ASSOCIATION-RULES-BASED DATA IMPUTATION WITH SPARK

Zhaowei Qu, Jianru Yan and Sixing Yin

Beijing University of Posts and Telecommunications, Beijing 100876, China
{zwqu, yanjianru, yinsixing}@bupt.edu.cn

Abstract: Due to technical bottlenecks and errors caused by artificial operation, the problem of incomplete data always exists in big data research. Traditional data imputation algorithms incur high complexity and the accuracy cannot reach the desired level. At the same time, analysis and computation involved in mass data makes limitation of traditional algorithms and computing platform more noticeable. In this paper, we propose a data imputation method based on Apriori algorithm, and implement the corresponding algorithm on the distributed computing system built with Spark. The experimental results show that the proposed algorithm outperforms a traditional data imputation algorithm in terms of efficiency and accuracy.

Keywords: Association rules; Data preprocessing; Spark; Distributed algorithm

1 Introduction

Along with the flourish of mobile Internet and wireless sensor networks, the amount of data generated by the network system is experiencing explosive growth. The data contains a variety of information which existing techniques are unable to dig out. At the same time, due to technical limitations in data collection and processing, analyzing and mining the data also leave a variety of problems unsolved. Therefore, dealing with missing fields in the original data and calculating massive data has become new hot topics.

1.1 Incomplete data imputation

The presence of missing data is a recurrent problem in the large database analyses, where there are many inevitable reasons, for example: for sensitive material, importers refuse to provide relevant information; missing part of the data in imputing data or the value of input exceeds the acceptable range of system and so on. Dealing with the problem of missing data properly has become an important research in data preprocessing.

In order to make better use of data with missing parts, researchers put forward some methods to predict the missing field, and use the predicted results to replicate the missing parts. There are three common imputation methods: (1) maximum (minimum) imputation method means choosing the maximum (minimum) value of data sets to replicate the missing part; (2) arbitrary value imputation method means selecting a random number between maximum and minimum to replicate the missing parts ; (3) mean value imputation method means, apparently, replicating the missing parts by average or median of data sets [1,2].

978-1-5090-1256-5/16/$31.00 ©2016 IEEE

1.2 Association rule algorithm

Data mining is an effective way to solve the problem of information overload, which extracts information or knowledge which people are interested in from a large number of data. Association rule mining is an important branch of data mining research. In 1993, Agrawal et al. proposed the important association rule mining in the field of machine learning [3].The earliest use of association rules is based on the breadth first algorithm, including Apriori, AprioriTid [4] and DHP [5]. Researchers have conducted a lot of research on this issue, while most of them just focus on improving the existing algorithms.

1.3 Distributed memory computing model

Spark is an open source distributed cluster computing system [6] developed by UC Berkeley AMPLab，which supports Java, Scala, python, R and other language interfaces. Using resilient distributed datasets (RDD), it effectively solves the problem of low computational performance of traditional MapReduce model in dealing with the iterative calculation of the external storage device. RDD mainly includes two kinds of operations: transforms and actions. Compared to the MapReduce model, it provides a more flexible operation, which can greatly simplify the construction of the program model based on spark. In 2012, Matei Zaharia proposed a large distributed cluster which was named as Discretized Streams[7]. In dealing with the large distributed cluster, this system can provide an efficient and fault tolerant model. Up to now, spark system has made remarkable achievements in various fields, such as stream processing, graph mining, machine learning, structured data query, etc.

1.4 Contribution

After reviewing the existing imputation methods, we mainly focus on two specific aspects here: (1) improving the overall efficiency of imputation algorithms by less frequently accessing database; (2) taking advantage of spark system in large scale iterative computations. In a word, the algorithm proposed in this paper not only advances filling preciseness, but also improves storage efficiency of data structure and time efficiency of mining algorithm, and finally the problem of massive data

computation has been solved by applying the imputation algorithm to a distributed system.

This paper is structured as follows: in Section 2, we put forward an algorithm for dealing with discrete data imputation and optimize it; Section 3introduces the distributed processing of the algorithm in spark which includes the construction of DAG logic diagram and the computation efficiency optimization, then presents a complete set of filling solution of missing fields in large data sets;Section 4 introduces the experimental results of the algorithm and makes a comparison the original one;Section 5 presents the conclusion and future work.

2 Imputation algorithm based on Apriori

In this paper, incomplete data imputation is the key issue should be solved, and a imputation algorithm based on improved Apriori is proposed. One of this algorithm's novelty is that not only the complete data is used, but also the incomplete data are used when the rules be created. The definition and properties of the Apriori algorithm are as follows:

Assume $I = \{I_1, I_2, ..., I_m\}$ is a item-sets which has different m kinds of items, and database $D= \{T_1, T_2,..., T_n\}$ which has n records, each record has a unique label denoted as T. Meanwhile，each T has several different items, $T=\{I_1,I_2,...,I_k\}$ and $T \in I$. An item set includes k kinds items which called(k)item-set. And an association rule is a form of "A=>B", with the understanding that $A \in I$, $B \in I$ and $A \cap B$ is empty. When the support degree of a item-set is higher than min-sup, this items-set is frequent and when the confidence is larger than the min-conf, a new association rule be created.

Property: Non-empty sub-sets of frequent must be frequent. So when X is a frequent (k+1)item-sets, then the count of k-items frequent is at least k +1.

2.1 Data Pre-processing

The original data sets usually contain two types of data: quantitative and categorical data，for example: user's gender and phone planbelongto categorical data, while user's age is one kind ofquantitative data. In the pre-processing stage, we transform continuous data, discrete data and multidimensional items into Boolean data. Methods for quantitative association rule mining mainly include: (1) discretization-based; (2) Statistics-based; (3) non-discretization-based.Generating quantitative association rules that based on statisticsmust set up the effective object properties. According to the algorithm, dispersing the categorical and continuous attributes, which can be realized by equaling the internal width, equaling frequency or methods based on entropy or clustering and so on. In order to use the existing association rule mining algorithm to get frequent item sets in binary data, we usually map the discrete interval to a binarynonsymmetric properties. The difficulty of this method lies in how to determine the width of the interval,because a large interval may lead to a lack of confidence and the loss of some segments in the model,while a narrow interval may get into trouble in the process of generating frequent item sets for lacking of support.

For example: age attributes can be divided into the following section: [12,16], [16,20],[20,24] …[60,80]. In the process, a problem may be encountered is how to determine the interval width, if range of a segment is too large, it may lead to a lack of confidence degree and lost some subdivision of mode. In contrast, if the range is too narrow may result in lack of support and abundant frequent item-sets.

2.2 Self-growth strategy based on Descartes

Traditional Apriori algorithm can only deal with Boolean item sets and does not apply to multidimensional item sets, so converting symmetric binary attributes and category attributes for mining association rules is necessary. For example: $A = \{A_1, A_2, A_3...\}$ is transformed into: A_1 = True/False; A_2 = True/False... Similarly, converting binary attribute (such as the user's sex) into a complete Boolean term set: male =True/False; female =True/False. At the same time, there is a problem with the original algorithm for mining frequent item sets: the original strategy of item sets growth has a large number of computational redundant data, which means traditional strategy of item sets growth consume a large amount of computing resources in summing every attribute and verifying whether it is a frequent item set, while in fact, once one of the category attributes is selected, other attributes will never be selected.In view of this situation, we need to optimize self-growth strategy by using of the Cartesian, and decideif aclassification attribute should be selected.Mathematical description of Cartesian: $A \times B=\{(x,y)|x \in A \wedge y \in B\}$. And according to the prior theorem, using this self-growth strategy to prune the non-frequent (n-1)-item-sets when the item's support is smaller than min-sup.

2.3 Rulesselection strategy

Frequent item-sets'propertiesas follows:

Property 1:Each k-1 dimensional subset is frequency is the necessary condition for thatk-dimensional item-set is frequency.

Property2: If in the k-dimensional item-set $I=\{i_1, i_2, i_3 \dots i_n\}$, there is a item $j \in I, |L_{n-1}(j)<n-1|$, so I is not a frequent item-set. $|L_{n-1}(j)|$is the count of n-1 dimensional frequent item-set L_{n-1} which contains the item J.

Prove: Assume that I is a n-dimensional frequent item-set, sothere are the number of C_n^{n-1}n-1 dimensionalsubsetin L_{n-1}. All n-1dimensional frequent item-sets are concentrated in the I's subsets, the number of occurrences of the item-set i is n-1, so $\forall i \in I$, $|L_{n-1}(j)|>n-1$. Conflict with those conditions, so I is not a frequent item set.

The traditional Apriori needs to generate a set of rules by scan all records of frequent item-sets. At the same time,a strong association rule would be selected when it's confidence is larger than theminimum confidence min-conf. By contrast, there is only one rule would be selected amongthe imputationalgorithm.Therefore, the rule which has the maximum confidence is used to fill themissing value.

2.4 Algorithm Summary

The main proceduresof theincomplete data imputation algorithm based on Apriori are as follows:

(1) To build a query classification attribute dictionary D for theitem-set I;

(2) Preprocessing for the original dataset, andmake discretization and numericalitem totransform to the type of Boolean;

(3) To produce the one-item sets by using the classification attribute dictionary D, at the same time, calculateeach frequency number of corresponding one-item;

(4) Using the self-growth strategy based on Cartesian to calculate the support of (n + 1)item-sets. If the result is larger than *min-sup* which is shown that this item is a frequency, so add to setF;

(5) Loop growth of itemsto get all of the frequencies;

(6) To create rules by using rule selection strategy, expound in this paper, from frequent set F, and collect each rule to the set T;

(7) Missing value would be found in the rules set T when the data imputation system is running.

3 Computing with spark

Currently, much attention has been paid to the research of Hadoop MapReduce distributed system. As far as we know, intermediate results under Hadoopframework inevitably incurfrequently handling I/O operation of external storage device, which greatly draws back computationalefficiency.So this platform is no longer applicable to machine learning and other computation containing multi-iteration. In contrast,Spark,which is based on the DAG logic diagram, avoids the emergence of a large amount of redundant calculation results and reduces the cost of the intermediate results in HDFS. There are two advantages: firstly, completing all calculations just requires once I/O logic device,and the action which is triggered, greatly saves the computation time; secondly, mechanism ofcache in memory supports iterative computation and reducing number of times of I/Oof access. In this sense, Spark is more suitable than Hadoop for the application of machine learning.

There are several portions which cost a long time in the imputation algorithm, such as formattingthe data, counting frequencies and creating rules. We make these portions distributed through the use of DAG logic diagramsuch thatexecution of those procedures can be efficiently improved.

3.1 Design of DAG diagram

According to the imputation algorithm model, those computation intensive modulesaredistributed, and the DAG logic diagram is constructed as shown in Figure 1:

Input: the text file of original data.

Output: frequent item-sets C = {(c_1, v_1), (c_2, v_2), (c_3, v_3) … (c_n, v_n)}, among the c_n denotes the contents of frequent item-sets, and v_n denotes the frequency number of frequent item-sets.

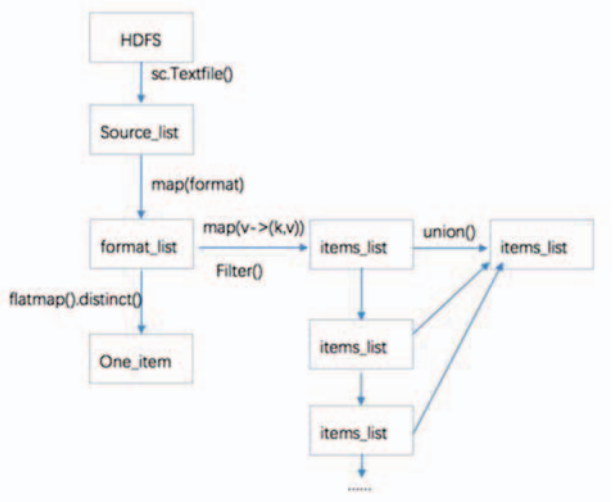

Figure 1 DGA logic diagram

In Figure 1, the original data is recorded on the HDFS (file system of Hadoop framework)as the form of text. The spark engine through initializeclass of scto call the function textFile(), which could load the source data into memory and generate RDD. Several kinds of Spark-API be used when calculating iteration datamainly includes: map()，flatMap()，distinct()，union()，reduceByKey()，filter(), etc. Because the RDD only can be created by other RDD or system shall not be modified, so it's high availability is particularly important. In view of this situation, we need to cache RDD into the memory of drivers.And in order to facilitate the use of RDD，system shouldmake methods to bepersistence. According to logic programs, DAGScheduler of Master divides calculation process into several stages. Then, TaskScheduler will schedule divided stages to run on the drivers in the cluster, and keep in touch with drivers by though periodic "heartbeat".

3.2 Relevant details of computational process

(1) Because in the process of task execution, there are many subtasks need to share the memory, so in view of this kind of situation, a method of using broadcastvariablewould improve the efficiency of task execution.This kind of operation will cache the shared variable to every driver in the actual calculation of cluster andreduce the delay cost of direct exchange between this cluster's different drivers.

(2) In thecalculation process of item-sets, we need to createkey-value type resultsbefore initialize operation. The description as follows: transform the source data {i_1, i_2, i_3…} to the type of <k, v> result <{i1，i2，i3..}, num_{item}>. when the job's stage of map. Scan all records of the original RDD, according to item's attribute to calculatevalue ofnum_{item}in the map stage, and collect the total frequency numberin the reduce stage.

(3) There will be a new RDD and one-item frequency be created after the map(), which would be repeated use. It is fixed in memory for the entire system to improve the performance. In this case,we can use the action ofSpark-API: cache() to save the RDD in memory before the computation of the sub-stage DAG. This scheme can avoidby repeating calculation to generate the RDD.

(4) In the case of serial calculation, thetime complexity of format dataset I is O(n), andthe time complexity of cumulative operation is $O(n^2)$, n is the count of datasets. In order to objectively define the time complexity of the parallel data computation based on spark, C is defined as the communication cost between the cluster nodes. For distributed programs, thetime complexity of format dataset is O(n/p) + C(n/t), andthe time complexity of cumulative operation is $O(n^2/p)$ + C(n/t), p is the count of nodes in the cluster and d represent bandwidth of transmission network.

4 Experiment Results

In order to make result more objective and reliable, the experiment adopts real registration information (excepted sensitive information) of 5 million mobile subscribersas the training set. The data sample is shown in Table I as follow:

Table I The sample data of experiment

user-id	14325	15321
sex	1	2
OS	IOS	Android
start-time	201509	201305
package-type	4	3
GSM	3G	4G
user-level	3	2

Note: sex by 2, 1 respectively on behalf of male and female, the type of package-type, GSM, user-level is countable, the source data use digital to replace.

Experimental environment: building spark cluster and configuring three nodes assigned by 2G memory by virtual machine respectively, where one of the nodes is master and the remaining are slavers; dual core processor: Intel-i3; software environment: Ubuntu15.4; Hadoop2.6; spark 1.3.

Baseline method is the algorithm based on statistics which adoptitem of maximum frequencies, and the recallof test-set is 60%. In the selected test sets, we randomly delete an item of the first row; then let two new sets of data, generated in data imputation algorithm based on association rules and maximum imputation algorithm, replace the deleted one successively. Repeating the process until all rows complete the above operation, we will get two new data sets. Correction rate can be obtained by respectively comparing the two new data sets with the original one, whose rates' cumulative probabilityfunctionare shown in Figure 2, and at the same time,probability density functions of both methods are shown in the Figure 3 and the Figure 4 as follows:

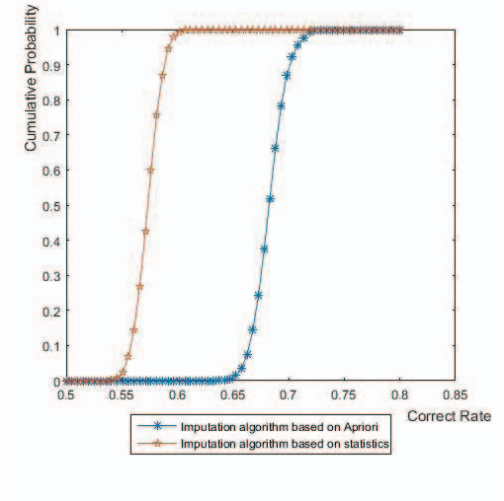

Figure 2 Comparison for cumulative probability function

Note:the horizontal axis represents the correction rate, and the vertical axis represents the rates' cumulative probability.

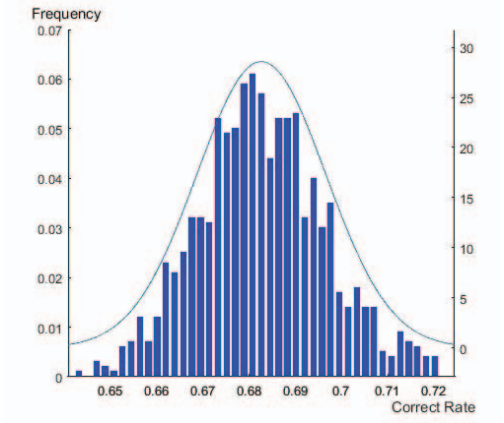

Figure 3 probability density distribution of imputation algorithm based on Apriori

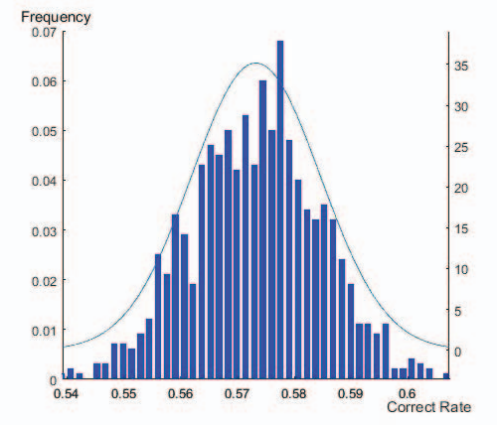

Figure 4 probability density distribution of baseline method

From those figures, it is obvious that the correction rate of data imputation algorithm based on association rules is about 10% higher than that of maximum imputation algorithm on average. In addition, the correction rate of both kinds ofimputation algorithm arealso stable, which fully certify the algorithm in this paper is better thanprevious algorithm.

5 Conclusions

This paper presents a novel approach for imputation algorithm based on Apriori, aiming at the issue of incomplete data. The results showed the feasibility and effectiveness of the method proposed in this paper. In addition, an efficient distributed strategy is used to decrease the planning time of calculation. There are still many problems to be solved in the future work, such as utilizing newly collected data based on the original model and investigating the influence of minimum support degree, etc.

References

[1] Cheng Y,Miao D,Feng Q. Positive approximation and converse approximation in inter-valued fuzzy rough sets[J]. Information Sciences,2011,181(11): 2086-2110.

[2] Meng Z,Shi Z. Extended rough set-based attribute reduction in inconsistent incomplete decision systems[J]. Information Sciences, 2012,204(10) : 44-69.

[3] Agrawal.R. Mining association rules between sets of items in large databases [C], Proc. of the ACM SIGMOD Int. Conf. on Management of Data, Washington D. C., 1993, 207-216.

[4] Agrawal,R.Imielinski,T. and Swami.A. Mining association rules between sets of items in large databases, Proceedings of the ACM SIGMOD Conference on Management of data, May 1993, 207-216.

[5] Park,J. Chen,M. and Yu.P. An effective hash based algorithm for mining association rule, In Proceedings 1995 ACM SIGMOD International conference Management of Data (SIGMOD'95), San Jose, CA. May 1995, 175-186.

[6] Zaharia M, Chowdhury M, Franklin M J, et al. Spark: cluster computing with working sets[C]//Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, Boston, USA, Jun 22-25, 2010. Berkeley, CA, USA: USENIX, 2010.

[7] Zaharia M,Das T,Li H,et al. Discretized streams :an efficient and fault-tolerant model for stream processing on large clusters[C]//Proceedings of the 4th USENIX conference on Hot Topics in Cloud Computing. USENIX Association, 2012:10-10.