ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Detection of Outliers in Large Dataset using Distributed Approach

Jyoti N. Shinde, Prof. D.B.Kshirsagar

PG Student, Department of Computer Engineering, SRES, College of Engineering, Kopargaon, India

Head of Department, Department of Computer Engineering, SRES, College of Engineering, Kopargaon, India

.

ABSTRACT: A distributed method is presented to detect distance-based outliers, based on the concept of outlier detection solving set. In this paper we have introduced a distributed method for detecting distance based outliers in a large datasets. The approach is based on concept of outlier detection solving set [2] which is a small dataset that can be used to detect novel outliers. Work done on two algorithms namely Distributed Solving Set and Lazy Distributed Solving Set. In this the data is distributed among no of nodes and then algorithm is applied on each node to find the outliers. The method exploits parallel computation in order to obtain vast time savings. Experimental results show that the algorithms are efficient and that it's running time scales quite well for an increasing number of nodes. So that introduced algorithm is suitable to be used over distributed data sets.

KEYWORDS: outliers; Distance based outliers; parallel and distributed algorithms

I.INTRODUCTION

An Outlier is an observation that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error. In most larger samplings of data, some data points will be further away from the sample mean than what is deemed reasonable. This can be due to incidental systematic error or flaws in the theory that generated an assumed family of probability distributions, or it may be that some observations are far from the center of the data. Outlier points can therefore indicate faulty data, erroneous procedures, or areas where a certain theory might not be valid. However, in large samples, a small number of outliers are to be expected (and not due to any anomalous condition). Hawkins (Hawkins, 1980) defines an outlier as an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.

Now a days outlier detection play important role in data mining because this method is helpful to mine correct data from irrelevant data, using outlier detection finding the noisy data, missing data and many others. Recently, researchers have proposed distance-based, density-based outlier detection methods. Outlier detection is a data mining task. Data mining is the process of extracting valid, previously unknown and actionable information from a large data base. The main goal of outlier detection is to isolate the observations which are dissimilar from the rest of the data. This task has practical applications in several fields such as fraud detection, intrusion detection, data cleaning, medical diagnosis and many others [1]. Data mining includes supervised and unsupervised approaches. Here we are using the unsupervised approach. With unsupervised learning it is possible to learn larger and more complex models than with supervised learning. In unsupervised learning, the learning can proceed hierarchically from the observations into ever more abstract levels of representation. Each additional hierarchy needs to learn only one step and therefore the learning time increases linearly in the number of levels in the model hierarchy.

Identification of potential outliers is important for the following reasons:

- An outlier may indicate bad data. For example, the data may have been coded incorrectly or an experiment may not have been run correctly. If it can be determined that an outlying point is in fact erroneous, then the outlying value should be deleted from the analysis (or corrected if possible).

In some cases, it may not be possible to determine if an outlying point is bad data. Outliers may be due to random variation or may indicate something scientifically interesting. In any event, we typically do not want to simply delete the outlying observation. However, if the data contains significant outliers, we may need to consider the use of robust statistical techniques. We considered a potential attack on collaborative data publishing. We used slicing algorithm for

DOI: 10.15680/ijircce.2015.0305179

anonymization and L diversity and verify it for security and privacy by using binary algorithm of data privacy. For high dimensional data, slicing algorithm is useful. It divides the data in both vertical and horizontal manner. Encryption can increase security, but the limitation is there could be loss of data utility. Our main goal is to publish an anonymized view of integrated data, which will be immune to attacks. We improve the security and privacy with the help of slicing technique which fulfils privacy verification with better performance than provider aware (base algorithm) and encryption algorithm.

II.RELATED WORK

The outlier detection task can be very time consuming and recently there has been an increasing interest in parallel/distributed methods for outlier detection. Parallel versions, called PENL, of the basic NL algorithm have been proposed by Hung and Cheung [3.]. PENL is based on a definition of outlier employed a distance based outlier is a point. Bay's algorithm, which is based on a definition of distance-based outlier coherent with the one used here. Moreover, this parallel version does not deal with the drawbacks of the centralized version, which is sensitive to the order and to the distribution of the data set.

Defining outliers by their distance to neighboring data points has been shown to be an effective non-parametric approach to outlier detection. A fast algorithm for mining distance based outliers exist [2.]. As we process more data points, the algorithm finds more extreme outliers, and the cut-off increases giving us improved pruning efficiency. The state-of-the art distance based outlier detection algorithm uses the NL algorithm with a pre-processed data set. The algorithm facilitates fast convergence to a point's approximate nearest neighbors. RBRP improves upon the scaling behavior of the state-of-the-art by employing an efficient pre-processing step that allows for fast determination of approximate nearest neighbors. We validated its scaling behavior on several real and synthetic data sets. RBRP consistently outperforms ORCA, the state of-the-art distance-based outlier detection algorithm, often by an order of magnitude.

III.PROPOSED SYSTEM

Many prominent data mining algorithms have been designed on the assumption that data are centralized in a single memory hierarchy. Moreover, such algorithms are mostly designed to be executed by a single processor. More than a decade ago, it was recognized that such a design approach was too limited to deal effectively with the issue of continuous increase in the size and complexity of real data sets, and in the prevalence of distributed data sources. Consequently, many research works have proposed parallel data mining (PDM) and distributed data mining (DDM) algorithms as a solution to such issue. This paper introduced a distributed method for detecting distance based outliers in very large data sets. The proposed approach is based on the concept of outlier detection solving set, which is a small subset of the data set that can be also employed for predicting novel outliers. The block diagram of proposed approach is shown in figure 1. The method exploits parallel computation in order to obtain vast time savings. Indeed, beyond preserving the correctness of the result, the proposed schema exhibits excellent performances. From the theoretical point of view, for common settings, the temporal cost of our algorithm is expected to be at least three order of magnitude faster than the classical nested loop like approach to detect outliers.The technique proposed for identifying outliers will be applied initially at distributed clients and their results of detected outliers would be integrated on server machine at final stage computation of outliers. To do this, the outlier detection strategies proposed are: a) DSS Algorithm b) LDSS Algorithm.

Distributed Solving Set algorithm

A distributed method (called Distributed SolvingSet) [1.] to detect distance-based anomalies is presented. Anomaly detection refers to the task of identifying abnormal or inconsistent patterns from a data set. While outliers may seem as undesirable entities in a data set, but identifying them gives us many important information.

The Distributed Solving Set algorithm [1] adopts the same strategy of the Solving Set algorithm.In the existing system, the request from query node is sent to the super node (Global candidate). Super node is having its own data set, where it searches for the data. Then sends the query to other sub nodes (Local nodes) These sub nodes are also having their own data sets, in which they search for the data searched by super node. After completion of the search, all the nodes detect outliers and then remove it from the data set and create one index of the data. Later when data

DOI: 10.15680/ijircce.2015.0305179

computation is done all the nodes communicate to each other and match their data set to remove the duplicate data from the data set. For which the LazyDistributedSolvingSet algorithm [1.] is used. And then they send it back to the super node which then forwards the whole search result to the user.

Lazy Distributed Solving Set Algorithm

A generic approach to Lazy distributed set solving [1.] that permits exploitation of the increasing availability of clustered and/or multiprocessor machines. From the analysis accomplished in the preceding section, it follows that the total amount TD of data transferred linearly increases with the number of employed nodes. Though in some scenarios the linear dependence of the amount of data transferred may have little impact on the execution time and on the speedup of the method and, also, on the communication channel load, this kind of dependence is in general undesirable, since in some other scenarios relative performances could sensibly deteriorate when the number of nodes increases. In order to remove this dependency, we describe in this section a variant of the basic DistributedSolvingSet algorithm [1.] previously introduced. With this aim, an incremental procedure is pursued. Several iterations are accomplished: during each of them only a subset of the nearest neighbors' distances [2.], starting from the smallest ones, is sent by each local node to the supervisor node. At each iteration, the supervisor node collects the additional distances, puts them together with the previously received ones, and checks whether additional distances are needed in order to determine the true weight associated with the candidate objects.

IV.RESULT AND RESULT ANALYSIS

We now present the experimental results for the outlier detection algorithms for DSS algorithm and LDSS algorithm both in terms of effectiveness. We performed the experiments on various real time datasets; those were available on UCI repository, such as cover type data set, credit approval dataset, KDD 99 r2l and KDD 99 u2r. The experimental results are shown with outliers detected in the above datasets.

The following table (Table I) shows the accuracy of outliers found by this system using Distributed Solving Set approach:

Datasets	***Total Instances***	***Actual Outliers***	***Initial Outliers using DSS***	***Final Outliers using DSS***	***Total Percentag e***
Credit approval	690	460	689	356	77.39%
KDD r2l	2196	1370	2195	1093	79.78%
KDD u2r	518	416	318	257	61.77%
KDD probe	941	509	443	319	62.67%
KDD DOS	4422	1770	2430	1200	67.79%

TABLE I: RESULTS USING DSS ON DIFFERENT DATASETS

The following table(Table II) shows the accuracy of outliers found by this system using Lazy Distributed Solving Set approach:

Datasets	*Total Instances*	*Actual Outliers*	*Initial Outliers using DSS*	*Final Outliers using DSS*	*Total Percentage*
Credit approval	690	460	689	376	81.73%
KDD r2l	2196	1370	2188	1154	84.23%
KDD u2r	518	416	498	306	73.55%
KDD probe	941	509	468	374	73.47%
KDD DOS	4422	1770	2769	1460	82.48%

TABLE II : RESULTS USING LDSS ON DIFFERENT DATASETS

V.CONCLUSION AND FUTURE WORK

To summarize a learned lesson, we started from an algorithm founded on a compressed form of data (the solving set) and derived a parallel/distributed data version by computing local distances and merging them at the Coordinator site in an iterative way. The "lazy" version, which sends distances only when needed, showed the most promising performance. This schema could be useful also for the parallelized version of other kinds of algorithms, such as those based on Support Vector Machines. Additional improvements could be to find rules for an early stop of main iterations or to obtain a "one". The proposed solution produces an overall speedup close to linear w.r.t. the number of computing nodes. When the number of nodes increase, the proposed system scales accordingly in the case of computation of coordinator node and data transmission.

REFERENCES

1. Fabrizio Angiulli, Stefano Basta, Stefano Lodi, and Claudio Sartori" Distributed Strategies for Mining Outliers in *Large Data Sets"* IEEE Transactions on Knowledge and Data Engineering VOL. 25,NO. 7,July 2013.
2. F. Angiulli, S. Basta, and C. Pizzuti, "Distance-Based Detection and Prediction of Outliers ''IEEE Trans. Knowledge and Data Eng.,vol. 18, no. 2, pp. 145-160, Feb. 2006.
3. E. Hung and D.W. Cheung, "parallel Mining of Outliers in Large Database,Distributed and Parallel Databases", vol. 12, no. 1, pp. 5-26,2002.
4. E. Knorr and R. Ng, A"Algorithms for Mining Distance-Based Outliers in Larqe Datasets*,"* Proc. 24rd Inti Conf. Very Large Data Bases (VLDB), pp. 392-403, 1998.
5. S.D. Bay and M. Schwabacher, "Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule", Proc. Ninth ACM SIGKDD Inti Conf. Knowledge Discovery and Data Mining (KDD), 2003.
6. M.E. Otey, A. Ghoting, and S. Parthasarathy, "Fast Distributed Outlier Detection in Mixed-Attribute Data Sets'", Data Mining Knowledge Discovery, vol. 12, nos. 2/3, pp. 203-228, 2006.
7. Koufakou and M. Georgiopoulos, " A Fast Outlier Detection Strategy for Distributed High-Dimensional Data Sets with Mixed Attributes*",* Data Mining Knowledge Discovery, vol. 20, pp. 259-289, 2009.