ISSN 2278-3091
International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 526- 528 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Detecting Outliers in Large Dataset using Distributed Method

D. Madhavi[1] **Ch. Sreenubabu**[2]
[1] PG –M.Tech, Dept of CSE, India, madhulika511@gmail.com
[2] Associate Professor, Dept of CSE, India, sreenubabu.ch@gmail.com

ABSTRACT
In this paper we introduce a distributed method for detecting distance-based outliers in very large data sets. Work done on two algorithms namely supervisor, NodeInit procedure algorithm. Supervisor algorithm distributes the data sets to sub nodes and NodeInitprocedure algorithm executes at each node to finding outliers. The algorithm exploits parallel computation in order to meet two basic needs: (i) reduction of the run time with respect to the centralized version and (ii) ability to deal with distributed data sets. 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.

Key words: Outlier detection, Distance-based outlier, Distributed method.

INTRODUCTION
An outlier is an observation that differs so much from other observations. 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. In Unsupervised approaches for outlier detection are able to discriminate each data set as normal or exceptional when no training examples are available. Among the unsupervised approaches, distance-based outlier detection methods disti -nguish an object as outlier on the basis of the distances to its nearest neighbors [8,7,9].These approaches differ in the way the distance measure is define, but in general given objects can be associated with weights. There exist several approaches to the identification of outliers named as model-based outlier detection where data in parametric distribution [1]. This is not suitable for high dimensional data so overcome those problems we introduce new methods are distance–based, index-based, nested-loop and dolphin's method. Working of these methods are index-based method is take index values[5], nested loop is based on buffer values, dolphins method is based on disk-resident data sets[6]. The outlier detection task can be very time consuming and recently there has been an increasing interest in paral -lel/distributed implementations of outlier detection method. These methods are not suitable for large data sets. Hung and Cheung [8] presented parallel version, called PENL. It is not suitable for mining distributed data sets, because it requires all data set is transferred among all the network nodes. To overcome those problems we present distributed method for large dataset [3] values. The work is based on PDM/DDM approach to the computation of distance-based outliers. Approach of this exploit the locality properties of the problem at hand to partition the computation among the processors of a multiprocessor system or the host nodes of a communication network to obtain vast time savings.

DEFINITIONS AND TASK

In the following, we assume that a limited subset of the data set in a given metric space

Definition (outlier weight)
We can estimate the weight of each dataset object and compare with the neighbor objects, if the weight of the object is lower than the greatest weight of the candidate objects is called non-outlier weights.

Definition (Distance-based outlier)
Given parameters k and R, an object is a distance-based outlier if less than k objects in the input data set lie within distance R from it.

ALGORITHMS

In this paper we introduce two algorithms named supervisor algorithm and nodeInit procedure.

Supervisor algorithm
In here we present a distributed approach to detect dist-ance- based outliers. Previously so many researches work on distance-based outliers but not for large set of data [5]. Here both parallel/distributed techniques are used for detecting distance-based outliers. Supervisor algorithm is used for distribute the data. it take data from global data set and consisting a main cycle executed by a supervisor node, which iteratively schedules the following two tasks
1) Core computation here data is simultaneously carried out data to all other nodes.
2) After completion of the job, the partial results returned by each node to supervisor [7].
The core computation executed at each node consists in the following.

- Receiving the current solving set objects
- Generate the distances of each object to another object
- Calculate weights of each record

Finally gives the correctness of results. It reduces data transformations and communication cost

```
Algorithm supervisor
Begin
Dsize=selected data from global data;
Temp N=0;
int v=data size;
n=no. of nodes;
for(int i=0;i<n;i++)
{
f3.show();
}
Distribute load to n nodes;
If(f is node)
{
V=temp N;
(f as node)receivenewtaskfromsupervisor(v);
TempN+=100;
}
```

Fig.1 Supervisor algorithm

The supervisor algorithm show in fig.1 it runs for distribute data to sub nodes and data pass simultaneously to other nodes and synchronization of the partial results returned by each node after completing its job.
The comparison is performed in several distinct cycles, in order to avoid redundant computations. The above data are used in the synchronization step by the supervisor node to generate a new set of global data 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 lower bound for the weight.

NodeInitprocedure algorithm

The communication among the supervisor node and the local nodes is carried out by the NodeInitprocedure and runs at each node. It takes data from the supervisor node and work done as follows

Finding the distance one to each other and also know weights of each data set
Distance(r, di) = $\sqrt{(xi - xj)^2 + (yi - yj)^2}$
Wait (r,w) =d1+d2+d3+……………dn
All weights of data sets are arranged in order of least to greatest [5].
Calculate the median of the data set. The median of a data set is the data point above which half of the data sets and below which half of the data sets - essentially, it's the "middle" point in a data set. If the data set contains an odd number of points, this is easy to find - the median is the point which has the same number of points above as below it. However, if there is an even number of points, then, since there is no single middle point, the 2 middle points should be averaged to find the median.
Find the lower bounder. This point, to which we will assign the data point below which 25 percent (or one quarter) of the observations sit. In other words, this is the halfway point of the points in your data set below the median. If there are an even number of values below the median, you once again must average the two middle values for lower bound values otherwise directly take lower bound value

- Finding the upper bounder. Which is assigned the data point above 25 percent of the data sits.

The work is done like above based upon the lower and upper boundary values we find the outlier data.

RESULTS

To check effectiveness of the proposed algorithms we eva -luated performance of the algorithms through several experiments on large data sets. Results of that are shown in the form of graphically.
For the sake of brevity, we abbreviate names of the algorithms supervisor and Nodeinitprocedure
respectively.

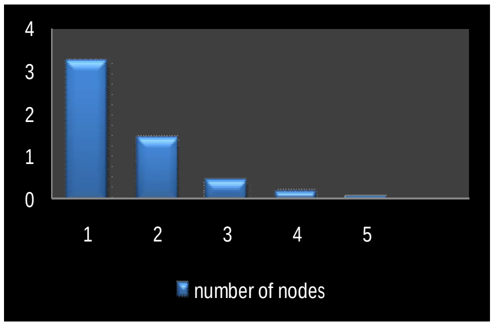

Fig.2 Speed up

Speedup: Fig.2 shows the speedup obtained by using the supervisor and nodeinitprocedure algorithm. We note that, for all the considered data sets, the algorithm Experimental results show that the algorithms is efficient and that it's running time scales quite well for an increasing number of nodes. These good performances can be explained by analyzing the communication time and the supervisor node processing time. Communication time is concerned as the time spent to transfer data from the local nodes to the supervisor node and whole execution time. The algorithms are written in .net frame work. As experimental platform we used I3 processor and 2GB of RAM.

In this section the values of parameters are number of nodes 'n' is gradually increase at fixed number of data size N=1000, in poker hand data. The time is taken at every number of nodes size what we taken and exhibits performance. we examine the empirical performance of the simple algorithm on several large real data sets. In here increasing the number of nodes time is gradually reduced means effiency is increased. We also examined how the total running time scales.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 526- 528 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Table 1: Total execution time Total

Data set size=1000	
No. of nodes	Time(in minutes)
1	3:25
2	1:45
3	0:45
4	0:21
5	0:09

Table 1 The below table report the total execution time at (in minutes) different number of local nodes at size of fixed. It shows that supervisor algorithm guarantees vast time savings with respect to the centralized algorithm nodeInitprocedure. Shows the nodeinitprocedure is computation of the distance between two records in data set. It can be noted the total number of distances computed is slightly increasing with the number of nodes. Total number of distances computed is little sensitive to the number of nodes. All distances are computed and calculate weights associated to distances

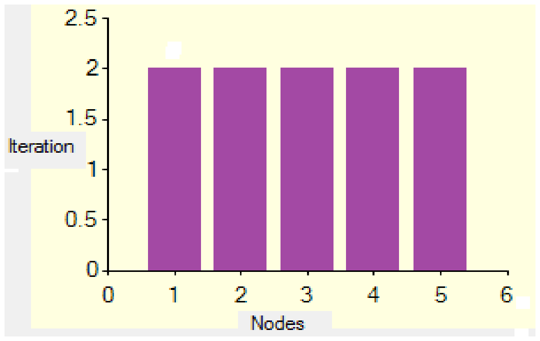

Fig.3 Load balance

Load balance: fig shows the load balance of each node. Here number of nodes n=5, Data set size N=1000, each node contain the number of iterations at each node. The above figure represents equal load balance of each node at specific data size. So the total time to mine outliers on the data sets perform well on this data.

Table 2: Load balance

Data se t size=1000	
No. of nodes	No. of itreratations
1	2
2	2
3	2
4	2
5	2

Table 2 shows load balance of each node at run time. All nodes had same load balance for increasing the efficiency and Accuracy of used algorithms.

Conclusion and Future Work

We presented supervisor algorithm, a distributed method for computing an outlier detection and top-n distance-based outliers according to definitions in [7], [5]. By computing local distances and merging them at a coordinator site in an iteratively by the help of supervisor, Nodeinitprocedure algorithms. This are used for work in distributed environment and gives Experimental results show that the algorithms is efficient and that it's running time scales quite well for an increasing number of nodes. 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-shot" merging method of the local information with some approximation guarantees.

REFERENCES

[1]V. Barnett and T. Lewis. Outliers in Statistical Data. John Wiley and Sons, 1994.

[2]Algorithms for Mining Distance-Based Outliers in Large Datasets .Edwin M. Knox and Raymond T. Ng Department

[3]F. Angiulli, S. Basta, S. Lodi, and C. Sartori, "A Distributed Approach to Detect Outliers in very Large Data Sets," Proc. 16th Int'l Euro-Par Conf. Parallel Processing (Euro-Par), pp. 329-340, 2010

[4]" Distributed Strategies for Mining Outliers in Large Data Sets".Fabrizio Angiulli,

[5] Algorithms for Mining Distance-Based Outliers in Large Datasets ,"Edwin M. Knox and Raymond T. Ng"

[6] DOLPHIN: An Efficient Algorithm for Mining Distance-Based Outliers in Very Large DatasetsFABRIZIO ANGIULLI and FABIO FASSETTI

[7] Knorr, E., Ng, R.: Algorithms for mining distance-based outliers in large datasets. In: Proc. Int. Conf. on Very Large Databases (VLDB 1998), pp. 392–403 (1998)

[8] Ramaswamy, S., Rastogi, R., Shim, K.: E□cient algorithms for mining outliers from large data sets. In: Proc. Int. Conf. on Managment of Data (SIGMOD 2000), pp. 427–438 (2000)

[9]Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: Proc. KDD (2003)