A Performance based Analysis of Birch Algorithm over Clustering uncertain Data

T. Vijayasaratha¹, A Arul Rozario², K.K.Kavitha³

¹Assistant Professor, Department of Computer Science, Selvam Arts and Science College(Autonomous), Namakkal,
²Research Scholar, Department of Computer Science, Selvam Arts and Science College(Autonomous), Namakkal
³Vice Principal & Head, Department of Computer Science, Selvam Arts and Science College(Autonomous), Namakkal

Abstract: Data mining is the method of evaluating knowledge from totally different views and summarizing it into useful information. This information that can be used to increase profits, cuts costs, or both. BIRCH algorithm is a clustering algorithm useful for very large data sets. The Birch threshold value is the most important value of the BIRCH algorithm, and it is the most actual factor of the efficiency and accurateness results. If the threshold value parameter is reduced from its best value, then the number of tree sets lead to by BIRCH algorithm is increased exponentially. In this paper proposed to a modified BIRCH, threshold is automatically obtained through the software, by calculating the mean of a selected variable in a data set. The algorithm branches the data to the left, when the value is lesser than the threshold mean and to the right when the value is greater than the threshold mean. This process will be repeated until the complete clustering of given variable or variables.

1. Introduction

Clustering could be a data processing technique that creates useful cluster of objects that have similar characteristics using reflex technique. The clustering technique defines the classes and puts objects in every class, whereas within the classification techniques, objects are assigned in predefined classes to create the idea clearer, here will take book management in library as an example. In a library, there's a large vary of books on varied topics obtainable. The challenge is how to keep those books in a very way that readers will take many books on a specific topic while not problem. By using clustering, techniques, here will keep books that have some variations of similarities in one cluster or one shelf and label it with a purposeful name. If readers need to grab books thereon topic, they might solely have to be compelled to move to that shelf rather than looking for the complete library.

In several clustering problems we need to deal with very large data and complex data sets (gigabytes or even terabytes). The large data sets could have variant objects represented by tens, many attributes or variables, a little range of the clustering algorithmic rule have an honest efficiency in step with vast knowledge sets. The BIRCH algorithm could be a sensible robust resolution within the case of big knowledge sets. It generates a hierarchical partitioning of the data set and ensures that the most distance between two parts among a cluster is a smaller amount than the given single threshold. The efficiency and therefore the accuracy of the birth algorithm rely primarily on this threshold value, employing a single threshold cause several shortcomings within the birch algorithmic rule and during this paper we tend to try and overcome these shortcomings by modified threshold birch algorithmic rule.

2. Literature Review

Danyang Cao at el.[1] present an improved algorithm, called CFk-medoids, for mining k clustering based on k-medoids and CF-Tree. This algorithm uses a new data structure (clustering feature) to store data points of a large dataset, and make use of data points constructing a CF-Tree.

K-medoid is a classical partitioning technique of clustering that clusters the data set of n objects into k clusters known apriori. A useful tool for determining k is the silhouette. It is more strong to noise and outliers as compared to k-means. A medoid can be defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set.

In the k-medoids method, they can pick actual objects to represent the clusters, using one representative object per cluster. Each outstanding object is clustered with the representative object to which it is the most similar. The dividing method is performed based on the value of minimizing the sum of the dissimilarities between each object and its corresponding reference point.
Renata M.C.R. de Souza et al. [2] was the introduction of a $K$-medoids clustering method using standardized Euclidean distance for mixed feature-type symbolic data set. In addition, the method was compared with a clustering method for mixed feature-type symbolic data introduced. An application of the methods with different real data sets was carried out. The experimental evaluation for these data sets showed clearly the superiority of the proposed method in terms of the quality of clusters.

$K$-medoids clustering algorithm using standardized Euclidean distance for mixed feature-type symbolic data is presented. This method is an extension of the $K$-medoids clustering algorithm proposed by to mixed feature type symbolic data. To show the usefulness of this clustering method, experiments with four real symbolic data sets are considered. The aim is to show the efficiency of the $K$-medoid clustering methods for different types of symbolic data using standardized Euclidean distance.

Wenbo Xu et al.[3] discuss the problem of spatial clustering with obstacles constraints and propose a novel spatial clustering method based on Genetic Algorithms (GAs) and $K$-Medoids, called GKSCOC, which aims to cluster three-dimensional data with obstacles constraints. It can not only give care to higher local constringency speed and stronger global optimum search, but also consider the obstacles constraints and make the results of spatial clustering more practice. Its performance has compared to GAs, $K$-Medoids; and the results on real datasets show that it is better than standard GAs and $K$-Medoids. The drawback of this method is a moderately slower speed in spatial clustering.

They discuss the problem of spatial clustering with obstacles constraints and propose novel GKSCOC based on GA and $K$-Medoids. The comparison proves that they method can not only give attention to higher local constringency speed and strongerglobal optimum search, but also consider the problems that exit in the real world and make the clustering result more practice. The results of the tests on real datasets show that it is better than standard GAs and $K$-Medoids.

Aun Haider et al. [4] investigates the application of widely used $K$-Medoids based clustering algorithm on data collected through CoMon facility for the PlanetLab tested. The averaged values of various metrics in passively collected slice-centric data have been considered for clustering purposes. Various groups of slices, depicting similar resource usage patterns have been recognized in original data set.

These clusters have been signified in reduced dimensional space formed by first two principal components of original data set. In order to capture variations in pattern of resource usage by various slices at a Planet Labnode, clustering of standard deviations of various metrics have also been carried out. It has been originate that $K$-medoid based clustering can efficiently split the original data space into various sub-spaces of different resource usage behavior of slices. Thus, it can lead to better resource management and control in publicly available test beds.

Bharat Pardeshi et al. [5] proposed an improved k-medoids algorithm. In this work instead of random selection of initial k objects asteroids they have proposed a new technique for the initial representative object selection. The method is based on density of objects. They find out set of objects which are densely populated and choose medics from each of this obtained set. These k data objects selected as initial medoids are further used in clustering process. The cogency of the proposed algorithm has been proved using iris and diet structure dataset to find the natural clusters in this datasets.

The algorithm is based on the density distribution of data objects. They firstly find the dense region and then initial cluster medoids are selected from these obtained regions. In contrast to the previous algorithm based on k-medoids, the proposed algorithm can find natural clusters present in the dataset. This clustering approach is independent of the order of input dataset objects. The developed algorithm is applied in the partitioning of diet structure dataset and iris plant dataset. The number of meaningful clusters obtained for both the dataset was 3. These obtained clusters reflected the actual meaningful classes present in the dataset.

BIRCH algorithm by observing the processing of new data is found: it uses data and classes from all the other data, and predetermined distance threshold to determine whether the data belong to the sub-cluster, but the distance near a cluster center point does not necessarily belong to the cluster. It issue to produce first sub-cluster of data points to the center shift, and pre-set threshold
is fixed, cannot dynamically change the threshold to adapt to the situation, it will happen.

To solve the above problem, Du HaiZhou and Li Yong Bin at el. [6] presents a method based on dynamic changes BIRCH threshold D-BIRCH (Dynamic-BIRCH Cluster Algorithm) algorithm. Mainly through the BIRCH clustering results of the first cluster reanalysis, in order to obtain more accurate clustering results.

In the literature, there are two dominant variants providing an intra-cluster similarity: the Quality Threshold method and the BIRCH method. The BIRCH method is a good a robust solution in the case of huge data sets. It generates a hierarchical partitioning of the data set and safeguards that the extreme distance between two elements within a cluster is less than the given threshold. It can be easy verified that the size of the partitioning tree and of the query operation depends on this threshold value.

Beside the quality threshold there are two other core parameters influencing the efficiency of the algorithm, namely the branching factor of the index tree and the method of splitting. The branching factor determines how many child nodes can be generated to a given parent node. This parameter determines the depth of the tree where larger depth value causes larger cost in query operation. The method of split operation is used when a node gets full and its content is split into two new child nodes. The assignment of target child node depends on the distance values from given pivot elements. It can be seen that the selection of pivot elements and of the border of domains determines the distribution of elements and the balancing of the tree.

It follows from the previous reasoning, that the BIRCH algorithm can be considered as the primary candidate for pre-clustering. As the efficiency of pre-clustering is a key requirement, the selection of optimal parameters is a key issue in pre-clustering. The paper contains an efficiency analysis of the parameters in the base BIRCH method and it provides algorithms for optimal parameter selection.

Anand H.S. et al [7] give an delay to the Apriori algorithm, a traditional rule mining algorithm. Apriori finds its application in areas of data mining, finding suggestion between attributes and in forecast systems. Even though Apriori suits in various applications it owns various disadvantages. To increase the efficiency of the present Apriori algorithm a method for incorporating a new correlation factor (threshold) is being presented. First part of the paper provides a quick summary of basic Apriori algorithm and second half particulars the implementation of correlation threshold. Presentation of their designed algorithm is evaluated and is compared with the traditional Apriori algorithm. The evaluation shows a peak improvement in the mining result. They reduce the time complexity of the newly designed algorithm into O (n). In an application level, qualitative satisfied analysis of water was also showed to affirm the results.

3. Birch algorithm

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm [9] is an unsupervised data mining integrated hierarchical clustering algorithm. In the BIRCH tree a node is called a Clustering Feature (CF). It is a small illustration of an original cluster of one or many points. BIRCH builds on the idea that points that are close enough should always be considered as a group.

4. CF Trees Characteristics

The Birch clustering algorithm builds a dendrogram called clustering feature tree (CF tree) while scanning the large data set.

Each entry in the CF tree represents a cluster of objects and is characterized by a Three-tuple: (N, LS, SS), where N is the number of objects in the cluster and LS, SS are defined in the following.

\[ LS = \sum_{P_i \in N} \bar{P}_i \]

\[ SS = \sum_{P_i \in N} |\bar{P}_i|^2 \]

CF entry is a lot of compact. It Stores considerably but all of the information points within the sub-cluster. A CF entry has sufficient info to calculate D0-D4. Additively theorem permits us to merge sub-clusters incrementally & systematically. Given N d-dimensional data points in a cluster: \{X_i\}
where \( i = 1, 2 \ldots N, \)
\[
CF = (N, LS, SS)
\]
The above equation \( N \) is the number of data points in the cluster, \( LS \) is the linear sum of the \( N \) data points; \( SS \) is the square sum of the \( N \) data points.

If \( CF1 = (N1, LS1, SS1) \) and \( CF2 = (N2, LS2, SS2) \) are the \( CF \) entries of two disjoint sub-clusters. The \( CF \) entry of the sub-cluster formed by merging the two disjoint sub-clusters is: \( CF1 + CF2 = (N1 + N2, LS1 + LS2, SS1 + SS2) \).

5. Proposed Algorithm

In this System, the threshold is automatically obtained through the software, by calculating the mean of a selected variable in a data set. The algorithm branches the data to the left, when the value is lesser than the threshold mean and to the right when the value is greater than the threshold mean. This process will be repeated until the complete clustering of given variable or variables.

Clustering Features provide this level of abstraction.

Birch based document clustering steps

Phase 1: In the data set scan all data and build an initial in-memory \( CF \) tree.
Phase 2: Reduce into desirable length by building a smaller \( CF \) tree.
Phase 3: Global clustering
Phase 4: Cluster refining – In this phase is optional, and requires more passes over the data to refine the results.

Start1: Phase 1

1) Start with initial threshold and insert points into the tree
2) If run out of memory, increase threshold value, and rebuild a smaller tree by reinserting values from older tree and then other values
3) Good initial threshold is important but hard to figure out
4) Outlier removal – when rebuilding tree remove outlie

Phase 2:

1) Optional

2) Phase 3 sometime have minimum size which performs well, so phase 2 prepares the tree for phase 3.
3) Removes outliers, and grouping clusters.

Phase 3:

1) Problems after phase 1:
   a. Input order affects results
   b. Splitting triggered by node size
2) Phase 3:
   a. cluster all leaf nodes on the \( CF \) values according to an existing algorithm
   b. Algorithm used here: agglomerative hierarchical clustering

Phase 4:

1) Optional
2) Do additional passes over the dataset & reassign data points to the closest centroid from phase 3
3) Recalculating the centroids and redistributing the items.
4) Always converges (no matter how many time phase 4 is repeated)

The Birch Algorithm first step is to scan data from database file. While scanning the data from database the Birch algorithm consider those data points which are near to each other. Points in sparse region are treated as outlier and remove optionally. We have to form a hierarchical tree which is same like B+ Tree. If the tree is big sufficient to fit in memory, it will be spitted in two. The leaf contains the unique data points of the database.

6. Proposed work

In our proposal we are slightly modifying the Existing BIRCH algorithm with the small modification in threshold value. In the existing BIRCH algorithm, the threshold value should be defined by the User. Whereas, in our proposed system the mean value of the all data points are taken and the tree is formed like a binary search tree, where the left side nodes are smaller than the root and the right side nodes are greater than the mean value. This step is repeated until all the data
points are clustered based on the centroid mean value.

### 7. Experimental result

The experimental result for the previous and improved birch algorithms is shown in Table (6.1), and Table (6.2) and purpose to show the main characteristics of the produced CF tree created by the two algorithms. Here fix the branching factor to the value 7 which is suitable for this experiments to eliminate its effect in building the CF tree. Table (6.1) shows the Total CF-Nodes, Total CF-Entries, and Total CF-Leaf Entries of the CF tree built by the basic birch algorithm and modified birch algorithm for weather data set using initial threshold value and calculate threshold value, the value of the calculate thresholds depends on the natural of the data set samples. As shown in table (6.2) the size of the CF tree built by the improved threshold birch is about 80% less than the CF tree built by basic birch algorithm, when here calculate the dataset mean value of threshold to the size of the CF tree built by the improved birch algorithm is about 25% of the size of CF tree built by previous birch algorithm, the experiments shows that less than 1% of the samples was engrossed in the wrong sub cluster in the improved birch algorithm.

#### Table 7.1. Total Clustering Feature in Basic Birch Algorithm

<table>
<thead>
<tr>
<th>Total CF-Nodes</th>
<th>Total CF-Entries</th>
<th>Total CF-Leaf Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1836</td>
<td>1887</td>
<td>2087</td>
</tr>
</tbody>
</table>

Time milliseconds: 3808

#### Table 7.2. Total Clustering Feature in Modified Birch Algorithm

<table>
<thead>
<tr>
<th>Total CF-Nodes</th>
<th>Total CF-Entries</th>
<th>Total CF-Leaf Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1215</td>
<td>1178</td>
<td>2317</td>
</tr>
<tr>
<td>1425</td>
<td>1273</td>
<td>2546</td>
</tr>
<tr>
<td>1735</td>
<td>1596</td>
<td>2785</td>
</tr>
<tr>
<td>2538</td>
<td>2595</td>
<td>2955</td>
</tr>
<tr>
<td>2770</td>
<td>2877</td>
<td>3587</td>
</tr>
</tbody>
</table>

Time milliseconds: 3399

### Fig. 7.1 The bar chat of Basic Birch Algorithm Total Clustering Feature

### Fig. 7.2. The bar chat of Basic Birch Algorithm Total Clustering Feature
The decreasing in the CF tree size will increase the efficacy of the birch algorithm in the different phases, the calculate mean value of threshold birch algorithm ensure that the accuracy of the clustering process will not be negative affected by the decreased CF tree size. The time is decrease for the existing birch algorithm.

8. Conclusion
BIRCH algorithm delivers a clustering method for very large datasets. It makes a large clustering problem controllable by intent on densely occupied portions, and creating a dense summary. It utilizes capacities that capture the natural intimacy of data and can be stored and updated incrementally in a height-balanced tree. BIRCH algorithm can work with any given amount of memory, and the I/O difficulty is a little more than one scan of data. Experimentally, BIRCH algorithm is shown to do very well on several large datasets, and unimportantly superior to CLARANS and KMEANS in terms of quality, speed, stability and scalability overall.

9. Feature work
Proper parameter setting and further simplification are two important subjects to explore in the future. Additional issues include other experiential methods for cumulative the threshold dynamically, how to handle non-metric attributes, other threshold requirements and related insertion, transformation algorithms, and clustering confidence measurements.

References
4. “K-Medoids based Clustering of PlanetLab’s Slice-Centric Data” by Aun Haider Department of Electrical Engineering, University of Management and Technology, Lahore, Pakistan Received: 21 Mar. 2013, Revised: 21 Jul. 2013, Accepted: 24 Jul. 2013
8. “Clustering Uncertain Data With Possible Worlds”Peter Benjamin Volk, Frank Rosenthal, Martin Hahmann, Dirk Habich, Wolfgang Lehner,IEEE International Conference on Data Engineering.