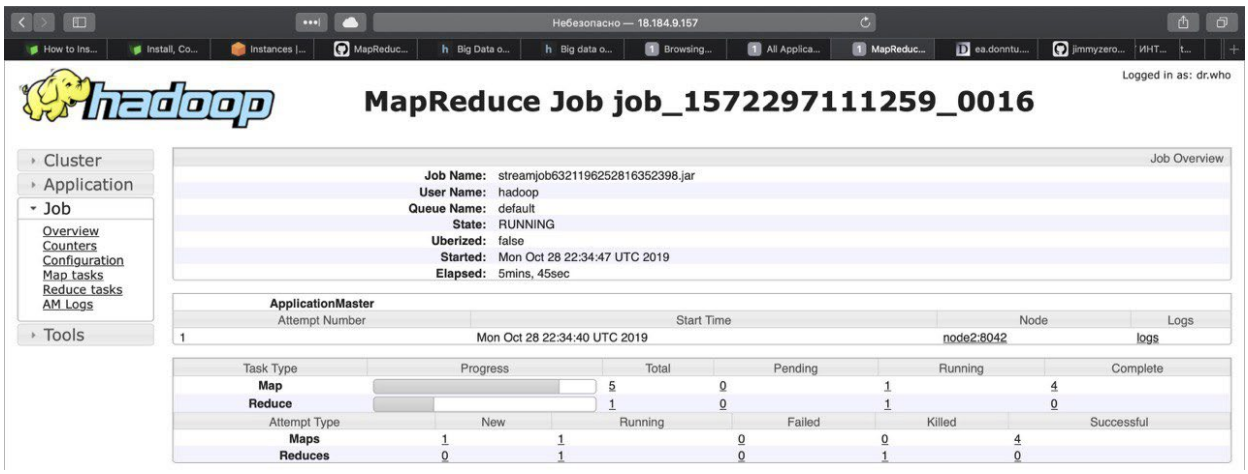

Fig. 2. Process of AIS algorithm operation on Hadoop MapReduce framework

Table 4 shows that during linear calculations (experiments 1-6), the running time of the AprioriTID algorithm is shorter. However, in the 7th experiment, the running time of the AIS algorithm dropped sharply, due to the fact that the volume of the database had reached sufficient size to effectively data parallelize on the MapReduce framework.

Thus, even though AIS is a slower associative rule search algorithm than Apriori and AprioriTID algorithms, its parallel implementation allows for large volumes of data to be processed in practice, even on machines with a small amount of RAM.

Table 2

Comparison of AprioriTID and AIS Algorithm Results.

№	Number of items	AprioriTID (linear), h	AIS (parallel), h
1	100000	00:00:01	0:01:21
2	500000	00:00:05	0:01:22
3	1000000	00:00:28	0:01:29
4	2000000	00:01:32	0:02:09
5	4000000	00:03:22	0:02:36
6	8000000	00:05:04	0:05:12
7	14000000	00:10:19	0:04:06
8	20000000	00:13:20	0:05:51
9	50000000	MemoryError	0:14:39
10	100000000	MemoryError	0:28:14
11	200000000	MemoryError	0:56:13

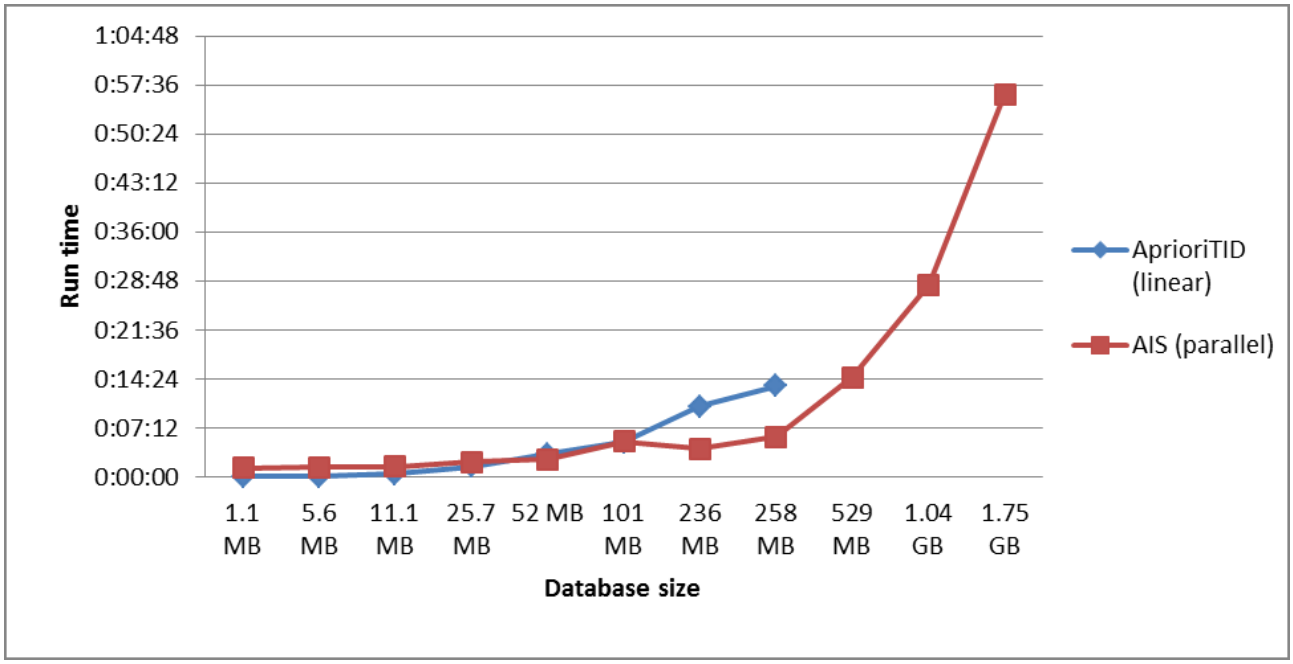

Fig 3. Operation of AprioriTID (linear implementation) and AIS (parallel implementation) algorithms

Summary and conclusions

The paper substantiates the use of associative rules search algorithms for working with unstructured data of large volumes. It should be noted that Apriori algorithm is the most popular associative rules search algorithms. The basic modifications of Apriori group algorithms for searching for associative rules are considered.

The performance of Apriori and AprioriTID linear algorithms has been investigated on various volumes using computers with standard memory capacity. It is found that when processing big data in the stated conditions, the work of algorithms be-comes impossible due to the lack of RAM.

To solve this problem, we analyzed the algorithms from Apriori group. According to the results, the AIS algorithm was selected to process large volumes of data on computers with low memory. It is slower, but does not require loading transactions into the computer RAM. And there before it does not depend on the amount of memory.

To solve the problem of performance of the AIS algorithm, we proposed its parallel implementation on the Hadoop MapReduce framework. Results of operation of the parallel algorithm on the large data are given, as well as the comparative characteristics of the linear implementation of the AprioriTID algorithm and the parallel implementation of the AIS algorithm. A significant time gain was obtained when processing data larger than 128 MB. Also, the operation of the parallel algorithm on large data sets is tested. It is worth noting that in the presence of more powerful computers for cluster deployment, it is possible to implement Apriori and AprioriTID algorithms on the Hadoop MapReduce framework in parallel.

References.

1. Zayko T.A., Oleynik A.A., Subbotin S.A., “Associative rules in data mining”, Vestnik NTU "KHPI, no.39 (1012), pp. 82-96, 2013.

2. Belim S.V., Mayorov-Zilbernagel A.O., Seliverstov S.A., “Using associative rules to restore noisy images”, Vestnik Omskogo universiteta. Series "Information Technolo-gy", no. 4, pp. 197-200, 2013.

3. Billig V.A., Ivanova O.V., Tsaregorodtsev N.A., “The construction of associative rules in the task of medical diagnosis”, Programmnyye produkty i sistemy, no. 2 (114), 2016. https://cyberleninka.ru/article/n/postroenie-assotsiativnyh-pravil-v-zadache-meditsinskoy-diagnostiki.

4. Maslova N.O., Polovinka O.L., “Application of methods of search of associations at creation of tests on information security” [Zastosuvannya metodiv poshuku asotsiatsiy pry stvorenni testiv z informatsiynoyi bezpeky], Zbírnik naukovikh prats' Donets'kogo natsíonal'nogo tekhníchnogo uníversitetu. Seriya: “Computer science is a cybernetics and a calculating technique.”, no.1 (29), pp 47-53, 2019.

5. Subbotin S.A., Oleynik A.A., Hoffmann E.A., Zaitsev S.A., Oleinik Al.A. Intelligent information technology for designing automated systems for diagnosing and recogniz-ing images: monograph.. LLC "Company Smith" (2012), 317.

6. Adamo J.-M. Data mining for association rules and sequential patterns: sequential and parallel algorithms. Springer-Verlag (2001), 259.

7. S. Singh, R. Garg and P. K. Mishra, "Review of Apriori Based Algorithms on MapRe-duce Framework", 2014 International Conference on Communication and Computing (ICC - 2014), pp. 593–604, 2014.

8. Jyoti Yadav, Neha Sehta, “Implement Mapreduce Apriori Algorithm to Generate Fre-quent Itemsets”, International Journal of Computer Applications, Vol. 179., no.38, pp. 7-10, 2018.

9. Singh S, Garg R, Mishra PK., “Performance optimization of MapReduce-based Aprio-ri algorithm on Hadoop cluster”, Computers & Electrical Engineering, pp. 348-364, 2018. https://arxiv.org/ftp/arxiv/papers/1807/1807.06070.pdf.

10. Dmitrieva O.A., Polovinka O.L., “Algorithmic support for parallel methods of search-ing for associative rules”, Zbirnyk naukovykh prats′ Donets′koho natsional′noho tekhnichnoho universytetu. Series: "Computer Engineering and Automation", no.1 (31), pp. 62-69, 2018.

11. Agrawal R., Imielinski T., Swami A., “Mining association rules between sets of items in large databases”, In Proc. of the ACM SIGMOD Conference on Management of Data, 1993. https://rakesh.agrawal-family.com/papers/sigmod93assoc.pdf.

12. Online implementation of the Apriori algorithm. https://github.com/asaini/Apriori

13. Online implementation of AprioriTID algorithm. https://github.com/ymoch/apyori

14. Agrawal Rakesh, Ramakrishnan Srikant, “Fast algorithms for mining association rules”, Proc. 20th int. conf. very large data bases, VLDB, Vol. 1215, pp. 487-499, 1994.

15. Polovinka O., Nikulin D., Dmitrieva O., “Research of speed of work of a priori algorithms on data of large volumes” [Doslidzhennya shvidkodiї robots apiornyh algorithms on tribute to the great obligations]. Science and production. Series: Information Technology, no 22, pp. 246-253, 2020.

Анотація. *У статті представлені результати аналізу роботи асоціативних алгоритмів пошуку правил для обробки великих даних. Розглянуто найвідоміші модифікації алгоритмів Апріорі для пошуку асоціативних правил. Представлені результати дослідження ефективності лінійних алгоритмів Apriori та Apriori-TID на даних різних обсягів із використанням комп'ютерів зі стандартним об'ємом оперативної пам'яті. Обґрунтовано доцільність використання алгоритму AIS та його паралельного впровадження в рамках Hadoop MapReduce. Наведено порівняння лінійної реалізації алгоритму AprioriTID та паралельної реалізації алгоритму AIS. Дано висновки щодо ефективності використання паралельного алгоритму для вирішення сформульованої задачі.*

Ключові слова: *алгоритми пошуку асоціативних правил, Apriori, AIS, лінійна реалізація, паралельна реалізація, Hadoop, MapReduce.*

Науковий керівник: д.т.н., проф. Дмітрієва О.А.

Стаття відправлена: 16.12.2020 г.