Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Detection of Outliers in Large Dataset using Distributed Approach

Jyoti N Shinde[1],
[1]*Computer Engineering, SRES College of Engineering, Kopargaon,*

Abstract — In this paper, a distributed method is introduced for detecting distance-based outliers in very large data sets. The 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 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 orders of magnitude faster than the classical nested-loop like approach to detect outliers. Experimental results show that the algorithm is efficient and that it's running time scales quite well for an increasing number of nodes. We discuss also a variant of the basic strategy which reduces the amount of data to be transferred in order to improve both the communication cost and the overall runtime. Importantly, the solving set computed in a distributed environment has the same quality as that produced by the corresponding centralized method.

Keywords- Outlier, Distance-based outliers, outlier detection, parallel and distributed algorithms

I. INTRODUCTION

Outliers are those points in a data set that are highly unlikely to occur given a model of the data. Since outliers and anomalies are highly unlikely, they can be can be indicative of bad data or malicious behavior. Examples of bad data include skewed data values resulting from measurement error, or erroneous values resulting from data entry mistakes. An example of data indicating malicious behavior is anomalous transactions in a credit card database, which may be symptoms of someone using a stolen card or engaging in other fraudulent behavior.

Outlier detection is the data mining task whose goal is to isolate the observations which are considerably dissimilar from the remaining data. This task has practical applications in several domains such as fraud detection, intrusion detection, data cleaning, medical diagnosis, and many others. Unsupervised approaches to outlier detection are able to discriminate each datum as normal or exceptional when no training examples are available. Among the unsupervised approaches, distance-based methods distinguish an object as outlier on the basis of the distances to its nearest neighbors. These approaches differ in the way the distance measure is defined, but in general, given a data set of objects, an object can be associated with a weight or score, which is, intuitively, a function of its k nearest neighbors' distances quantifying the dissimilarity of the object from its neighbors. In this work, we follow the definition given in a top-n distance based outlier in a data set is an object having weight not smaller than the nth largest weight, where the weight of a data set object is computed as the sum of the distances from the object to its k nearest neighbors.

II. RELATED WORK

The 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 [1].

The outlier detection task can be very time consuming and recently there has been an increasing interest in parallel/ distributed methods for outlier detection. Defining outliers by their distance to neighbouring data points has been shown to be an effective non-parametric approach to outlier detection. In recent years, many research efforts have looked at developing fast distance-based outlier detection algorithms. Several of these efforts report log-linear time performance as a function of the number of data points on many real life low dimensional datasets.

Hung and Cheung [3] presented a parallel version, called PENL, of the basic NL algorithm [4]. PENL is based on a definition of outlier employed in [4]: a distance-based outlier is a point for which less than k points lie within the distance in the input data set. This definition does not provide a ranking of outliers and needs to determine an appropriate value of the parameter. Moreover, PENL is not suitable for distributed mining, because it requires that the whole data set is transferred among all the network nodes. Lozano and Acuna proposed a parallel version of Bays algorithm [8], which is based on a definition of distance-based outlier coherent with the one used here. However, the method did not scale well in two out of the four experiments presented. Moreover, this parallel version do es not deal with the drawbacks of the centralized version in [8], which is sensitive to the order and to the distribution of the data set.

Otey et al. in [6] and Koufakou and Georgiopoulos in [7] proposed their strategies for distributed high-dimensional data sets. These methods are based on definitions of outlier which are completely different from the definition employed here, in that they are based on the concept of support, rather than on the use of distances.

Dutta et al. [10] proposed algorithms for the distributed computation of principal components and top-k outlier detection. In their approach, outliers are objects that deviate from the correlation structure of the data: A top-k outlier is an object having at most the k^{th} largest sum of squared values in a fixed number of the lowest order principal components, where each component is normalized to its deviation. This definition neither implies nor is implied by the definition employed in this work. For example, if all clusters are located far from the mean of the data set, distance-based outliers close to the mean are not necessarily exceptional in the correlation structure. On the other hand, objects having large values in the first principal components need not have smaller weight than objects which deviate from the correlation structure in the low-order components.

III. SYSTEM OVERVIEW

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.

IV. SYSTEM DESIGN

4.1.Data Collection and Data Pre-processing

In data collection the initial input data for this system will be collected from standard dataset portal i.e. UCI data set repository. As proposed in system, the standard dataset will be used for this system includes Covertype, IPS datasets, etc. Collected datasets may be available in their original, uncompressed form therefore, it is required to preprocess such data before forwarding for future steps.

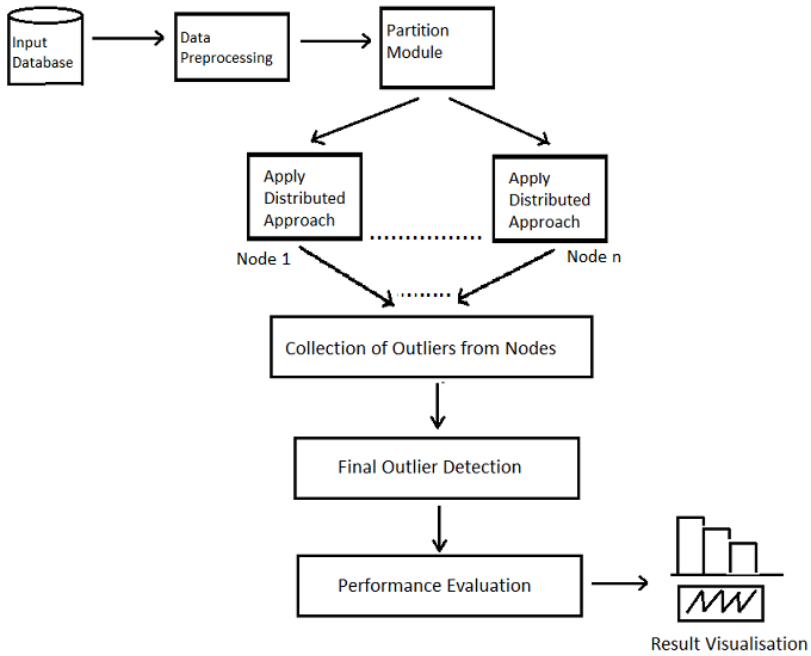

Fig 1. Block diagram of proposed system

To preprocess large dataset contents, techniques available is data mining such as data integration, data transformation, data cleaning, etc. will be used and cleaned, required data will be generated.

4.2.Data partitioning

In this module, the preprocessed data is divided in to number of sub datasets form central dataset. As per the data request made by desired number of nodes. This partitioned data will be then processed by individual sub datasets to identify outliers based on applied algorithm strategy.

4.3.Outlier detection

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.

4.3.1. Distributed Solving Set Algorithm - The Distributed Solving Set algorithm [1] adopts the same strategy of the Solving Set algorithm. It consists of a main cycle executed by a supervisor node, which iteratixvely schedules the following two tasks:

i The core computation, which is simultaneously carried out by all the other nodes;
ii The synchronization of the partial results.

These results are returned by each node after completing its job. The computation is driven by the estimate of the outlier weight of each data point and of a global lower bound for the weight, below which points are guaranteed to be nonoutliers. The above estimates are iteratively refined by considering alternatively local and global information.

The core computation executed at each node consists in the following steps:

- Receiving the current solving set objects together with the current lower bound for the weight of the top n-th outlier,
- Comparing them with the local objects,
- Extracting a new set of local candidate objects (the objects with the top weights, according to the current estimate) together with the list of local nearest neighbors with respect to the solving set and, finally,
- Determining the number of local active objects, that is the objects having weight not smaller than the current lower bound.

The comparison is performed in several distinct cycles, in order to avoid redundant computations. These data are used in the synchronization step, from the supervisor node, to generate a new set of global candidates to be used in the following iteration, and for each of them the true list of distances from the nearest neighbors, to compute the new (increased) lower bound for the weight.

4.3.2. Lazy Distributed Solving Set Algorithm - From the analysis accomplished in the preceding section it follows that the total amount TD of data transferred linearly increases with the number *l* of employed nodes. Though in some scenarios the linear dependence on 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, a variant of the basic DistributedSolvingSet algorithm [1] previously introduced is described here in this section

The variant, named LazyDistributedSolvingSet algorithm[l], which is shown in figure 3.4 employs a more sophisticated strategy that leads to the transmission of a reduced number of distances for each node, say kd, therefore replacing the term lk in the expression TD of the data transferred with the smaller one lkd, such that lkd is O(k). This strategy, thus, mitigates the dependency on 1 of the amount of data transferred, so that the relative amount of data transferred can be approximated to,

$$TD\ \% \approx Qk/a.$$

Moreover, the first term in below equation, representing the temporal cost pertaining to the supervisor node, is replaced by LazyDistributedSolvingSet as follows,

$$0(\ Q\ |D|\ k(\ k\ 1og\ k + 1og\ n)),$$

Thus relieving the temporal cost from the direct dependency on the parameter *l*. LazyDistributedSolvingSetAlgorithm[l] differs from the preceding one for the policy adopted to collect the k nearest neighbours distances of each candidate object q computed by each node. With this aim, an incremental procedure is pursued. Several iterations are accomplished: during each of them only a subset of the nearest neighbours distances, 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 associated with the candidate objects. If it is the case, a further iteration is performed, differently the incremental procedure steps.

V. CONCLUSION

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 a 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”.

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] A. 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.
[8] E. Lozano and E. Acuna, Parallel Algorithms for Distance-Based and Density-Based Outliers, Proc. Fifth IEEE Inti Conf. Data Mining (ICDM), pp. 729-732, 2005
[9] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-Based Detection and Pre- diction of Outliers," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 2, pp. 145-160, Feb. 2006.
[10] H. Dutta, C. Giannella, K.D. Borne, and H. Kargupta, Distributed Top**ic** Outlier Detection from Astronomy Catalogs Using the DEM AC System, Proc. SIAM Inti Conf. Data Mining (SDM), 2007.

