University of Texas Rio Grande Valley

ScholarWorks @ UTRGV

Theses and Dissertations - UTB/UTPA

5-2005

Empirical performance analysis of two algorithms for mining intentional knowledge of distance-based outliers

Enbamoorthy Prasanthi
University of Texas-Pan American

Follow this and additional works at: https://scholarworks.utrgv.edu/leg_etd

Part of the Computer Sciences Commons

Recommended Citation

Prasanthi, Enbamoorthy, "Empirical performance analysis of two algorithms for mining intentional knowledge of distance-based outliers" (2005). *Theses and Dissertations - UTB/UTPA*. 772.
https://scholarworks.utrgv.edu/leg_etd/772

This Thesis is brought to you for free and open access by ScholarWorks @ UTRGV. It has been accepted for inclusion in Theses and Dissertations - UTB/UTPA by an authorized administrator of ScholarWorks @ UTRGV. For more information, please contact justin.white@utrgv.edu, william.flores01@utrgv.edu.

EMPIRICAL PERFORMANCE ANALYSIS OF TWO ALGORITHMS FOR MINING INTENTIONAL KNOWLEDGE OF DISTANCE-BASED OUTLIERS.

A Thesis

by

ENBAMOORTHY PRASANTHI

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

May 2005

Major Subject: Computer Science

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

EMPIRICAL PERFORMANCE ANALYSIS OF TWO ALGORITHMS FOR MINING INTENTIONAL KNOWLEDGE OF DISTANCE-BASED OUTLIERS.

A Thesis
by
ENBAMOORTHY PRASANTHI

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

May 2005

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ABSTRACT

Prasanthi, Enbamoorthy. Empirical Performance Analysis of Two Algorithms for Mining Intentional Knowledge of Distance-Based Outliers, Master of Science (MS), May, 2005, 47 pp., 25 Figures, references, 23 titles.

This thesis studies the empirical analysis of two algorithms, Uplattice and Jumplattice for mining intentional knowledge of distance-based outliers [19]. These algorithms detect strongest and weak outliers among them. Finding outliers is an important task required in major applications such as credit-card fraud detection, and the NHL statistical studies. Datasets of varying sizes have been tested to analyze the empirical values of these two algorithms. Effective data structures have been used to gain efficiency in memory-performance. The two algorithms provide intentional knowledge of the detected outliers which determines as to why an identified outlier is exceptional. This knowledge helps the user to analyze the validity of outliers and hence provides an improved understanding of the data.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

TABLE OF CONTENTS

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF TABLES

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF FIGURES

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER I

INTRODUCTION

Databases have become important means of storing large amounts of information. This makes it a necessity to develop methods which can efficiently extract information from these databases. Such a necessity has paved way for the popularity of data mining. Data mining which is otherwise called knowledge discovery in databases is defined as the retrieval of useful and unknown information from data in databases. Data mining and knowledge discovery go hand-in-hand where the latter is actually a part of the former one. Data Mining is technically defined as the non-trivial extraction of implicit, previously unknown, and potentially useful information from data [1]. This name was derived from the similarity between its procedure and the procedure of mining rocks for a vein of valuable core. The other synonyms to data mining are data dredging, knowledge extraction and pattern discovery. The data can be of varying kind and 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 1-3 below are examples of a relational database of a video store which maintains all the user information, time-series database etc. 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 database consists of a collection of records which represents

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

transactions. Multimedia databases store video, images, text, etc. Databases which store geographical information such as maps are called the spatial databases.

Customer

customerID	date	itemID	#	....
C1234	2005/01/01	98745	1	

customerID	name	address	birthdate	family_income	group	....
C1234	JohnSmith	120mainstreet	1965/10/10	$45000	A	

Figure 1 Sample relational database of a video store [24].

Rentals

transactionalID	Date	time	CustomerID	ITEMLIST
T12345	99/09/06	19:38	C1234	12,16,110,145,....

Figure 2 Sample transaction database [24].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

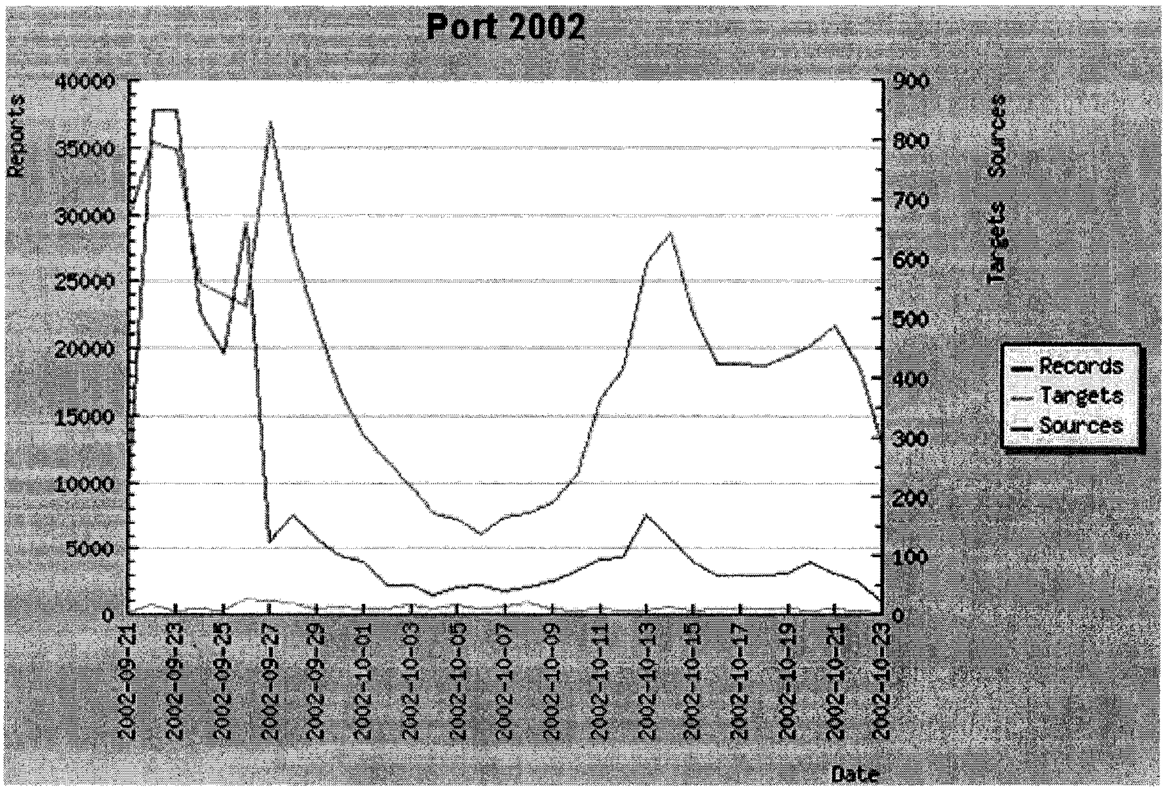

Figure 3 Sample time-series database [24].

Data mining techniques are 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. According to the recent survey analysis, researches detected unexpected similarities between the molecules which were earlier found to be unrelated. Data mining techniques have been useful in analyzing large volumes of data and gain their knowledge.

One data mining application which is focused a lot is the market basket problem [2, 3, 4]. This problem deals with the associations between the items of the customer's transactions. These rules are useful in deciding which items to be put on sale. These

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

association rules are used in many other applications such as stock market analysis, library usage, text mining, Web mining 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. Among such as credit card fraud or telephone calling card fraud, there may be some patterns existing which differ much from the other patterns [5]. 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 observations which deviate much from the other observations of a dataset. There 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 which 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 the dimensions are considered.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

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. These values being extreme are usually defined by the underlying model. In process such as manufacturing, 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 exits outside the range of μ-3σ and $\mu+3\sigma$, then such an observation is determined to be an outlier [25]. But these methods could only detect single outliers. When coming to the question of multiple outliers, they failed to detect them. Then the median strategy was introduced which could detect multiple outliers. All these earlier methods could only work with univariate or single-dimensional 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 phenomenon under study.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

PREVIOUS RESEARCH

There were different methods introduced to detect outliers. The outlier detection depends on various conditions such as the data distribution, the distribution parameters, the number of expected outliers 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 dataset is required to determine if an attribute fits a particular distribution. These disadvantages gave rise to depth-based outlier detection methods. In this method, the dataset is organized into various layers with the assumption that shallow layers contain outliers when compared to deep layers [9, 10]. But even these depth-based methods proved to be disadvantageous for higher dimensions because the computation of k-dimensional layers depends on the computation of k-dimensional convex hulls [11].

Then the notion of clustering was introduced. Clustering is basically grouping datasets based on similarities. Here the notion of outliers is defined indirectly which would optimize clustering more than outlier detection [22]. The next category of outlier detection is the distance-based methods. This thesis discusses about the properties of distance-based outliers and the various detection methods [11]. Once the outliers are detected, the intentional knowledge of these outliers is studied to gain knowledge about the data in the dataset. This intentional knowledge helps in evaluating the validity of the identified outliers and increases the understanding of the data [19]. To provide intentional knowledge, the notion of strongest and weak outliers is introduced. 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 [19].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

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 [8]. 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 [8]. 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 [24]. 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 [11]. They are suitable for higher dimensions. Hence they can handle many non-standard applications.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

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.

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

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

Clustering: Grouping data based on their similarities.

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

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].

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

Intentional knowledge: A description which says why an identified outlier is exceptional [19].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

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 does an empirical study of the algorithms used in detecting the distance-based outliers for k-dimensions with varying datasets [19]. All the earlier detection algorithms have advancement over each other in terms of time-complexities and methods of approach. Researchers considered various factors in developing their algorithms for detecting these outliers. Some felt that the 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

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

they came up with another factor called distance which could scale up to any number of dimensions and was feasible with large datasets. Various algorithms have been proposed in detecting outliers based on distance. Algorithms such as nested loop and cell based algorithms proved to give a time-complexity of O (kn^2) where k is the number of dimensions and this could handle large datasets [11]. The same people had also come out with an idea as to identify why an outlier is exceptional when compared with the other members of the dataset [19]. That kind of knowledge or information was named to be intentional knowledge. Algorithms named uplattice and jumplattice were developed which could drill out the intentional knowledge of these outliers using the notion of strongest and weak outliers [11].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

HISTORY OF 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 [24]. 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 at its best selling. This information is studied by researchers to find patterns and association between similar data [12].

This information is basically converted to knowledge. With this comes the concept of data warehouse which is nothing but integration of various databases from which data can be extracted [5]. Data mining is an important analytical tool used to manipulate databases. Nowadays everything needs a database to store information and manipulate it. Large datasets are needed to be stored in these databases. Hence associating these datasets is not an easy work. There need to be some kind of associations or patterns formed which could minimize the computations by grouping data [12].

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 [12]. For example, the details of a customer's visit and their orders from a restaurant. The next category is the clusters. Clusters are formed based on the logical relationship among the data. Then comes the associations and sequential patterns [12].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

There are various levels of analysis of data mining. The non-linear models which resemble neural networks can be used for data analysis. There are genetic algorithms which can analyze relevant data based on natural evolution [12]. 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]. Another application which is overlooked is the outlier detection ability. The concept of outliers is used in fraud detection applications such as credit-card fraud detections, telephone billing etc [4]. The next section discusses about these outliers and their relationship with data mining.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

OUTLIERS AND THEIR PROPERTIES

An outlier is an observation which differentiates itself from the rest of the members of the dataset. It is an observation which is exceptional. For example a student who scores extraordinarily high marks when compared to the other students of the class is considered to be an outlier. Researchers have come out with various definitions of these outliers. The various definitions of outliers are stated below.

- An outlier is an observation which appears to be varying with the other members of that dataset [7].
- An observation that is so much different from the other members such that it gives rise to suspicion about its method of creation [8].
- Observations that do not follow the same statistical model as the other members of the dataset are said to be outliers [15].

The figures below give a graphical representation of outliers. Data in the dataset is distributed in a 2D space where the outlying objects are considered to be outliers. In figure 4 the data distribution is represented in the form of a cluster at one end and sparsely at the other end of the graph. Hence the objects o1 and o2 which are lying away from the rest are considered to be outliers. In figure 5 data is grouped into clusters and the objects lying outside the clusters are outliers.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

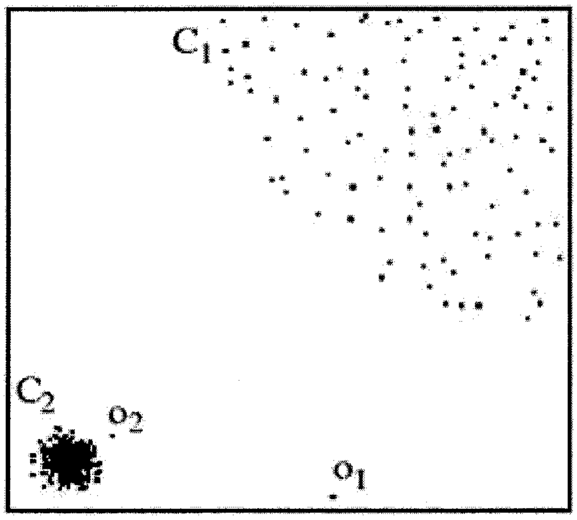

Figure 4 Graph representing outliers o1 and o2 among sparse and clustered distribution of data [26].

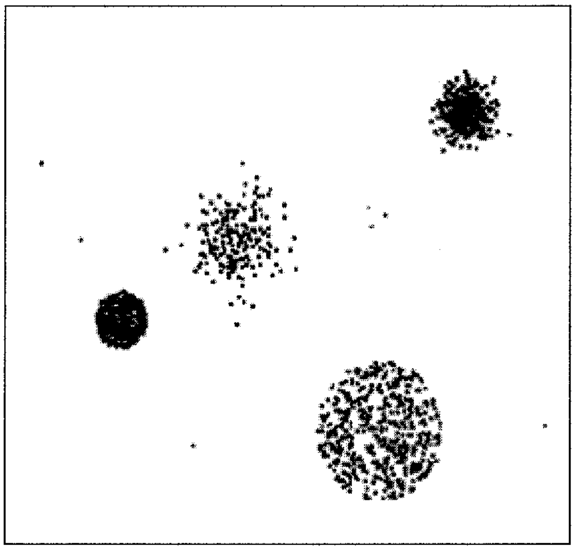

Figure 5 Graph representing outliers among various clustered distribution of data [26].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Outliers are of 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 those which 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 [16]. 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 as we discussed in the representations above. Hence outliers originate depending on the type of distribution. Draper and Smith stated that outliers could provide useful information They 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].

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. It is then data mining helps in mining data efficiently by using its mining tools.

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

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

developed. They were based on data distribution, distribution parameters, the number of expected outliers and their types [7]. Since outliers could be univariate or multivariate, they were several methods proposed by researchers to detect these outliers. The 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. This method is based upon quartiles and median. The 2D space is divided into upper and lower quartile [20]. A box is drawn form 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. Hence the extreme points can be detected using this method [20]. Figure 6 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.

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 evaluates the relationship between two variables. Scatter plots give answers to queries related to data such as how are the variables related, are there any outliers in the dataset, does the change of one variable depend on the other etc [20]. Figure 7 shows the graphical representation of a scatter plot. The X-axis represents one attribute while the Y-axis represents the other attribute. Different types of relationships between the attributes can be shown. For example, Figure 8 shows a linear relationship between data, Figure 9 shows no relationship among the two

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

attributes of a dataset and Figure 10 shows a quadratic relationship between the attributes of the dataset [20].

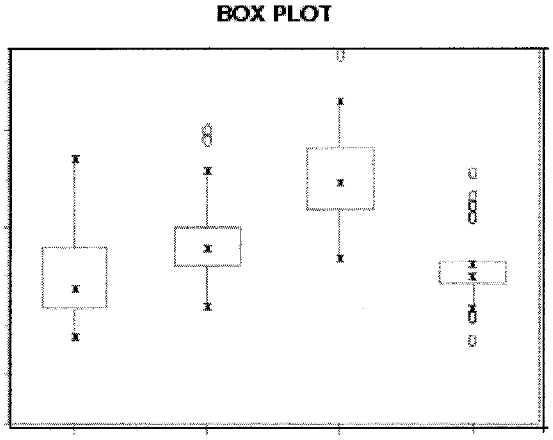

Figure 6 Representation of a box plot [20].

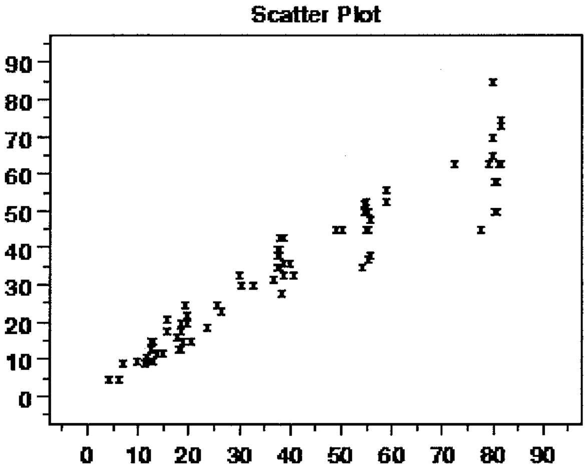

Figure 7 Representation of a scatter plot [20].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

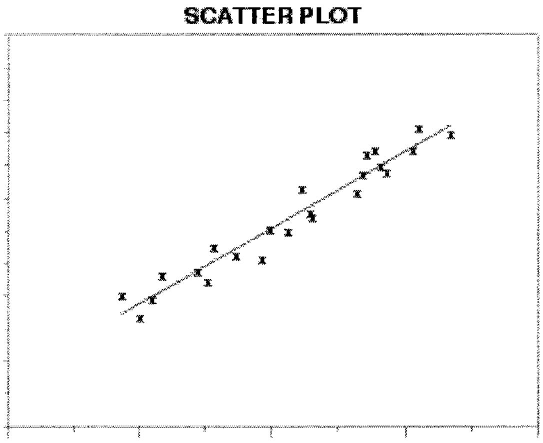

Figure 8 Scatter plot showing linear relationship between two attributes [20].

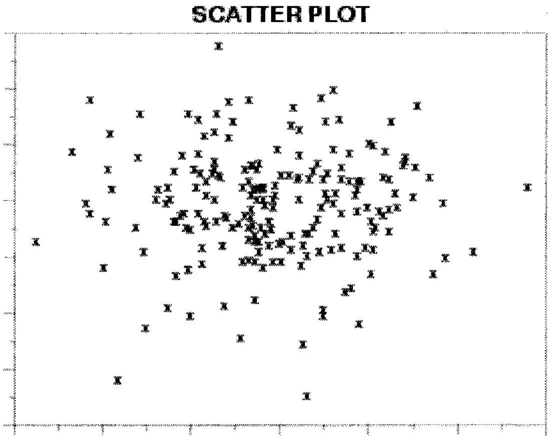

Figure 9 Scatter plot showing no relationship between two attributes [20].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

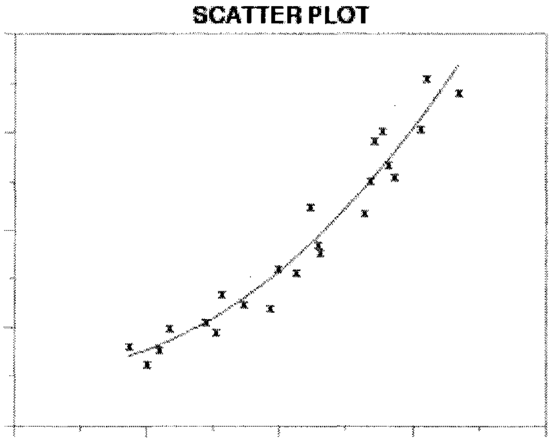

Figure 10 Scatter plot showing a quadratic relationship between two variables [20].

Since graphical methods could scale only up to a maximum of 3D dataset, researchers try to find some other ways of detecting outliers from datasets of any dimensions. It is then when 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 when the idea of distribution methods for outlier detection comes into existence. Some authors stated that outliers could be detected by plotting the residuals of a dataset and examining those residuals. By assumption, the residual values lying 3 or more deviations away from the mean value could be stated as outliers [17]. But some authors did not agree with this argument because they felt that this method 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 discordance 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 tests for normal, gamma and exponential distributions.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

But these tests were found to be unsuitable for multidimensional tests or large datasets, when distribution is unknown; the number of outliers is unknown, or when outliers are not the extreme values [19].

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 [9]. The concepts of peeling and depth contouring 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] although depth-based methods could avoid the problem of finding out what kind distribution the dataset belongs to.

Data clustering algorithms have been developed to group data into clusters which can indirectly screen out outliers. Many clustering algorithms, such as CLARANS [22], BIRCH [23] and DBSCAN [24], have their own kind of approaches in detecting outliers. These algorithms depend on input parameters and are usually focused on clustering. With two parameters which is a count of neighbors and the distance among the objects, distance-based methods for outlier detection have been developed which is discussed in the following section [11].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

DISTANCE-BASED OUTLIERS

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 time-complexities etc. We start with Hawkins's definition of outliers. According to the author, 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. Our area of interest is to study the distance-based outliers and to find intentional knowledge about these outliers along with the empirical analyses of the two algorithms uplattice and jumplattice established in [19].

According to [19, 11], an object in a dataset T is called a DB (p, D) outlier if at least p fraction of the objects in the dataset lie greater than distance D from the object o [11]. 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. They rule over depth based methods because depth based methods can work efficiently only for smaller values of k. The authors of [11] have developed their methods with an assumption that interaction with user and his ideas as an attribute would make the application even more efficient. Thus they have kept the two important parameters of this method as user-defined [11].

There are many algorithms developed to detect distance-based outliers. But our area of focus would mainly lie on the Nested loop and Cell algorithms [11] which detect the distance-based outliers effectively. Nested loop algorithm is best suited for a dimensionality of k >4 while the cell based algorithm scales for dimensions lesser then 4.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

There is also a partitioning algorithm proposed which would scale linearly with the number of data's but become exponential with respect to dimension k. Various properties of distance-based outliers have been stated. A sample NHL statistics has been used to test these algorithms. The choice of 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 [11]. 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 a count M which is considered to be the count of the number of objects within the D-neighborhood of an object [11]. Basically M is the neighbor count for a particular object o. Thus the algorithm searches for objects within a radius D of the object o 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]. The time complexity of this algorithm is, in the worst case, 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].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

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 [11]. 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 $l = D/2\sqrt{2}$ where D is distance parameter and l is the length of each cell [11]. 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 [11]. 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].

The layer 2 cells for a cell C(k,m) would be the cells within the range of k= { x+3,x-3} and m={y+3,y-3} and provided they are not members of the layer 1 of that particular cell [11]. Based on this the next property states that if any cell is not a layer 1

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

or layer 2 neighbor of a cell C(x,y), then they would be apart at distance greater than D. If a cell has objects greater than the count M, then none of its objects can be outliers. The time complexity of this algorithm is O(m+N) were m is the number of cells and N is the size of the dataset [11]. To scale with higher dimensions, a cell size of $l = D/2\sqrt{k}$ is required and the cells of the layers are determined accordingly.

INTENTIONAL KNOWLEDGE OF DISTANCE-BASED OUTLIERS

All the already existing methods on outlier detection only relate to the identification of outliers. There is not much study on the reasons as to why these outliers are identified to be exceptional. Such information or study is called to be the intentional knowledge of outliers. This intentional knowledge could prove to be a boon to the user in understanding as to what are the reasons which made the outliers exceptional, and are those reasons valid to declare them as outliers, and would increase the user's interpretation and understanding of the data. This thesis uses some of the efficient algorithms described above to detect the distance based outliers and then use them to find the intentional knowledge of those outliers by implementing the algorithms such as uplattice and jumplattice [19]. For example, the statistics of a class room might show that some students are outliers with respect to some combination of attributes but might not be the same when judged with some other combinations. This kind of intentional knowledge would improve our understanding of the data. To address the issue of intentional knowledge, concepts of strongest and weak outliers are to be introduced. An object is considered to be a strongest outlier in space A containing one or more outliers if there is

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

no outlier in any of the subspaces of the space A. For the condition above, if there exists an outlier and it is not the strongest outlier, then it is called a weak outlier. The algorithms used to implement this are discussed in the chapters to follow.

DESCRIPTION OF REMAINING CHAPTERS

The remainder of this document will discuss the implementation of the two algorithms uplattice and jumplattice used for performing this study, the results obtained, and the conclusions drawn from these results. It will further discuss the application and the areas in which this would prove useful in aiding the process of mining intentional knowledge from distance based outliers.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER III

DESCRIPTION OF UPLATTICE AND JUMP LATTICE ALGORITHMS

This thesis is mainly focused on studying the empirical analysis of the two algorithms, which determine the intentional knowledge of distance-based outliers. Before going into the details of the algorithms, we will discuss the idea of intentional knowledge and concepts related to it. To start with, the concepts of strongest and weak outliers need to be discussed. These concepts are similar to the subspace clustering discussed in the paper written by Agrawal [2]. Clusters are formed not only in the attribute space but also in the subspaces based on the already obtained information. Similarly the concept of strongest outliers is described based on the subsets of attributes [19].

The properties of distance-based outliers are studied to explore its strengths and then are classified into strongest and weak outliers. To clearly understand the concept of strongest outliers, let us consider an example where o is an outlier in 2D space and t is an outlier in 3D space. Then 'o' would be considered to be a strongest outlier when compared to the outlier t since in a 2D space two attributes are enough to decide that the object is a outlier while three attributes need to be considered in a 3D space Many inferences similar to this can be derived from the results. Once the outliers are identified, statistics such as the parameters required for identifying the object as an outlier, the distance of the neighbors in the dataset from the identified outlier, the count of neighbors

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

etc can be reported. To identify strongest outliers the attribute space needs to be divided into subsets of attribute spaces. For example an attribute set of 3 will contain 7 non-empty attribute subset spaces. Thus an attribute set with k elements will contain 2^k-1 non-empty subsets. An attribute space is considered to be a strongest outlying space if and only if, there is at least one outlier in it but there is no outlier found in any of its subspaces; else such a space is called a non-strongest space. Any outlier in the strongest outlying space is called a strongest outlier and any outlier in such a space which is not a strongest outlier is a weak outlier [19].

The algorithm uplattice finds strongest outliers in the lattice representation of attributes. The lattice representation of attributes is nothing but a level-by-level organization of all the non-empty subsets by their increasing order of cardinalities. Consider an attribute space containing a set of six attributes. The lattice representation of this attribute space will contain the subsets arranged in order from 1D till 6D in levels. Then each level is processed so that the strongest outliers can be found. The strongest subspaces and their super spaces are removed. This would help in ruling out the checking of many subspaces. Moreover if the user only wants the top-u outliers then, depending on the dimensionality and the number of strongest outliers found in the lower levels, only fewer subspaces would need to be considered. Similarly to search for weak outliers, we might only need to search for the non-strongest space if we have already found the strongest outliers. These strongest and weak outliers are termed as non-trivial outliers. Thus the uplattice procedure traverses the lattice in bottom-up method to find outliers in 1D space, 2D space and so on up to k-D spaces. This procedure is found to be very efficient if all the strongest outliers are found in level 1 itself [19].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Similarly the jumplattice procedure also traverses the lattice but with small differences. The uplattice procedure might undergo unnecessary computations before it encounters a space with outliers. To overcome this problem, the authors of [19] have come up with another method, which could jump from one node to a node in another level so that the unnecessary search could be avoided. In this procedure, when a subspace with cardinality k is found to have some outlier, then a small sub procedure called drilldown is used to prune this subspace with cardinality k. All the proper subsets of this subspace with decreasing order of cardinalities are formed and each one is processed to find out the outliers. The fact here is that if there is no outliers found in the starting subset, and then the procedure omits processing the rest of the subsets. Let us consider that the procedure jumplattice encountered a subspace A with cardinality k. This subspace A is passed on to the drilldown procedure, which would form all the proper subsets of A in decreasing order of cardinality. Let the first proper subset be B. This subset is processed to see if it has outliers. If there are no outliers found, then the procedure drilldown would not process any of the remaining proper subsets. Thus the purpose of the algorithm is saved [19].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

IMPLEMENTATION STRATEGY OF THE ALGORITHMS

These algorithms are implemented using the programming language Java. The algorithm uplattice requires a queue data structure to hold all the subsets of the attribute set. A set of attribute's N would have 2^N-1 non-empty subsets. A dynamic linked list is used to hold the subsets of the set A which are formed in accordance with their increasing order of their cardinalities. Each subset is then passed to the nested loop or the cell procedure based on the cardinality. If the cardinality is <=4 then the procedure cell is invoked else nested loop is invoked. The nested loop procedure mainly minimizes the number of reads and writes of the dataset. The nested loop procedure is also implemented using the linked list data structure, which holds b% of the dataset. Then it divides the dataset into two halves and then does the pair wise distance computations to detect the outliers. Once it detects the outliers, then the procedure uplattice removes all the supersets of that particular set from the list. This shows that once an attribute subspace A is found to have any outlier, then none of its superset can contain strongest outlier. This continues till the list is traversed completely [19].

Similarly for the algorithm jumplattice also, a linked list is used to store all the subsets of the attribute set A in increasing order of their cardinalities. Then the initial steps are the same as the procedure uplattice. Then each subset is processed using the nested loop or the cell algorithm depending on their cardinalities as in the uplattice algorithm. If any outlier is detected in the subset, then the supersets are removed from the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

list just like the algorithm uplattice. Algorithm jumplattice requires two parameters. One is the attribute set A and the other is the cardinality k. If this cardinality k is equal to the cardinality of the subset A, which was found to have outliers, then the procedure drilldown is invoked. This procedure also requires a list, which can hold the proper subsets of the set A, which was detected to have outliers. Then each proper subset is processed to find out if it has any outliers in the same way as it was done in the earlier jumplattice algorithm. If that particular proper subset does not contain any outlier, then procedure drilldown does not process any of the remaining proper subsets which is the proof of the fact that if no outlier exists in the subset, then there would not be any outliers in any of its subsets also. This minimizes the unnecessary processing of attribute subspaces, which takes place in algorithm uplattice [19]. The empirical performance analysis of these algorithms is presented in the following chapters.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER IV

EXPERIMENTAL DATA AND RESULTS

The following are two tables showing the time-complexity analysis of the algorithms uplattice and jumplattice for various sizes of the dataset and varying dimensions. The values in the tables are measured in milliseconds. The datasets are generated using a Gaussian distributor function. Thus 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 Empirical Analysis of algorithm uplattice

Dimension k	1000 tuples	2000 tuples	3000 tuples	4000 tuples	5000 tuples
k = 2	172 ms	609 ms	1156 ms	2218 ms	3094 ms
k = 3	438 ms	1203 ms	3594 ms	4032 ms	4281 ms
k = 5	1359 ms	4235 ms	6531 ms	5859 ms	17094 ms
k = 10	969 ms	10906 ms	21000 ms	26375 ms	40094 ms
k = 15	1547 ms	18079 ms	54860 ms	69843 ms	90969 ms

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 2 Empirical Analysis of algorithm jumplattice

Dimension k	1000 tuples	2000 tuples	3000 tuples	4000 tuples	5000 tuples
k = 2	219 ms	625 ms	1156 ms	3062 ms	4578 ms
k = 3	282 ms	1219 ms	3391ms	3782 ms	5750 ms
k = 5	2281 ms	8000 ms	17203 ms	29281ms	46688 ms
k = 10	41890 ms	191825 ms	269477 ms	739594 ms	946351 ms
k = 15	14703 ms	24750 ms	39656 ms	69203 ms	102968 ms

The table above shows different running times for the algorithm jumplattice for datasets varying from 1000 to 5000 tuples and the dimensions varying from k=2 to k=15. These values are plotted in a graph to have a comparative analysis of time-complexity between the uplattice and the jumplattice algorithms.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

COMPARATIVE ANALYSIS OF TIME-COMPLEXITY FOR 2-DIMENSIONAL DATASETS

The figure below shows the graphical representation of the empirical analysis of the algorithms jumplattice and uplattice for a dimension D=2. The running times increase exponentially. At certain points both are almost equal in their values. As the number of tuples increase and there are outliers found, and then the supersets are removed from the queue which brings down the running time, else it increases consistently. Here the value of k is chosen to be one.

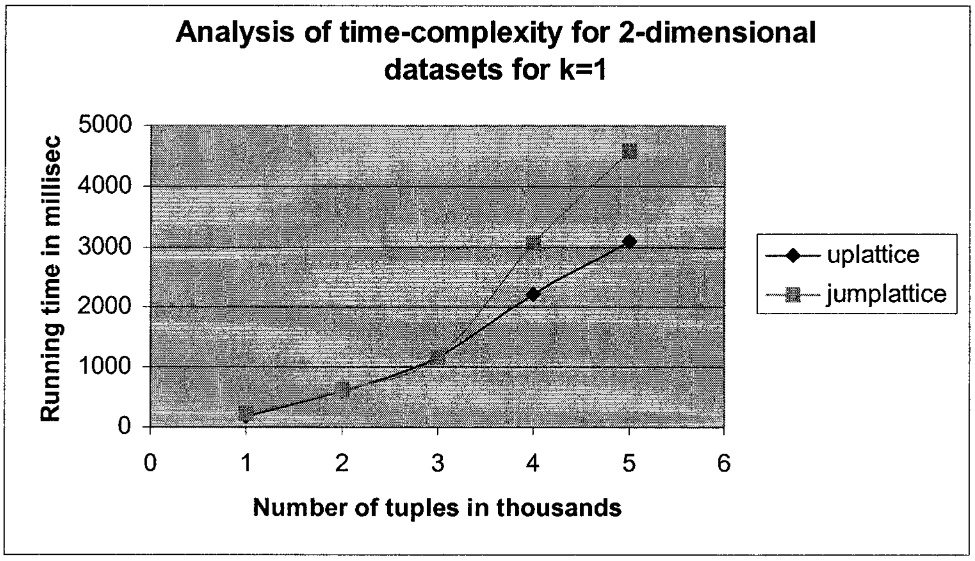

Figure 11 Comparative time-complexity analysis graph for 2-dimensional datasets.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

COMPARATIVE ANALYSIS OF TIME-COMPLEXITY FOR 3-DIMENSIONAL DATASETS

The figure below shows the graphical representation of the empirical analysis of the algorithms jumplattice and uplattice for a dimension D=3. In this graph for some values of d and k, the jumplattice procedure has a better running time when compared to the uplattice but then increases again. Here also the value of k is chosen to be one.

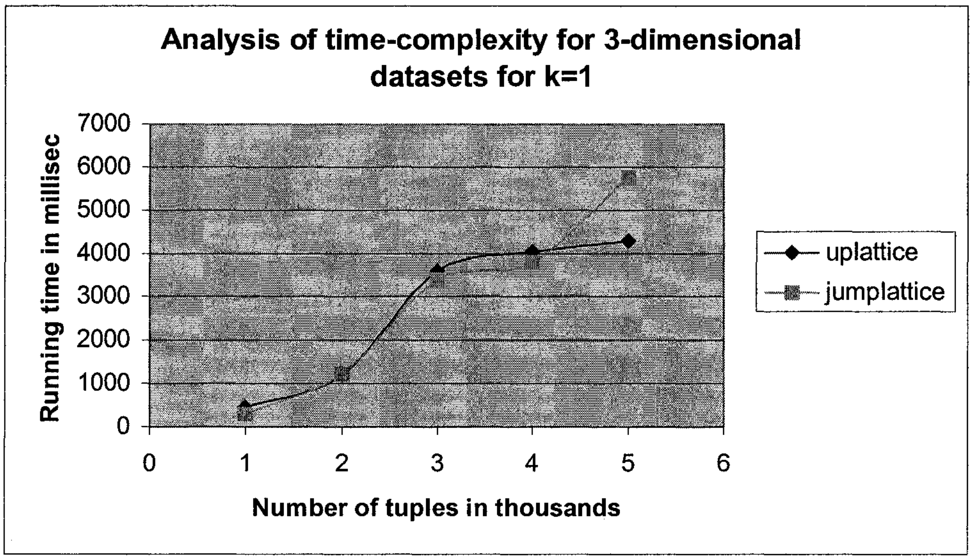

Figure 12 Comparative time-complexity analysis graph for 3-dimensional datasets.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

COMPARATIVE ANALYSIS OF TIME-COMPLEXITY FOR 5-DIMENSIONAL DATASETS

The figure below shows the graphical representation of the empirical analysis of the algorithms jumplattice and uplattice for a dimension D=5. In this graph the running time analysis of the procedure uplattice is much better when compared to jumplattice especially for larger values of the dataset. This shows that improper selection of the value k for the procedure jumplattice might result in an exponential growth of running time of the procedure. Here the value of k is chosen to be 2.

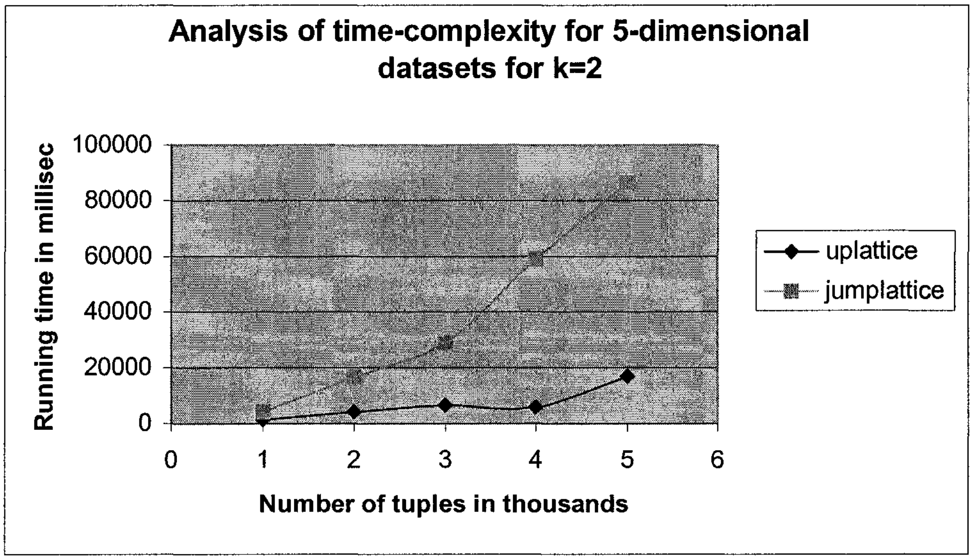

Figure 13 Comparative time-complexity analysis graph for 5-dimensional datasets.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

COMPARATIVE ANALYSIS OF TIME-COMPLEXITY FOR 10-DIMENSIONAL DATASETS

The figure below shows the graphical representation of the empirical analysis of algorithms jumplattice and uplattice for a dimension D=10. In this graph the running time analysis of the procedure uplattice is better when compared to jumplattice for a value of k=2 and for a large distance D. This again shows that a proper value of k would be must to bring down the running time of the algorithm.

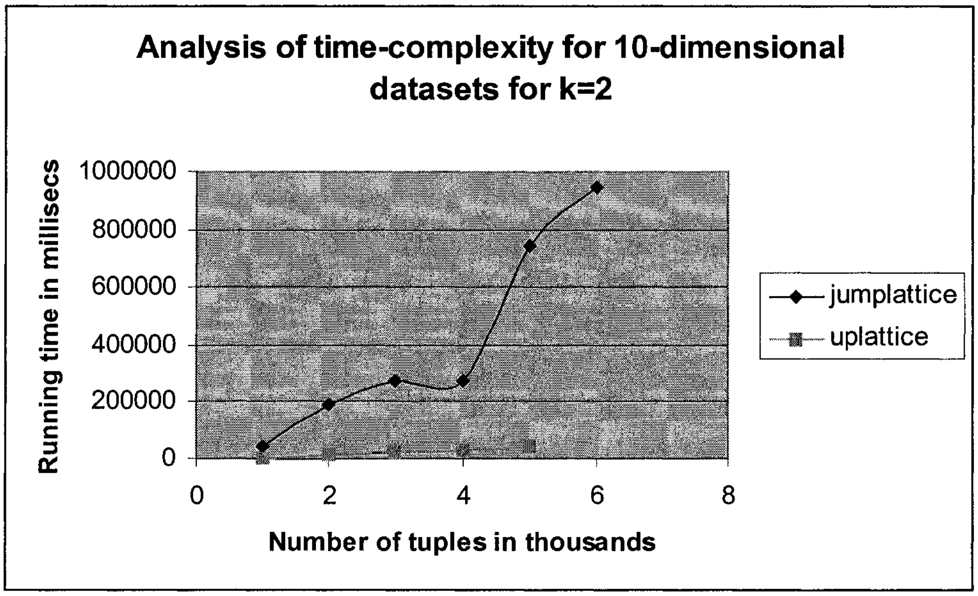

Figure 14 Comparative time-complexity analysis graph for 10-dimensional dataset.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

COMPARATIVE ANALYSIS OF TIME-COMPLEXITY FOR 15-DIMENSIONAL DATASETS

The figure below shows the graphical representation of the empirical analysis of algorithms jumplattice and uplattice for a dimension D=15. In the graph below there is a downfall in the slope of the jumplattice graph. This takes place if the value of k chosen is accurate and here k=2 which makes the algorithm check for outliers in that particular subset with attribute k. If there is an outlier, then all the subsets would be skipped from processing which would reduce the number of iterations and decrease the running time.

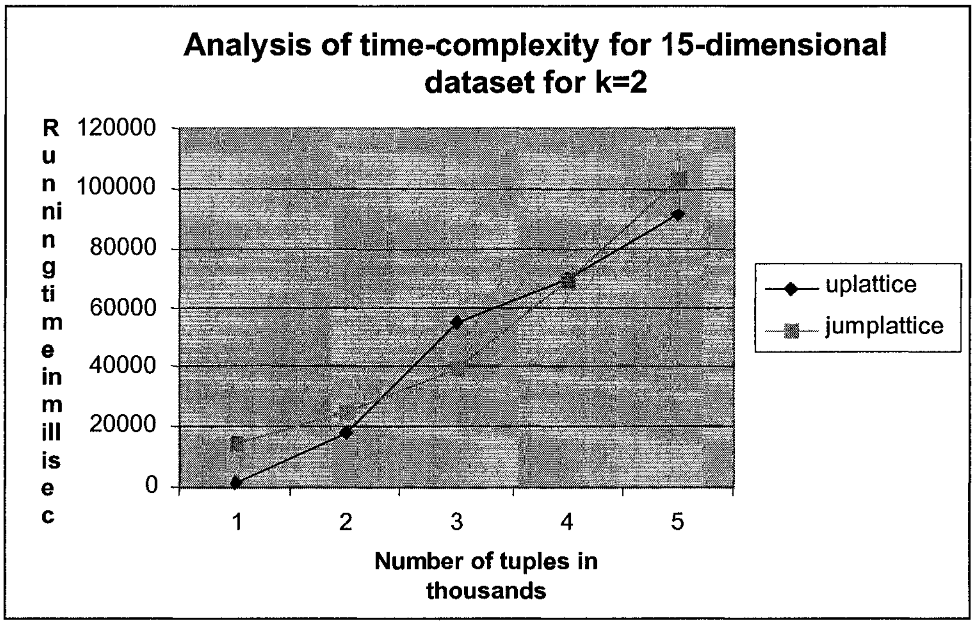

Figure 15 Comparative time-complexity analysis graph for 15-dimensional dataset.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

COMPARITIVE TIME-COMPLEXITY ANALYSIS FOR 10-DIMENSIONAL DATASETS FOR VARYING VALUES OF K

The graph shown below represents the variation in the running time of algorithm uplattice in comparison with algorithm jumplattice. For a 10-dimensional dataset, the algorithm jumplattice varies its running time in accordance with value of k. For k =1 to 3 the algorithm jumplattice has a better running time when compared to uplattice, but as k increases the running time grows exponentially. This shows that, for an appropriate value of k, the running time might be as expected otherwise, a worst case complexity occurs.

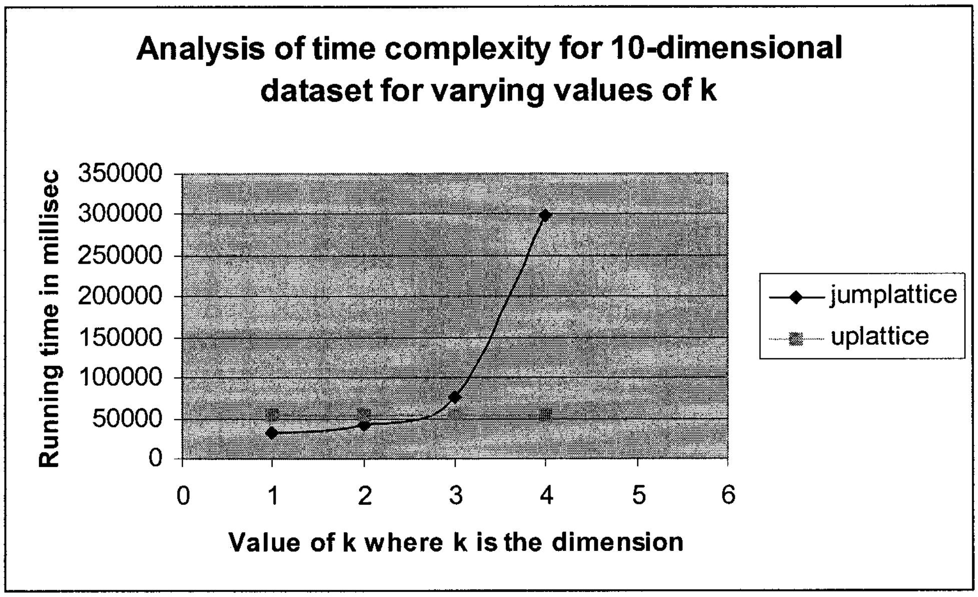

Figure 16 Comparative time-complexity analysis graph for 10-dimensional datasets for varying values of k

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

RESULTS

The results show that though jumplattice algorithm was developed to be more efficient than uplattice, it requires a good estimation of the value k which is passed to the drilldown procedure. With k=2 for the 5-dimensional, 10-dimensional and 15-dimensional datasets, the graph represents the growth in the running time which is sometimes low for uplattice and sometimes for jumplattice. This shows that an appropriate choice for k might yield better results. If the value is not a precise one, then the results would be as given above. Thus according to the results we can derive that when the subspace of an attribute set contains no outliers, then the algorithm jumplattice skips all the subsets provided the cardinality of that subspace is k whereas the algorithm uplattice processes all the subsets and does a wasteful effort.

Similarly, if for an accurate value of k, the jumplattice drills down the subset to find outliers and if there exists any, then none of the subsets of that subspace is processed thereby reducing the run-time. But for a random value of k, the uplattice algorithm might prove better than the jumplattice especially when the size of the dataset increases with the increase in dimensions. Hence the determination of an accurate value for k is still needed to be considered.

We have tested the dataset for varying values of k and found that for a large value of distance D and less number of neighbors and, an appropriate k value proved to preserve the goal of running a lower time-complexity as stated in the algorithm. The

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

accurate determination of k still remains a question and might prove an important area to research on to further improve the complexity of this algorithm.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER V

CONCLUSIONS

This thesis has given a detailed description of various aspects of outliers and has presented with the empirical performance analysis of the algorithms uplattice and jumplattice for mining intentional knowledge of distance-based outliers. Starting from the history of data mining, we have presented the various arenas where data mining is of a great use to researchers. 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 etc. In order to implement the algorithms which mine the intentional knowledge, we have given a detailed description of the algorithms used in detecting distance-based outliers. The strength of distance-based outliers facilitates the mining of intentional knowledge of these outliers. Thus the properties of distance-based outliers are explored and the algorithms used to detect these outliers from a k-dimensional dataset is studied [11].

Once the methods of detection of outliers are studied, we now move to the concept of our thesis which is mining intentional knowledge of these outliers. Before the notion of intentional knowledge is explained, the concept of strongest and weak outliers is given attention. We then analyze the reasons for identifying an outlier as exceptional and what set of attributes are needed to identify them. For this a lattice representation of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the data space is explained where the subspaces are organized into various levels so that we could implement a bottom-up approach in searching for outliers in the data space as proposed by the author in [19]. Then we give the implementation strategy for the algorithms uplattice and jumplattice with the explanation of the idea behind them. These algorithms mainly detect the strongest and weak outliers according to their idea of implementation as proposed by the authors in [19].

Then an empirical performance analysis of the algorithms is done where datasets of various sizes is tested for different values of cardinality. Their time-complexity is noted to represent them graphically by plotting the time-complexity against the varying sizes of the datasets for different cardinalities. These graphs can be further used to infer the time taken to process a dataset of a particular size. Moreover, an estimate of outliers for a particular combination of attributes can also be determined. This would give us an improved understanding of the dataset which is the main aim of these algorithms and the purpose of this thesis.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ONGOING RESEARCH

Future work is going on to combine the robust space transformations with the distance function. This is believed to deliver outliers that are more relevant and meaningful [19]. Mining distance-based outliers have been using a parameter value close to unity. This has helped in identifying the isolated points. These points would have been away from the cluster of data.

The algorithms studied in this thesis do not focus their attention on the missing values or attributes. This may sometime prove to be a hurdle to an application development for cases such as medical databases. Because in this case, missing attributes could be any important information such as a surgery, instrument etc. Thus this is an area for consideration to avoid such issues [19].

Similarly the memory management of cell-based structures is also considered to be an important issue. These cells might not fit in the memory for large datasets. This might require splitting the whole cell structure into small pieces and then process them. Due to this splitting, some cells might be omitted which contain no information and which can be derived from the other cells. An issue of which cells to examine would arise. This is into consideration because many cells can be omitted from being processed with the assumption that corner cells might contain more outliers when compared to the centre cells. Another issue is that these cells can be processed in parallel to reduce the computation time. As soon as one process discovers an outlier in an attribute subspace, it

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

can declare it to the other processes so that the supersets of that space can be stopped from being processed. [19].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

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.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[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.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[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.

[25] D. Freedman, R. Pisani, and R. Purves, *Statistics*, W.W. Norton, New York, 1978.

[26] http://www-users.cs.umn.edu/~kumar/Presentation/minds.ppt#638,10,MINDS - Anomaly Detection.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

VITA

Enbamoorthy Prasanthi

1105, west mahl street, apt #9
Edinburg, Texas 78539

Education:

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

VLB Engg College of Engg &Tech, Coimbatore, India B.E. Major in CS 1998 April

Work Experience:

2003 – 2004 Research Assistant, University of Texas- Pan American

2004 – 2005 Teaching Assistant, University of Texas - Pan American,

Edinburg, Texas

Current Thesis: Empirical performance analysis of two algorithms uplattice and jumplattice for mining intentional knowledge of distance-based outliers.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.