ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 2, February 2013

DETECTION OF NOISE BY EFFICIENT HIERARCHICAL BIRCH ALGORITHM FOR LARGE DATA SETS

V.S.Jagadeeswaran[1], P.uma[2]

Assistant professor, Department of Information Technology, Dr.N.G.P Arts and Science College, Coimbatore, India[1]
M.phil Research Scholar, Department of Computer Science, Dr.N.G.P Arts and Science College, Coimbatore, India[2]

ABSTRACT: **Data mining plays a vital role in Computer Field. A huge and valuable Knowledge is extracted from the large collection of data. Various techniques and algorithms are used for finding patterns from the large datasets.. Finding useful patterns in large datasets has attracted considerable interest recently, and one of the most widely studied problems in this area is the identification of clusters or densely populated regions, in a multi-dimensional dataset. Clustering is one of the main techniques for grouping the data items based on their similarity. Outlier detection is one of the outstanding data mining tasks. Clustering methods have efficient algorithms for finding Outliers. Outlier detection has important applications in various data mining domains such as fraud detection, intrusion detection, customer's behavior and employee's performance analysis. In this paper we have taken the agriculture datasets for finding Outlier detection. Hierarchical Clustering methods have been compared and considered BIRCH Algorithm to be the best for finding noise and very effective for large datasets than the other hierarchical algorithms**

Keywords: Clustering, Outlier detection Hierarchicalalgorithms, BIRCH

I. INTRODUCTION

Data mining is the non-trivial method of identifying valid, potentially useful, and finally understandable patterns in data[1].Data mining is the method of Extracting patterns from data. It can be used to uncover patterns in data but is often carried out only on sample of data. Clustering has been widely used in areas such as pattern recognition, image processing and data analysis. Clustering has been recognizes as primary data mining method for knowledge discovery. It is an important technique used for outlier

analysis. Outlier detection has direct applications in a wide variety of domains such as mining for anomalies to detect network intrusion, fraud detection in mobile phone industry and recently for detecting terrorism related activities [2].Outlier detection based on clustering approach provides new positive results.

II. OVERVIEW OF OUTLIER DETECTION

Outlier detection is an important branch in data mining, which is the discovery of data that deviate a lot from other patterns. An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was

generated by a different mechanism. There are many studied have been conducted on outlier detection for large datasets. A lot of work has been done in this area of research which is detecting outliers. Some of the outlier detection techniques are as follows

- Distance based outlier detection
- Clustering based outlier detection
- Density based outlier detection
- Depth based outlier detection

Each of these techniques has two steps for finding outliers. The first identifies an outlier around a data set using a set of inliers (normal data).In the second step, a data request is analysed and identified as outlier when its attributes different from the attributes of inliers. All these techniques assume that all normal instances will be similar, while the

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 2, February 2013

anomalies will be different. A key challenge in outlier detection is that it involves exploring the unseen space. It is hard to enumerate all possible normal behaviours in an application. Handling noise in outlier detection is a challenge. Noise may distort the normal objects and blur the distinction between normal objects and outliers. It may help to hide outliers and reduce the effectiveness of outlier detection. Outliers being the most excessive observations may include the sample minimum or sample maximum or both depending on whether they extremely high or low. However the sample minimum or sample maximum is not always outliers because they may not be abnormally distant from other comments. Many statistical techniques are sensitive to the occurrence of outliers. Checking for outliers should be a usual part of any data analysis. This can be due to incidental systematic errors or flaws in the theory that has been generated

III. CLUSTERING TECHNIQUES

Clustering is a division of data into groups of similar objects. Each group, called cluster consist of objects that are similar between themselves and dissimilar to objects of other groups. Representing data by fewer clusters necessarily loses certain fine details, but achieves simplification. It represents many data objects by few clusters, and hence it models data by its cluster. Data modeling puts clustering in a historical perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning and the resulting system represents a data concept.

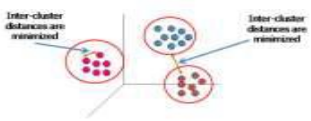

Figure1: Inter and Intra Class

The quality of a clustering result also depends on both the similarity measure used by the method and its implementation also the hidden patterns. Traditionally clustering are broadly divided into

- Partitioning methods
- Hierarchical methods
- Density based methods

Now Grid based clustering methods has been used in most of the field. These focus on spatial data i.e. data that model the geometric structure of objects in the space, their relationships, properties and operations. This technique quantizes the data set into a number of cells and then work with objects belonging to these cells. They do no relocate points but ratter builds several hierarchical levels of groups of objects. The merging of grids and consequently clusters does not depend on a distance measure. It is determined by a predefined parameter. In this paper we have compared the Hierarchical method algorithms for finding outlier detection for large data sets.

A. Hierarchical Methods

Hierarchical Clustering is the process of forming a maximal collection of subsets of objects (called clusters),with the property that any two clusters are either disjoint or nested. Equivalently, it can be viewed as forming a rooted binary tree having the objects as its leaves, the clusters the correspond to the leaves of sub trees. Hierarchical clustering creates a hierarchy of clusters, which may represent in a tree structure called a dendogram. The root of the tree consists of a single cluster containing all observations, and the leaves correspond to individual observations. Algorithms for hierarchical clustering are generally either agglomerative, in which one starts at the leaves and successively merges clusters together; or divisive, in which one starts at the root and recursively splits the clusters. Any valid metric use as a measure of similarity between pairs of observations.

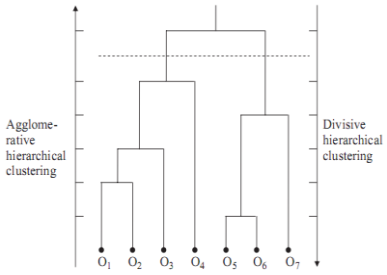

Figure2: Dendogram of Hierarchical

Agglomerative clustering starts with N clusters, each of which includes exactly one data point. A series of merge operations then followed, that eventually forces all objects into the same group. Hierarchical algorithms find successive clusters using previously established clusters. These algorithms can be either agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms begin with each element as a separate cluster and merge them into

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 2, February 2013

successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. . The basics of hierarchical clustering include Lance-Williams formula, idea of conceptual clustering, now classic algorithms SLINK, COBWEB as well as newer Algorithms CURE, CHAMELEON and BIRCH. Compared with the other algorithms BIRCH detected the noise better than the other ones for large datasets.

IV. HIERARCHICAL BIRCH ALGORITHM

In contrast to earlier work, an important contribution of BIRCH is the formulation of the clustering problem in away that is appropriate for very large datasets by making the time and memory constraints explicit. Another contribution is that BIRCH exploits the observation that the data space is usually not uniformly occupied, and hence not every data point is equally important for clustering purposes. So BIRCH treats a dense region of points (or a sub clusters) collectively by storing a compact summarization. BIRCH thereby reduces the problem of clustering the original data points into one of clustering the set of summaries, which is much smaller than the original dataset. Compared with prior distance-based algorithms, BIRCH is *incremental* in the sense that clustering decisions are made without scanning all data points or all currently existing clusters. If we omit the optional Phase 4 (Section 5). Compared with prior probability-based algorithms, BIRCH tries to make the best use of the available memory to derive the finest possible sub clusters (to ensure accuracy) while minimizing I/O costs (to ensure efficiency) by organizing the clustering and reducing process using an in-memory *balanced* tree structure of bounded size. Finally BIRCH does not assume that the probability distributions on separate attributes are independent.

A. CF Tree

A CF-tree is a height-balanced tree with two parameters: branching factor(B for non leaf node and L for leaf node) and threshold T. Each non leaf node contains at most B entries of the form[*CFi; childi]* where i=1,2,..B,'child' is a pointer to its i-th child node., and *CFi* is the CF entry of the sub cluster represented by this child. A leaf node contains almost *L* entries, and each entry is a CF. In addition, each leaf node has two pointers, '*prev*' and '*next*', which are used to chain all leaf nodes together for efficient scans. A leaf node also represents a sub cluster made up of all the sub clusters represented by its entries. But all entries in a leaf node must satisfy a *threshold requirement*, with respect to a threshold value T: *the diameter (alternatively, the radius) of each leaf entry has to be less than T*. The tree size is a function of *T*. The larger *T* is, the smaller the tree is. We require a node to fit in a page of size *P*, where *P* is a parameter of BIRCH. Once the dimension *d* of the data space is given, the sizes of leaf and no leaf entries are known, and then *B* and *L* are determined by *P*. So *P* can be varied for performance tuning. It is used to guide a new insertion into the correct sub cluster for clustering purposes just as a B+-tree is used to guide a new insertion into the correct position for sorting purposes. However the CF-tree is a very compact representation of the dataset because each entry in a leaf node is not a single data point but a sub cluster (which absorbs as many data points as the specific threshold value allows).

A. BIRCH Algorithm

Figure presents the overview of BIRCH. It consists of four phases: (1) Loading, (2)Optional Condensing, (3) Global Clustering, and (4) Optional Refining. The main task of Phase 1 is to scan all data and build an initial in-memory CF-tree using the given amount of memory and recycling space on disk. This CF-tree tries to reflect the clustering information of the dataset in as much detail as possible subject to the memory limits. With crowded data points grouped into sub clusters, and sparse data points removed as outliers, this phase creates an in-memory summary of the data. More details of Phase

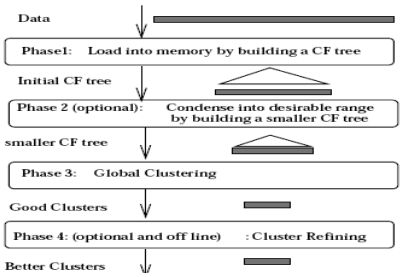

FIGURE3:BIRCH OVER VIEW

After Phase 1, subsequent computations in later phases will be: (1) fast because (a) no I/O operations are needed, and (b) the problem of clustering the original data is reduced to a smaller problem of clustering the sub clusters in the leaf entries; (2) accurate because (a) outliers can be eliminated, and (b) the remaining data is described at the finest granularity that can be achieved given the available memory; (3) less order sensitive because the leaf entries of the initial tree form an input order containing better data locality compared with the arbitrary original data input order. Once all the clustering information is loaded into the

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 2, February 2013

in-memory CF-tree, we can use an existing global or semi-global algorithm in Phase 3 to cluster all the leaf entries across the boundaries of different nodes. This way we can overcome Anomaly 1, (Section 4.4) which causes the CF-tree nodes to be unfaithful to the actual clusters in the data. We observe that existing clustering algorithms (e.g., HC, KMEANS and CLARANS) that work with a set of data points can be readily adapted to work with a set of sub clusters, each described by its CF entry. Phase 2 is an optional phase. With experimentation, we have observed that the global or semi-global clustering methods. After Phase 3, we obtain a set of clusters that captures the major distribution patterns in the data. However, minor and localized inaccuracies might exist because of (1) the rare misplacement problem (Anomaly 2 in Section 4.4), and (2) the fact that Phase 3 is applied on a coarse summary of the data. Phase 4 is optional and entails the cost of additional passes over the data to correct those inaccuracies and refine the clusters further. Note that up to this point, the original data has only been scanned once, although the tree may have been rebuilt multiple times. Phase 4 uses the centroids of the clusters produced by Phase 3 as seeds, and redistributes the data points to its closest seed to obtain a set of new clusters. Not only does this allow points belonging to a cluster to migrate, but also it ensures that all copies of a given data point go to the same cluster. Phase 4 can be extended with additional passes if desired by the user, and it has been proved to converge to a minimum (Gersho and Gray, 1992). As a bonus, during this pass, each data point can be labelled with the cluster that it belongs to, if we wish to identify the data points in each cluster. Phase 4 also provides us with the option. of discarding outliers. That is, a point which is too far from its closest seed can be treated as an outlier and not included in the result.

V. PERFORMANCE FACTOR

The performance of hierarchical BIRCH clustering algorithms is presented in this section. Agriculture data set has been taken into consideration and their outlier detection and time complexity is considered.

A Outlier Accuracy

Outlier detection accuracy is calculated in order to find out more number of outliers detected. Compared with the other Hierarchical algorithms Like CURE and CHAMELEON. The hierarchical BIRCH algorithm finds the outlier better.. The following chat shows the difference

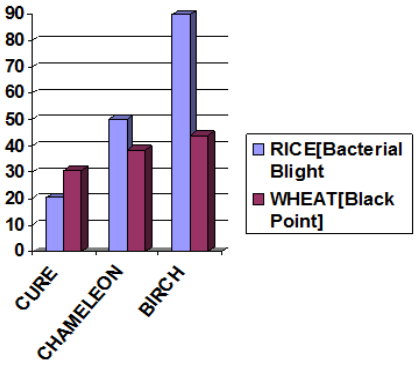

Figure4: Comparative graph for Hierarchical algorithms

In the above Chart the rice and wheat diseases has been find out as outliers. Compared with CURE and CHAMELEON,BIRCH has detected more number of outliers compared with the other ones for large data sets.

VI .CONCLUSION

Clustering is one of the techniques in Data mining for grouping similar objects. In this paper we compared the Hierarchical clustering algorithm for finding outlier detection for large datasets. We have taken the agriculture datasets for finding outliers in Rice and Wheat diseases. BIRCH Algorithm finds better accuracy results for finding noise than the other ones for large datasets.

REFERENCES

[1] Arun K Pujari: "Data Mining Techniques", Universities Press(India)Private Limited 2001.
[2] Ajay Challagalla,S.S.Shivaji Dhiraj,D.V.L.N Somayajulu,Toms Shaji Mathew,Saurav Tiwari,Syed Sharique Ahmad "Privacy Preserving Outlier Detection Using Hierarchical Clustering Method,34th Annual IEEE Computer Software and Applications Conference Workshops 2010.
[3] Tian Zhang,Raghu Ramakrishnan,Miron Livny,"BIRCH-A New Data Clustering Algorithm and its Applications",Data Mining and Knowledge Discovery,1,141-182(1997).
[4] George Kollios,Dimitrios,Gwopulos,Nick Koudas,Stefan Berchtold "Efficient Biased Sampling for Approximate Clustering and Outlier Detection in Large Datasets" IEEE Transactions on Knowledge and Data Engineering.2003.
[5] Murtagh, F., "A Survey of Recent Advances in Hierarchical Clustering Algorithms", The Computer Journal, 1983.
[6] Ng, Raymond T. and Han, Jiawei, Efficient and Effective Clustering Methods for Spatial Data Mining, Proc. Of VLDB, 1994.
[7] Fayyad U., Piatetsky-Shapiro G., Smyth P.: "Knowledge Discovery and Data Mining: Towards a Unifying Framework". Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining. Portland, OR, 1996, pp. 82-88.
[8] Irad Ben-Gal, Department of Industrial Engineering, Tel-Aviv University, Ramat-Aviv, Tel-Aviv 69978, Israel., Bengal@eng.tau.ac.il."OUTLIER DETECTION".
[9] Pradeep Rai,Shubha Singh "A Survey Of Clustering Techniques", International Journal Of Computer Applications(0975-8887)Volume 7-No.12,October2010