12-2005

Improving the performance of nested loop algorithm using separators

Nachiappan N. Nachiappan
University of Texas-Pan American

IMPROVING THE PERFORMANCE OF NESTED LOOP ALGORITHM USING SEPARATORS

A Thesis

by

Nachiappan N Nachiappan

Submitted to the Graduate School of the
University of Texas – Pan American
In Partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2005

Major Subject: Computer Science

IMPROVING THE PERFORMANCE OF NESTED LOOP ALGORITHM USING SEPARATORS

A Thesis
by
Nachiappan N Nachiappan

Approved as to style and content by:

Zhixiang Chen, Ph. D.
Chair of Committee

Richard H. Fowler, Ph. D.
Committee Member

John Abraham, E.d.,
Committee Member

Ping Sing Tsai, Ph.D.,
Committee Member

December 2005

ABSTRACT

Nachiappan, N. Nachiappan . Improving the performance of Nested Loop Algorithm Using Separators, Master of Science (MS), December 46 pp., 15 Figures, references, 25 titles.

This thesis studies the properties of distance-based outliers and a better detection method for large multi-dimensional datasets. Outlier detection is an important task to find out the objects that deviate in a high ratio from the rest of the objects. The proposed algorithm breaks the data set into divisions and sets the area of access for each division, thus reducing the unnecessary access for a major set of elements. This algorithm reduces the run time of the existing algorithm by using separators. Datasets of varying sizes have been tested to analyze the empirical values of these procedures. Effective data structures have been implemented to gain efficiency in memory-performance.

TABLE OF CONTENTS

LIST OF TABLES

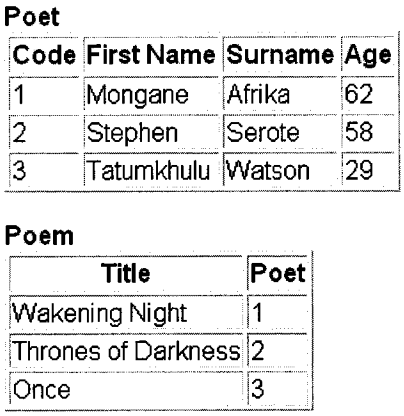
LIST OF FIGURES

CHAPTER I

INTRODUCTION

The entire world stores a lot of data, which are used for various purposes and database is a better way to store these data. The stored data size is growing day by day in giga bytes and tera bytes. Processing these data would be the biggest task than storing them. This makes the field of Data Mining into existence. Data mining is defined as an information extraction activity whose goal is to discover hidden facts contained in databases, using a combination of machine learning, statistical analysis, modeling techniques and database technology. This procedure gets its name because of the similarity to the procedure of mining rocks for a vein of valuable core. The other synonyms that indicate data mining process are knowledge extraction and pattern discovery. Data mining finds patterns and the relationships in data and infers rules that allow the prediction of future results. Data Mining is technically defined as the non-trivial extraction of implicit, previously unknown, and potentially useful information from data [1]. The data source file could be of any size. They could be in the form of flat files, relational databases, data warehouses, transaction databases, multimedia databases, spatial databases, time-series databases etc [24]. The figures shown below are examples of a relational database of a book store which maintains all the poets' information and poem information. Flat files are the frequently used data sources for data mining.

Relational databases constitute of values or attributes arranged in the form of tables. Transaction databases consist of a collection of records, which represent transactions. Multimedia databases store videos, images, texts, etc. Databases which store geographical information such as maps are called spatial databases.



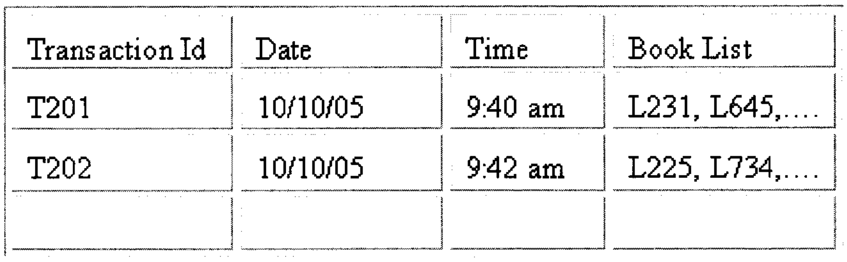
Poet

Code	First Name	Surname	Age
1	Mongane	Afrika	62
2	Stephen	Serote	58
3	Tatumkhulu	Watson	29

Poem

Title	Poet
Wakening Night	1
Thrones of Darkness	2
Once	3

Figure 1 Sample relational database of a book store



Transaction Id	Date	Time	Book List
T201	10/10/05	9:40 am	L231, L645,....
T202	10/10/05	9:42 am	L225, L734,....

Figure 2 Sample transaction database of a book store.

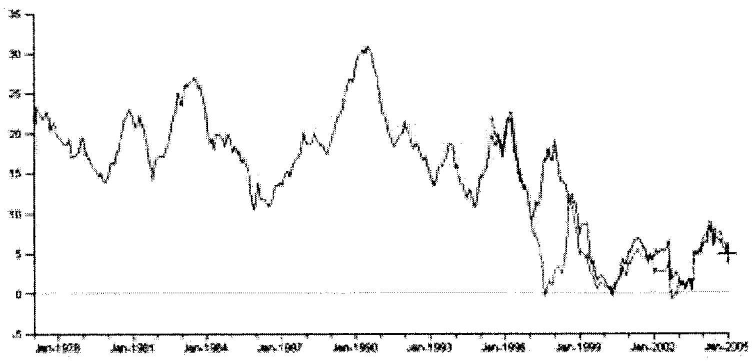


Figure 3 Sample time-series database [24].

Data mining techniques could be used in various areas such as marketing, stock market analysis, foreign exchange, computational bio-sequence analysis etc. The insurance companies use data mining techniques to detect the fraud patterns and predict high-risk situations. Data mining techniques have been useful in analyzing large volumes of data and gain their knowledge. This knowledge gained helps the user to improve the understanding of data and evaluate the validity of the identified outliers [19].

One important data mining application which is focused a lot is the market - basket problem [2, 3, 4]. It deals with the associations between the items of the customer's transactions. The association rules help to decide which items to be put on sale based on the customers. These association rules are used in many other applications such as stock market analysis, library usage and so on [5, 6].

Apart from the techniques applied in the above mentioned applications, one important data mining application which is widely used nowadays is the outlier detection.

The applications such as credit card fraud, may have some patterns existing which differ much from the other patterns. Such patterns need consideration which may reduce the accuracy of the results to a large extent. Those kinds of data are usually called the outliers.

Outliers exist widely in various real-world applications. Earlier studies on outlier-detection were developed in the field of statistics. Outliers are the rare or atypical data objects that do not comply with the general behavior or model of the data. They are many definitions of outliers proposed by different researchers. Barnett and Lewis stated an outlier to be "an observation which is inconsistent with the remainder of that set of data" [7]. Hawkins defined it as "The intuitive definition of an outlier would be "an observation which deviates so much from the other observations so as to arouse suspicion that it was generated by a different method" [8]. Outliers could be of different types. Univariate outliers are those which scale with a single dimension while multivariate outliers scale with multi dimensions. These multivariate outliers are further classified into gross and structural outliers. Gross outliers are the observations that differ from the other members for one or more attributes while structural outliers depend on the structure of the non-outlying data. Structural outliers are usually detected only when all k dimensions are considered.

Outliers are the observations which may be of vital importance for researchers. In a distribution of a dataset, the extreme values are usually called outliers. In manufacturing process, a 3σ rule is adopted. Here the standard deviationσ, and mean μ are used as methods of calculation. According to this 3σ rule, if an observation exists outside the range of μ-3σ and μ+3σ, then such an observation is determined to be an outlier. But these methods could only detect single outliers. When coming to the question of multiple outliers, they would fail. Then the median strategy was introduced which could detect outliers. Although outliers are unexpected values, they should not be removed. Instead a detailed study of these outliers could lead us to an improved understanding of literature and phenomena under study.

PREVIOUS RESEARCH

Existing work on outlier detection is based on the field of statistics. There were different methods introduced to detect outliers. The outlier detection depends on various conditions such as the distribution parameters, the number of expected outliers, the data distribution and their types. But all these conditions could only generate single-dimension or univariate outliers. Moreover the data distribution is a non-deterministic factor. Extensive testing of datasets is required to determine if an attribute fits a particular distribution. These are the disadvantages faced which gave rise to depth-based outlier detection methods. In this method, the dataset is organized into various layers with the assumption that shallow layers have high probability of having outliers when compared to deep layers. But even these depth-based methods proved to be disadvantageous for higher dimensions.

Clustering is basically grouping datasets based on similarities. Here the notion of outliers is defined indirectly which would optimize clustering more than outlier detection. For these, outliers are objects that do not belong to any cluster. The next category of outlier detection is the distance-based methods. This thesis discusses about the distance-based outliers detection methods. Once the outliers are detected, the intensional knowledge of these outliers is studied to gain knowledge about the data in the dataset. This intensional knowledge helps in evaluating the validity of the identified outliers and increases the understanding of the data. Distance-based outlier detection method can scale for any large value of k, where k is the number of dimensions unlike the depth-based methods.

Though outliers have originated in the field of statistics, researchers are seeing to it as an important tool in various areas such as fraud-detection analysis, identifying intrusions etc. In the previous times, the best treatment to outliers was to delete them because the assumption was that outliers could have been the result of erroneous data. But this would result in loss of possible information. Later, researchers came to a conclusion that detailed study of these outliers could reveal some important information of the dataset and thus possibly help in determining any further inconsistencies. Outliers could be univariate or multivariate.

The various existing outlier detection methods are visual based, distribution-based, depth-based and distance-based methods. Visual-based methods are useful only with low dimensions. They use box plot (1-D), scatter plot (2-D), spin plot (3-D) [20]. These methods are time-consuming. Distribution based methods require to know the kind of distribution of data. Depth based methods have a high complexity and are unsuitable for higher dimensions. Distance-based methods are quadratic with respect to the number of dimensions. They are suitable for higher dimensions.

There are many algorithms for detecting outliers, but most of them are not fast enough when the underlying probability distribution is unknown, the size of the dataset is large, and the number of dimensions in the space is high. There are, however applications that need tools for fast detection of outliers in exactly such situations. This thesis discusses a better solution for the above problem.

DEFINITION OF TERMS

Data Mining: It is defined as the non-trivial extraction of previously unknown and useful information from data.

Outlier: An observation or data which deviates from the other members of the dataset or the top n examples for which there are fewer than p other examples within distance d. [12]

Univariate outliers: The outliers which are detected based on a single attribute.

Multivariate outliers: The outliers which are detected based on many attributes.

Distance-based outliers: An object O is a DB (p, D) outlier if at least a p fraction of objects are > distance D from O [11].

Clustering: Grouping data based on the similarity of attributes.

Mean: The average of all the values of a dataset.

Trivial outliers: The outliers which are found in any subspace of an attribute space.

Non-trivial outliers: The outliers which are found in the strongest and non-strongest subspaces alone such as strongest and weak outliers [19].

CHAPTER II

REVIEW OF LITERATURE

A short review of literature is discussed in this section which focuses on the history of data mining, the applications of data mining, the relationship between outliers and data mining, outlier detection methods, various algorithms proposed and studied distance-based outliers and various methods of detection along with their time-complexities. This thesis proposes and calculates the efficiency of the proposed algorithm for the distance-based outliers for k-dimensions with varying datasets. Researchers considered various factors in developing their algorithms for detecting these outliers. Some felt that distribution of data could be an important factor where the data lying out of their assumed boundary conditions were identified as outliers. But then the time-consumption and the amount of overhead used in determining the distribution of the dataset lead to the discovery of other ways.

One such way was the depth factor [9, 10]. In this method researchers felt that by arranging the dataset into organized layers, the data lying in the deeper layers would form more of a cluster than the data lying in the shallow layers. This would rule out the outliers.

But this approach also had some flaws in it. Researchers felt that this kind of approach would be impractical for a dimensionality $k > 4$ for large datasets. [11]. Then dimensions and was feasible with large datasets. Various algorithms have been proposed in detecting outliers based on distance. Algorithms such as nested loop proved to give a time-complexity better and more attractive than the depth based approaches for k-D data sets with k >3. Cell based algorithm with a complexity linear on n, and guarantee of not more than 3 passes over the data. However Cell based algorithm is exponential on k and is recommended only for smaller values of k. Thus, we need an algorithm which could handle large data sets, in which the dimensionality of the dataset exceeds 20.

HISTORY OF DATA MINING

Analyzing and processing data in order to group them into categories is called data mining. Data mining is basically a tool which helps in analyzing data based on different factors. Data could be basically information or knowledge. It could be operational such as cost, non-operational such as industry sales or could also be meta data such as a database. The study of this data would reveal some information. For example a detailed study of the sales transaction of a shop could reveal information about which product is best selling. This information is studied by researchers to form patterns and association between similar data [12].

This information is basically converted to knowledge. The concept of data warehouse is integration of various databases from which data can be extracted for developing knowledge from the extracted data. The information stored in the databases is increasing dramatically as everything is stored in the databases. The tools and the facilities to understand and analyze this data have not increased at the same rate. Analyzing these data can be a valuable asset to any process. To efficiently analyze these issues computer science community has created a new field called Knowledge discovery in databases. Knowledge discovery uses methods and techniques that are derived from the areas of statistical and data analysis, decision support and machine learning.

The relationship among data can be categorized into four groups. One of them is called classes. Here the data is divided and stored in groups so that data can be searched from these groups. For example, the details of a customer's visit and their orders from a restaurant. The classes could develop knowledge in developing the customers menu based on their interest in this example. The next category is the clusters. Clusters are formed based on the logical relationship between the data.

There are various levels of analysis of data mining. The non-linear models resembles neural networks can be used for data analysis. Genetic algorithms can analyze relevant data based on natural evolution. Decision-trees can also be used which generate rules for classifying the dataset. Using the nearest neighbor method is also valid which classifies the records in the dataset. Visual interpretation of relationships in a dataset can also be used to mine data [12].

Various organizations use data mining techniques to facilitate various areas of development such as marketing, finance, stock analysis etc. Other applications include predicting foreign exchange rates, finding genes in DNA sequences [13]. One important application that is discussed frequently in the market is the market basket problem [2, 3, and 4]. It helps the retail stores to build a relationship between items which are non-deterministic. Another application which is overlooked is the outlier detection ability. The concept of outliers is used in fraud detection applications such as identity theft in credit-card, telephone billing etc. The next section discusses about these outliers and their relationship with data mining.

OUTLIERS

Outliers are the rare or atypical data objects that do not comply with the general behavior or model of the data. It is an observation which is exceptional. Applications such as fraud detection, customized marketing, network intrusion detection, weather prediction, and exploration in science databases require the detection of outliers. Researchers have come out with various definitions of these outliers. The various definitions of outliers are stated below. D.Hawkins defines an outlier as an observation that is so much different from the other members such that it gives rise to suspicion about its method of creation [8]. An outlier is an observation which appears to be varying with the other members of that dataset [7].

The figure's below give a graphical representation of outliers which could give us a better idea about how to identify an outlier from a given dataset. Data in the dataset is distributed in a 2 dimensional graph in which the outlying objects away are considered to be outliers. In figure 2 the data distribution is mainly at the centre and hence the data which are lying away from it are considered to be outliers. In figure 3 data is concentrated at the bottom of the graph. Thus the data which are away from this denser area are considered to be outliers.

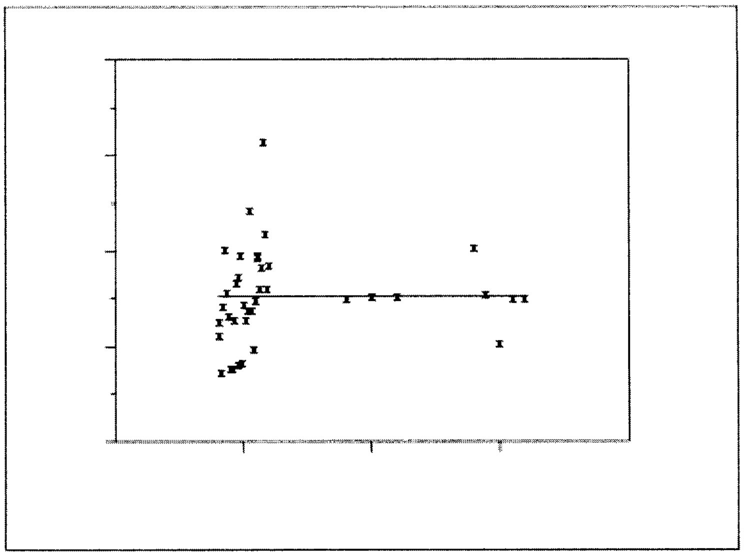


Figure 4 graph representing outliers with data concentrated at the centre of the graph [20].

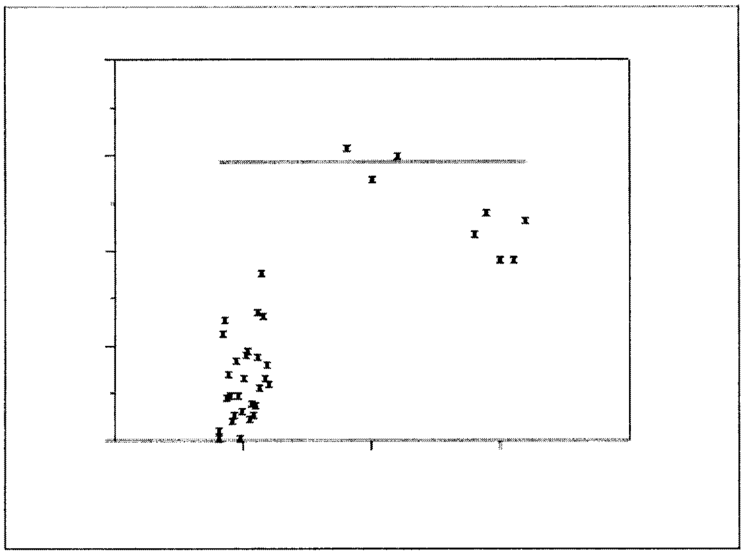


Figure 5 Graph representing outliers with data concentrated at the bottom of the graph [20].

Outliers can be classified into different types and have various properties associated with them. According to the authors of statistics literature, outliers can be classified into gross and structural outliers [16]. Gross outliers are identified with respect to one or more individual attributes while structural outliers might sometimes be invisible in 2D or 3D representation but may appear when all dimensions are put together. When data are organized linearly, the end values would be considered as outliers. But when they are distributed randomly, the outliers could arise at any point. Hence outliers originate depending on the type of distribution. Draper and Smith stated that outliers may arise due to some unusual combination of attributes which might need investigation for future use [17]. For example the astronomical data related to the orbital position of the planet mercury had some outlying observations which were unidentified till the discovery of the theory of relativity [18]. As said in signal processing, "One person's noise is another person's signal." For many applications, such as the exploration of satellite or medical images, and the monitoring of criminal activities in electronic commerce, identifying exceptions can often lead to the discovery of truly unexpected knowledge [19].

Outlier detection algorithms act as an important data mining tool for researchers and scientists who deal with large amounts of data and attributes of data. Scientists could effectively deal with smaller datasets and fewer numbers of attributes. Sometimes the dataset is so large that it cannot fit the memory. Data mining helps in mining data efficiently by using its mining tools for such large databases.

OUTLIER DETECTION METHODS

Outlier detection was mainly used in the field of statistics. There is no universally defined circumstance for the origin of an outlier. Hence various outlier tests have been developed. They were based on data distribution, distribution parameters, the number of expected outliers and their types [7]. Since outliers could be univariate or multi-variate, there were several methods proposed by researchers to detect the outliers. These methods can be broadly classified into graphical methods, distribution based, depth based and distance based outliers. Distance based outliers is the area of focus in this paper.

Box-plots were the earlier used methods for detecting univariate or single-dimensional data. Box plots were drawn by Vertical axis as response variable and horizontal axis as factor of interest. This method is based upon quartiles and median. The 2D space is divided into upper and lower quartile. A box is drawn from the median to both the quartiles. Then a line is drawn between the lower quartile and the minimum point and the upper quartile and the, maximum point. Figure 4 represents the box plot where boxes are drawn from the median value to both quartiles and are connected accordingly. Box plots accurately detect the presence of outliers. They are usually used for quick data comparisons. A single box plot can be drawn for one batch of data with no distinct groups. Alternatively, multiple box plots can be drawn together to compare multiple data sets or to compare groups in a single data set.

For a dataset in 2D space, a scatter plot can be use to detect outliers and a 3D spin plot can be used to detect outliers in a 3D space. These graphical methods can be used only up to three dimensions [19]. The scatter plot reveals a basic relationship between X and Y axis for most of the data and a single outlier. Figure 5 shows the graphical representation of a scatter plot. The X-axis represents one attribute while the Y-axis represents another attribute. Different types of relationships between the attributes can be shown. For example, Figure 6 shows a linear relationship between data, Figure 7 shows no relationship among the two attributes of a dataset and Figure 8 shows a quadratic relationship between the attributes of the dataset [20].

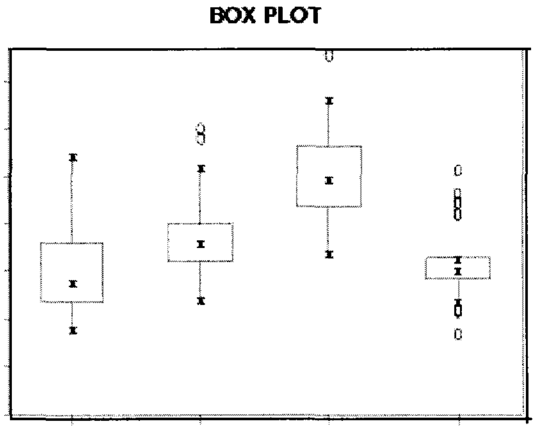


Figure 6. Representation of a box plot [20].

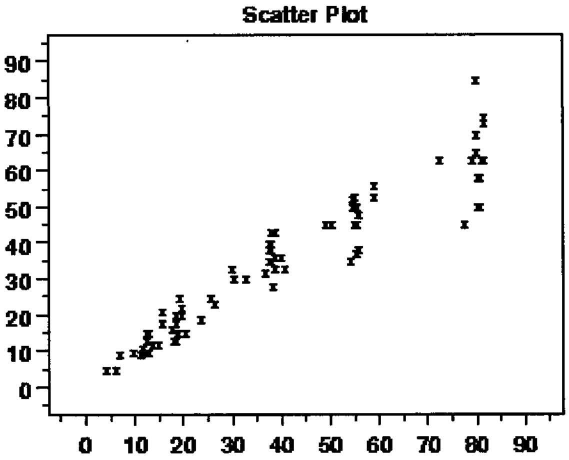


Figure 7. Representation of a scatter plot [20].

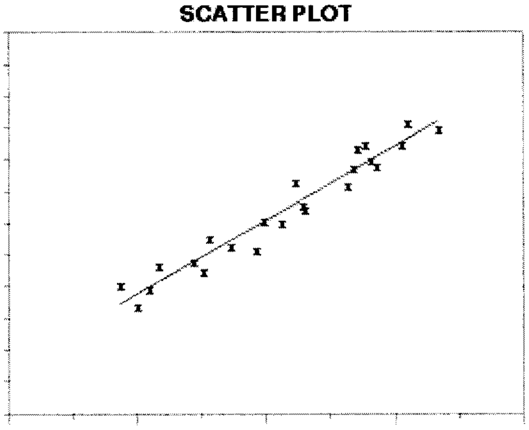


Figure 8 . Scatter plot showing linear relationship between two attributes [20].

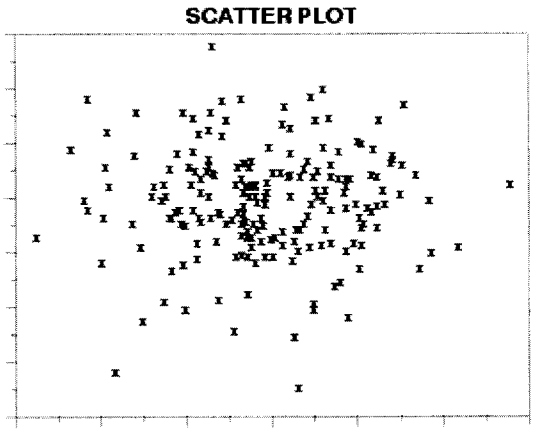


Figure 9. Scatter plot showing no relationship between two attributes [20].

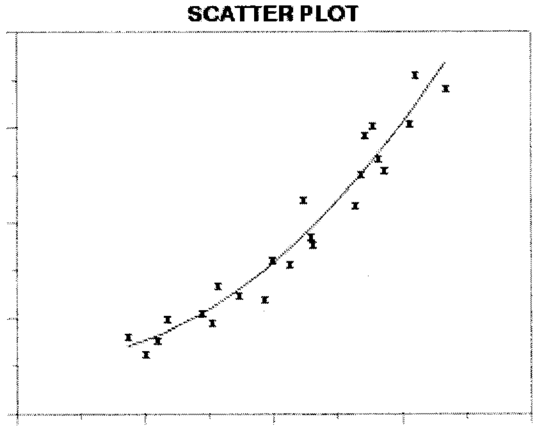


Figure 10. Scatter plot showing a quadratic relationship between two variables [20].

Graphical methods could scale only up to a maximum of 3D dataset. Detecting outliers from datasets of any dimensions is the main interest of research for researchers. They found out that outliers are nothing but those data which do not fit a particular distribution model followed by the other members of the dataset. This is how the idea of distribution methods for outlier detection came into existence. Some researchers stated that outliers could be detected by plotting the residuals of a dataset and examining those residuals. The residual elements lying 3 or more deviations away from the mean value could be stated as outliers, approximately [17]. But some did not agree with this argument because they felt that this may not be reliable due to the assumption of a specific model, the correlation among the residuals and the effect of even a single outlier on the residual values [21].

Many statistical tests were developed to detect outliers based on the distribution parameters, data distribution, the number of expected outliers and their types [7]. They developed numerous number of tests for normal, gamma exponential distributions. But these tests were found to be unsuitable for multidimensional, large datasets, when distribution of the elements is unknown; the number of outliers is unknown and when outliers are not the extreme values [19]. The outliers could be anywhere in the distribution in the distribution based on other elements or groups of element in the set.

Depth based methods were found to be useful in detecting outliers. In these methods the data space is organized into layers with the assumption that shallow layers are considered to contain outliers when compared to the deeper layers. The concepts of peeling and depth contours are used to implement depth based outlier detection. Peeling is stated as removal of the extreme values of the dataset and for the higher dimensions the outlying points on a convex hull [9, 10]. Due to high complexity in computation of the convex hulls, depth-based methods are found to be impractical for higher dimensions [9]. Though depth-based methods could avoid the problem of finding out what kind distribution the dataset belongs to, it is basically not suitable for higher dimension dataset due to the larger computations involved in the convex hulls.

Data clustering algorithms have been developed to group data into clusters which can indirectly screen out outliers. Many clustering algorithms have been developed. CLARANS [22], BIRCH [23] and DBSCAN [24] have their own kind of approaches in detecting outliers. DBSCAN is more into detecting outliers when compared to other two algorithms. CLARANS is based on randomized search. These algorithms are more dependants on input parameters and are usually focused on clustering.

RELATED WORK

A popular method of identifying outliers is by examining the distance to an example's nearest neighbors. In this approach, one looks at the local neighborhood of points for an example typically by the k nearest examples (also known as neighbors). If the neighboring points are relatively close, then the example is considered normal; if the neighboring points are far away, then the example is considered unusual. The advantages of distance based outliers are that no explicit distribution needs to be determined unusualness, and that it can be applied to any feature space for which we can a distance measure. Given a distance measure on a feature space, there are many different definitions of distance-based outliers [11].

Three popular definitions are

1. Outliers are the examples for which there are less than p other examples within distance d [17, 18].

2. Outliers are the top n examples whose distance to the kth nearest neighbor is greatest [23].

3. Outliers are the top n examples whose average distance to the k nearest neighbors is greatest [11].

Distance based outliers is the focus of this thesis. We study about various definitions of DB outliers stated by different researchers, their methods of detection, the algorithms proposed, and reducing the time complexity by using NL algorithm using separators, etc. According to Hawkins, an outlier is an observation which is so much different from the other members of the dataset such that it gives a suspicion about its creation [8]. These outliers can be identified based on various factors one of which is distance. The term DB is an acronym for the word distance based outliers. Here the fraction p and distance D are user inputs. Distance-based methods are suitable for k-dimensional datasets. It is suitable for situations where the observed distribution does not fit any standard distribution. More importantly, it is well defined for k dimensional datasets for any value of k. Unlike the depth based methods, DB outliers are not restricted computationally to small values of k. While depth based methods rely on the computation of layers in the data space, DB outliers go beyond the data space and rely on the computation of distance values based on a metric distance function. We do not claim that DB outliers can replace all existing notions of outliers and can be used universally. Indeed, depth based outliers would be more applicable than DB outliers to situations where no reasonable metric distance function can be used.

There are many algorithms developed to detect distance-based outliers. But our area of focus would be the Nested loop algorithm which detects the distance-based outliers effectively. Nested loop algorithm is best suited for a dimensionality of k >4 while the cell based algorithms scales for dimensions lesser then 4. Various properties of distance-based outliers have been stated. To choose a proper value of the parameters p and D is left up to the user who would have to involve trial and error methods to frame the correct parameters [11].

The first algorithm proposed was the Index-based algorithm. This algorithm uses an underlying distance function which finds the distance between a pair of objects in the dataset. The D-neighborhood of an object o is a set which contains all the objects which are within a distance D. The fraction p is assumed to be count M, which is considered to be the count of the maximum number of objects within the D-neighborhood. Thus the algorithm is implemented by searching for objects within a radius D and keeps a count of it. When the count reaches greater than M, then the search stops to declare the object as a non-outlier otherwise the object is an outlier. This approach can deal with datasets with a dimensionality k >= 5 [11]. As the dimension k increases the time complexity of this algorithm equates to a linear running time. But for the worst case it, it has a time complexity of O (kn^2) with k as the dimension and n as the size of the dataset. The one thing which makes this algorithm unsuitable for market is the index building cost [11].

The next algorithm is the nested-loop algorithm. This algorithm rules out the necessity of maintaining an index. A total buffer size which is equal to B% of the size of the dataset is assumed. This algorithm divides the whole space into two halves which can be a temporary storage. Then data is loaded into these storages and then a tuple by tuple comparison is done to find out the neighbors of an object. If the count reaches greater than the count M, then the comparison stops to declare that object as a non-outlier. This algorithm implements a straight forward pair wise distance computations and tries to reduce the number of reads and writes of the dataset. The time complexity in this case is O (kn^2) [11].

Then they came up with an algorithm which is based on a cell structure. The members of the dataset are represented in the form of 2D cells. These cells are determined to be of size $l = D/(2\sqrt{2})$. Depending on this, the cells of the layer one neighbors of a cell C(k,m) at row k and column m, are defined to be those cells which are in the range of k= {x+1, x-1} and m={ y+1, y-1} where x and y are the rows and columns of the cells of the 2D space. This shows that there would be eight cells in the layer 1 for a particular cell. Based on this, some properties of cells are defined which say that any pair of objects inside a similar cell would be at a maximum distance of D/2 where D is the distance. The next property states that the distance between objects in a particular cell and its neighbor would be at most at a distance of D away from each other [11]. Based on this the next property states that if any cell is not a layer 1 or layer 2 neighbor of a cell c(x,y), then they would be apart at distance greater than D. Cell structure is used to help determine outliers and non outliers on a cell by cell basis, rather than on an object by object basis. This helps to reduce execution time significantly because we quickly rule out a large number of objects that cannot be outliers. But as the dimensions increases the run time increases exponentially

So, we use nested loop algorithm for higher dimensions greater than 3. But in nested loop algorithm, if one element is determined as an outlier, it should be compared with all the other elements in the whole dataset. The number of comparisons would be a large factor in the run time. This could be reduced by dividing the dataset with one attribute into two datasets, in such a way that the elements in one dataset which are dependent on the other dataset could be gathered into access area for the dataset.

DESCRIPTION OF NL ALGORITHM

Algorithm NL uses a block-oriented, nested-loop design for finding DB outliers. Assuming a total buffer size of B% of the dataset size, the algorithm divides the entire buffer space into two halves called the first and second arrays. It reads the dataset into the arrays, and directly computes the distance between each pair of objects or attributes.

For each object t in the first array, a count of its D-neighbors is maintained. Counting stops for a particular attribute whenever the number of D-neighbors exceeds M.

As an example, consider Algorithm NL with 50% buffering, and denote the 4 logical blocks of the dataset by A, B, C, D, with each block/array containing 1/4 of the dataset. Let us follow the algorithm, filling the arrays in the following order, and comparing:

1. A with A, then with B, C, D-for a total of 4block reads;
2. D with D (no read required), then with A (no read), B, C-for a total of 2 block reads;
3. C with C, then with D,A, B-for a total of 2 blocks reads; and
4. B with B, then with C, A, D-for a total of 2 block reads.

Thus, in this example, a grand total of 10 blocks are read, amounting to y = 2.5 passes over the entire dataset.

CHAPTER III

DESCRIPTION OF NL ALGORITHM USING SEPARATORS

This thesis is mainly focused in improving the NL algorithm using separators, which could remove the half of the unnecessary computation of searching for the neighbors. To remove the unnecessary computation the dataset has to be preprocessed which would take one pass over the whole dataset. Let us consider n objects with one of the attributes A ranging from 0 to x. Let the distance factor be considered D where D << x. let the average of the attribute be considered as AV.

The dataset is divided into two datasets with the element's attribute A's value lesser than or equal to AV as dataset DS1 and the rest of the elements as dataset DS2. Elements in DS1 with the attribute A's value ranging from (AV – D) to AV may have neighbors in DS2 with the attributes A's value ranging from AV to (AV + D).Elements in DS2 with the attribute A's value ranging from AV to (AV + D) may have neighbors in DS2 with the attributes A's value ranging from (AV-D) to AV. So, the elements in DS1 does not check the neighbors in DS2 with values > (AV+D) and the elements in DS2 does not check the neighbors in DS1 with values < (AV-D), which could reduce the number of comparisons in a large factor.

Elements in DS2 with the attribute A's value ranging from AV to (AV + D) is entered in a dataset DSArea1 .Elements in DS1 with the attribute A's value ranging from (AV-D) to AV is entered in a dataset DSArea2. After doing this preprocessing, the elements in dataset DS1 could check for neighbors in DS1 and DSArea1, and elements in dataset DS2 could check for neighbors in DS2 and DSArea2. This reduces the number of computation.

Assuming a total buffer size of memory to swap the array is B% of the dataset size, divides the entire buffer space into two halves called the first and second arrays. It reads the dataset into the arrays, and directly computes the distance between each pair of objects or attributes.

For each object t in the first array, a count of its D-neighbors is maintained. Counting stops for a particular attribute whenever the number of D-neighbors exceeds M.

Algorithm NL Using separators

Preprocessing:

1. Choose one of the attribute. Find the average AV of the values of the attribute in the whole data set DS.

2. While there are elements in the data set DS.

    a. If the attributes value is between lower range to AV write the values in dataset DS1.

    b. If the attributes value is between AV to higher range write the values in datasetDS2.

    c. If the attributes value is between (AV –D) to AV write the values in dataset DSArea1.

d. If the attributes value is between AV to (AV +D) write the values in dataset DSArea2.

Processing:

Divide the dataset in DS1 into logical blocks for finding outliers and DSArea1 as additional neighbor's area for DS1.

1. Fill the first array (of size B/2 % of the dataset) with a block of tuples from T.

2. For each tuple t_i in the first array, do

a. $count_i = 0$

b. For each tuple t_j in the first array, if $dist(t_i, t_j) \leq D$: Increment $count_i$ by 1. If $count_i > M$, mark t_i as a non outlier and proceed to next t_i.

3. While blocks remain to be compared to the first array, do:

a. Fill the second array with another block (but save a block which has never served as the first array, for last).

b. For each unmarked tuple t_i in the first array does:

For each tuple t_i in the second array, if dist $(t_i, t_j) \leq D$: Increment count by 1. If $count_{i >}$ M, mark t_i as a non outlier and proceed to next t_i.

4. For each unmarked tuple t_i in the first array, report t_i as an outlier.

5. If the second array has served as the first array anytime before, stop; otherwise, swap the names of the first and second arrays and go to step 2.

Repeat the process again from step 2 for DS2 with DSArea1 as additional neighbor's area.

As an example, consider Algorithm NL with 50% buffering, and denote the 4 logical blocks of the dataset by A, B, C, D, with each block/array containing 1/4 of the dataset DS1. Denote the logical blocks in DSArea1 by E,F. Let us follow the algorithm, filling the arrays in the following order, and comparing:

1. A with A, then with B, E, F, C, D.
2. D with D, then with A, E, F, B, C.
3. C with C, then with D, E, F, A, B.
4. B with B, then with C, E, F, A, B.

At the end of each step, the elements in the array with neighbors lesser than M are considered as outliers. Repeat the same process on dataset DS2 with DSArea2.

IMPLEMENTATION STRATEGY OF THE ALGORITHM USING SEPARATORS

This algorithm is implemented using Java. The preprocessing of data is done by subdividing the dataset into two sets X and Y, by calculating the average of one of the attribute A. The access area of each of the attributes from the other Dataset is X' and Y', where the attribute A has values that ranges from the average to the distance D (avg ± D). After preprocessing, the algorithm finds the outliers in X by finding the distance between each element to all the other elements in X. If the distance between the compared elements to the other element is lesser than the distance D then the count of neighbors is increased by 1. If the number of neighbors is equal to K then the element is considered as non outlier and the next element is compared with all the other elements. If the element is compared with all the element in X and still the number of neighbors is lesser than k, then the element‘s distance to the elements in X’ is calculated. If the distance between the compared elements to the other element is lesser than the distance D then the count of neighbors is increased by 1. If the element is compared with all the elements in X and X’ and the number of neighbors is lesser than k, then the element is considered as an outlier and the element is written in the outlier file.

After finding the outliers in X, the procedure is repeated again for the elements in Y, with Y’ as the additional access area.

CHAPTER IV

EXPERIMENTAL DATA AND RESULTS

The following is a table showing the time-complexity of the NL algorithm for various dataset sizes and varying dimensions. The runtime is measured in milliseconds. The dataset is generated using a Gaussian distributor function. The value of d which is the distance parameter is chose randomly according to the range of the dataset generated. A value of p close to unity is used to decide the neighbors. Based on the values on the tables below the time-complexity graph for both the methods are shown as a comparative analysis.

Table 1 Experimental run time of NL algorithm

Dimension k	1 MB	2 MB	3 MB	4 MB	5 MB
k = 2	5728 ms	8912 ms	13224 ms	28238 ms	41738 ms
k = 3	7224 ms	12273 ms	17458 ms	38476 ms	59641 ms
k = 5	32753 ms	51732 ms	72563 ms	151741 ms	171841 ms
k = 10	41771 ms	71839 ms	100852 ms	184536 ms	224879 ms
k = 15	100387 ms	155358 ms	198036 ms	255526 ms	365270 ms

Table 2 Experimental run time of NL algorithm using separator

Dimension k	1 MB	2 MB	3 MB	4 MB	5 MB
k = 2	3418 ms	7012 ms	9451 ms	17843 ms	26652 ms
k = 3	4918 ms	69648 ms	12574 ms	24339 ms	41978 ms
k = 5	20854 ms	31872 ms	40898 ms	87973 ms	102756 ms
k = 10	26831 ms	44210 ms	65436 ms	100432 ms	128465 ms
k = 15	61492 ms	88246 ms	112045 ms	143572 ms	245579 ms

The table above shows different running times for NL Algorithm using separator with dataset ranging from 1 MB to 5MB on each dimension and the dimensions varying from k=2 to k=15. These values are plotted in a graph to have a comparative analysis of time-complexities between the NL algorithm and the NL algorithm using separator..

COMPARATIVE ANALYSIS OF RUN-TIME FOR D=2

The figure below shows the graphical representation for a dimension D=2. The running times increase exponentially. The runtime of the NL Algorithm using separators is less than the NL algorithm on data files of any size. When the file size increases, we could notice the time difference is higher.

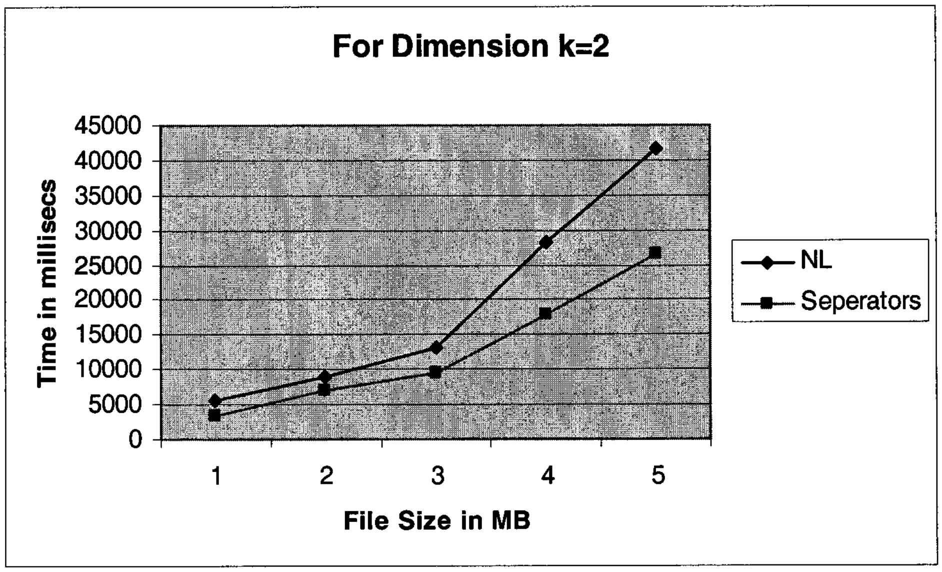


Figure 11. Comparative run-time analysis graph for dimension D=2

COMPARITIVE ANALYSIS OF RUN-TIME FOR D=3

The figure below shows the graphical representation for a dimension D=3. In this graph, the run time of the NL algorithm is lesser than the NL Algorithm using a separator. But the difference between the run-time is less on small files.. But the run time differs, when the file size increases.

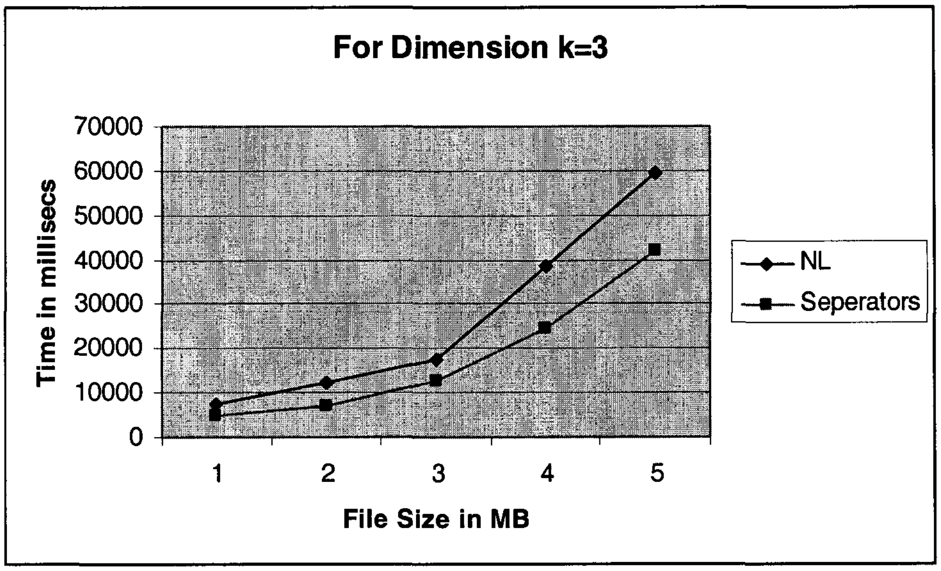


Figure 12. Comparative run-time analysis graph for dimension D=3

COMPARATIVE ANALYSIS OF RUN-TIME FOR D=5

The figure below shows the graphical representation for a dimension D=5. There is a constant difference between the run time of the NL algorithm and the NL Algorithm using separators, where the large data files is processed quickly by NL algorithm using separators.

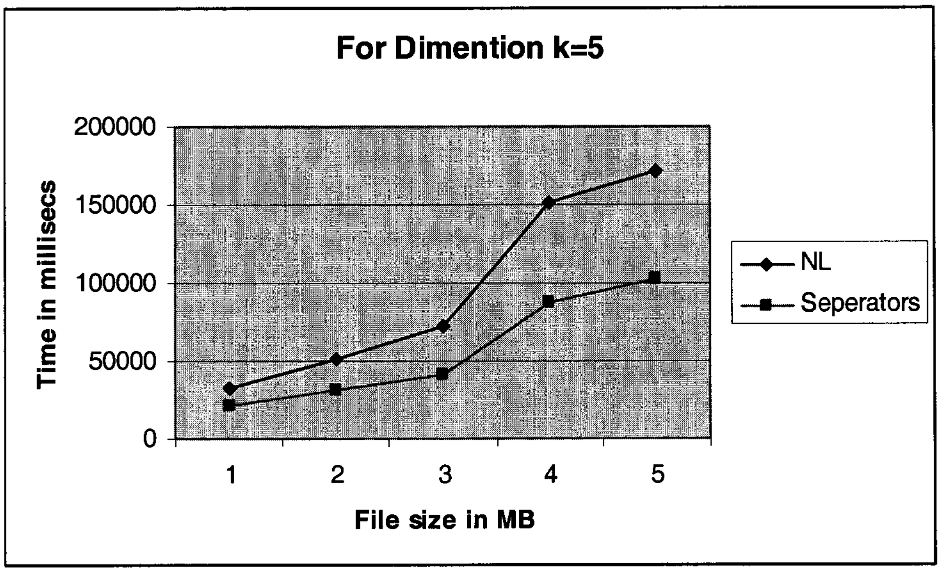


Figure 13. Comparative run-time analysis graph for dimension D=5

COMPARATIVE ANALYSIS OF RUN-TIME FOR D=10

The figure below shows the graphical representation for a dimension D=10. When the file size increases to 4 MB, the run time of the NL algorithm is twice the run time of the NL algorithm using separators.

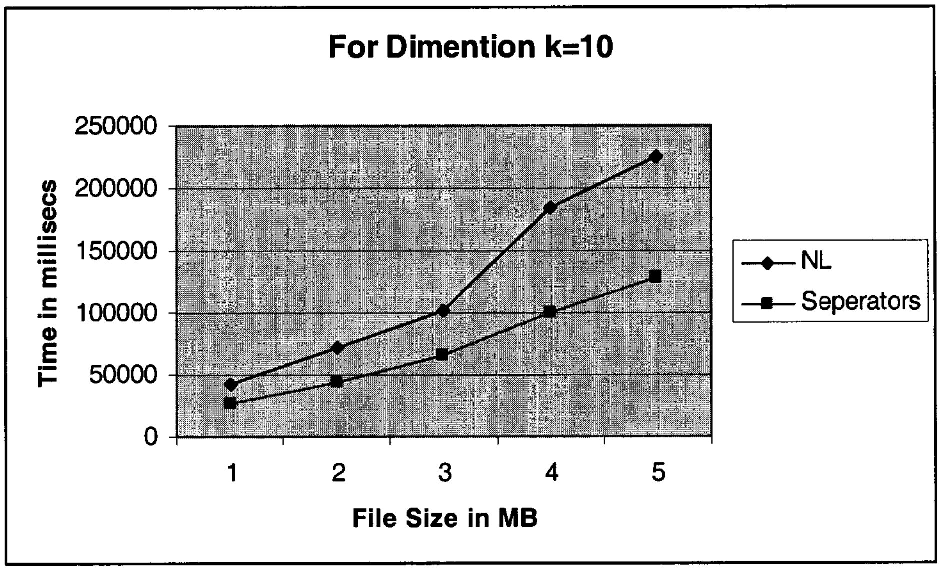


Figure 14. Comparative run-time analysis graph for dimension D=10

COMPARATIVE ANALYSIS OF RUN-TIME FOR D=15

The figure below shows the graphical representation for a dimension D=15. The runtime of the NL Algorithm using separators is less than the NL algorithm on data files of any size.

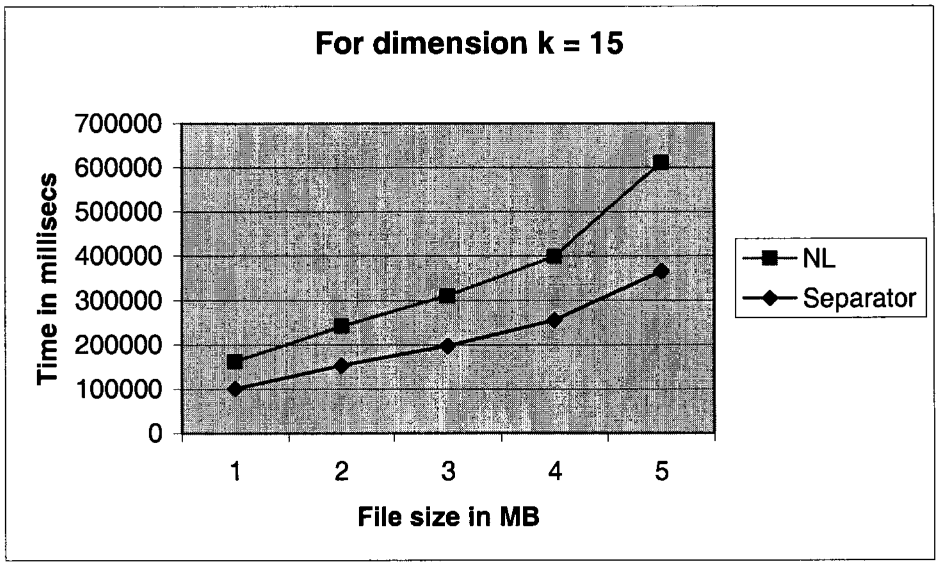


Figure 15. Comparative run-time analysis graph for dimension D=15

RESULTS

The results show that NL algorithm using separators is efficient than NL algorithm by reducing the number of computation for finding out , if an element is an Outlier or not. Dividing the dataset into two datasets using separators is the most important step in reducing the runtime of NL algorithm using separators. The algorithm also considers limited buffer memory for memory management on large database and a better strategy to reduce the memory swaps.

The separators value is determined using one attribute with higher range. We have tested the dataset for varying values of k and found that for large data sets, with value of distance D and less number of neighbors and, average of one the attribute as separator proved to reduce the run time of the NL algorithm. The reduction in the number of computations is the reason for the runtime of the algorithm.

The accurate determination of separator considering many attributes might further improve the complexity of this algorithm. Though, the run time could also be reduced by dividing the dataset into many small divisions by using many separators, determining many separators in a large dataset with unpredictable values could be a overhead in the preprocessing of the given data.

CHAPTER V

CONCLUSIONS

This thesis has given a detailed description of various aspects of outliers and has presented with the reduction of running time in NL Algorithm using separators.. Starting from the history of data mining, we have presented the various areas of data mining . In my thesis, Outliers have been given a detailed study. We started with the definition of outlier as an observation which is so much different from the other members of the dataset, their properties and attributes, various methods of detection. In order to implement the algorithms which mine the distance based outliers, we have given a detailed description of the existing algorithms and the NL algorithm that could be improved to reduce the runtime of the algorithm

Separators help to strengthen the NL algorithm by reducing the number of computations to determine outliers of a given dataset. Separators are determined using one of the attribute with a higher range of values and their average is computed. The steps for memory allocation are determined to reduce the number of memory swaps. Thus the distance-based outliers are explored and the NL algorithm using separators is used to detect these outliers from an n- dimensional dataset is studied [11].

Then a comparative analysis of the NL algorithms and the NL algorithm using separators is done where datasets of various sizes are tested for different values of dimensions. The algorithm is tested for data files of various size and dimensions. Their time-complexities are noted to represent them graphically by plotting the time-complexities against the varying size of the datasets for different dimensions. These graphs can be further used to infer the time taken to process a dataset of particular size. This would give us an improved understanding of the dataset which is the main aim of these algorithms and the purpose of using separators in this thesis.

ONGOING RESEARCH

Future work is going improve the algorithm by determining other factors that increases the computation and run time of the algorithm. This is believed to deliver outliers that are more relevant and meaningful. Another area which is given attention by various researchers is identifying outliers in clusters which are usually ignored. The study of this might result in various data patterns which would be of great interest to researchers [19]. Mining distance-based outliers have been using a parameter value close to unity. This has helped in identifying the isolated objects.

The researchers are now trying to detect the cluster of outliers for a condition where the user wants to identify the outliers from a single area. This would mean that an outlier cluster would have to be found where all the data's in the cluster are outliers [19]. As the separators are determined using one attribute, the dataset might not get the best efficiency, but would be better than the run time of NL algorithm. So, a better way to compute the separator considering all the attributes, with less overhead on preprocessing could improve the run time in the future.

Separators could be used in other methods of outlier detection like density based outliers to improve the algorithms. But, the way to determine the separators would differ for these methods. So, a better strategy to determine the separators which would reduce the computation has to be determined, process the given data and determine the outliers.

REFERENCES

[1] W. Frawley, G. Piatetsky-Shapiro and C. Matheus, "Knowledge discovery in databases: An overview", *Knowledge discovery in databases*, pp. 1-27, 1991.

[2] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", *Proc. ACM SIGMOD*, pp. 207-216, 1993.

[3] S. Brin, R. Motwani, J. Ullman, and S. Tsur, Dynamic itemset counting and implication rules for market basket data, *Proc. ACM SIGMOD*, pp. 255-264, 1997.

[4] R. Ng, L. Lakshmanan, J. Han and A. Pang, "Exploratory mining and pruning optimizations of constrained association rules", *Proc. ACM SIGMOD*, pp. 13-24, 1998.

[5] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2001.

[6] S. J. Stolfo, W. Lee, P. K. Chan, W. Fan, E. Eskin, "Data mining-based intrusion detectors: An overview of the Columbia IDS project", *SIGMOD Record*, 30(4):5-14, 2001.

[7] V. Barnett, T. Lewis, *Outliers in Statistical data*, John Wiley, 1994.

[8] D. Hawkins, *Identification of Outliers,* Chapman and Hall, London, 1980.

[9] I. Ruts and P. Rousseeuw, "Computing Depth Contours of Bivariate Point Clouds", *Computational Statistics and Data Analysis*, pp. 153-168, 1996.

[10] F. Preparata and M. Shamos, *Computational Geometry: An Introduction,* Springer-Verlag, 1988.

[11] E. Knorr and R. Ng, "Algorithms for Mining Distance-Based Outliers in Large Datasets", *Proc. VLDM*, pp. 392-403, 1998.

[12]http://www.anderson.ucla.edu./faculty/iason.frand/teacher/technologies/palace/datamin.htm

[13] U. Fayyad, D. Haussler, and P. Stolorz, "Mining Scientific data," *CACM*, 39(11):51-57, 1996.

[14] R. Porkess, *The HarperCollins Dictionary of Statistics* , HarperCollins, New york, 1991.

[15] S. Weisberg, *Applied Linear Regression,* John Wiley & Sons, 1985.

[16] R. A. White, "The detection and testing of multivariate outliers", Master's thesis, Dept. of Statistics, University of British Columbia, 1992.

[17] N. Draper and H. Smith, *Applied Regression Analysis,* John Wiley & Sons, 1996.

[18] R. L. Branham, *Scientific Data Analysis: An Introduction to Overdetermined Systems*, Springer-Verlag, 1990.

[19] E. Knorr and R. Ng, "Finding Intensional Knowledge of Distance-Based Outliers", Technical Report, Dept. of Computer Science, University of British Columbia, 1999.

[20] http://www.python.org.

[21] I. Bross, "Outliers in patterned experiments: A strategic appraisal", *Technometrics,* pp: 91-102, 1961.

[22] R. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining", *Proc. VLDB*, pp. 144-155.

[23] T. Zhang, R. Ramakrishna and M. Livny, "BIRCH: An efficient data clustering method for very large databases", *Proc. ACM SIGMOD,* pp. 103-114.

[24] http://www.exinfm.com/pdffiles/intro_dm.pdf

VITA

Nachiappan N Nachiappan,

505 N Bright CT, (700 W)
Salt lake City
Utah – 84116.

Education:

University of Texas – Pan American , Texas M. S. Major in CS 2005 December

C.S.I. Institute of technology Tamil Nadu, India B.E. Major in CS 1997 April

Work Experience:

2003 – 2005 Research Assistant, University of Texas- Pan American

2005 Teaching Assistant, University of Texas - Pan American,

Edinburg, Texas

2005 – Vitria Business ware administrator, Veteran Affairs, Salt lake City.

Current Thesis: Improving the performance of NL algorithm using separators.