International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 11 (2015) pp. 27661-27673
© Research India Publications
http://www.ripublication.com

Efficient Birch Clustering Algorithm For Categorical and Numerical Data Using Modified Co-Occurrence Method

P.Parameswari [1], Dr.J.Abdul Samath [2], P.Saranya
[1]*Department of Computer Applications,*
Kumaraguru College of Technology,Coimbatore,India.
p_param@rediffmail.com
[2]*Department of Information Technology, Sri Ramakrishna Institute of Technology,*
Coimbatore, India
abdul_samath@yahoo.com
[3]*Department of Computer Applications,*
Kumaraguru College of Technology,Coimbatore,India.
saranpalanichamy@gmail.com

Abstract

Discovery of good quality clusters is a complicated task in data mining which helps in decision making. Several clustering algorithms has been developed and works effectively either on pure numeric data or on pure categorical data. Aim of this paper is to propose a new methodology that facilitates in handling mixed data types and to fill the missing values. It integrates the new methodology with existing BIRCH algorithm which is known for stability and scalability. This methodology enhances the features and performance of BIRCH algorithm which is capable of clustering numerical datasets. The process starts with preprocessing by using a mean mode method to replace the missing values followed by a conversion methodology using a modified co-occurrence method which converts categorical attribute values to numerical attribute values, finally BIRCH algorithm is used for clustering the data..This new approach is known as EBIRCH which provides good number of clusters and greatly decreases the number of outliers and running time of clustering algorithm while compared with k-means and HAC (Hierarchical Agglomerative Clustering) algorithms.

Keywords: Data Mining, Clustering, Numerical Data, Mixed data types, BIRCH

Introduction

Nowadays computing technologies permit to record all activities related to business in enterprise data warehouses. Service providers are developing operational smartness and integration technologies to aware business activities answerable to changes in the venture that may require action [1]. Data mining (DM) helps us to pull out information from large amount of data and predict outcomes. We can describe clustering as a process of organizing similar objects; the objects in same cluster have towering similarity to objects in other clusters. Clustering methods are classified as Hierarchical and Partitioning method. Hierarchical clustering can be further divided in to two types like agglomerative clustering and divisive hierarchical clustering. Clustering used to discover natural groupings of objects [2].It is used in segmentation of large data sets into small subsets which is easily managed, and analyzed. There are many special algorithm was developed to handle data in medicine, radar scanning, manufacturing, gene data [3] etc.

Conventional algorithms used Euclidean distance measure to find the similarity between two data elements [5, 6]. This methods work well for numeric dataset not for a mix of categorical and numerical. Nowadays large amount of categorical data are collected from so many sectors like banking, health care, etc. It is a tedious process to cluster the data which has a mixed data types. As the data pre-processing is considered, it is processing step in knowledge discovery process which helps to reduce the complexity of the data and provides best results for analysis. Data pre-processing is a sturdy task. There are many methods used for performing pre-processing task .It includes many steps which removes and organizes data for more efficient access. Incomplete data is a challenge for achieving a good data mining process, The methods like statistical and machine learning are not designed to handle incomplete data ,if we replace all missing value with NULL and remove all instances then proper result cannot be obtained. Sufficient preprocessing techniques helps to improve the quality of raw data and reduce the risk of a DM process failure .So it is very essential to have a good preprocessing technique. The data clustering is a technique in data mining applications for discovering underlying data.

In this paper a new methodology for handling mixed dataset has been proposed and integrated with existing BIRCH algorithm which is known as EBIRCH and the results are compared with k-means and HAC algorithms and proved that EBIRCH is efficient than others. The rest of this paper is structured as follows. Section 2 gives a brief note on related works .Section 3 describes the preprocessing, conversion and clustering methodologies, In section 5 experimental results are discussed followed by research conclusion.

Related Works

Similarity Based Agglomerative Clustering (SBAC) algorithm was proposed [8] based on Goodall similarity measure [9]. The items in categorical attributes are processed to construct the similarity or relationships among them based on the ideas of co-occurrence; then the categorical attributes are converted into numeric attributes, based on these erect relationships. An initiative of co-occurrence was used for converting categorical attributes into numeric attributes [2] it aims to divide the data set into several groups which have the high degree of similarity. All hierarchical clustering algorithms are good for dealing very large databases but BIRCH [1] produces the best quality clustering with available resources. It finds the good clusters with single scan of the data and improves the quality further with additional scans. It is the first algorithm to handle the noise efficiently another clustering algorithm CURE [10] is also capable of handling numeric data sets only. The well-known algorithm is the k-means [11] which partitions the data set into k subsets and all points in a subset are closest to the same centre. K-means is an efficient algorithm in processing large data sets. CHAMELEON is a good algorithm based on graph-partitioning algorithm [12] cluster data items are divided into many sub clusters which provide genuine clusters.

Proposed Methodology

The proposed methodology is an enhancement to BIRCH algorithm which consists of three phases like preprocessing, conversion and clustering using BIRCH algorithm. We call this three phases together as enhanced BIRCH.

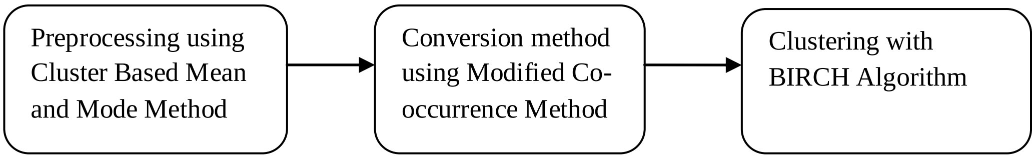

Preprocessing using Cluster Based Mean and Mode Method
Conversion method using Modified Co-occurrence Method
Clustering with BIRCH Algorithm

Figure 1: Overview of Enhanced BIRCH Algorithm

Preprocessing

Handling missing values is one of the data preprocessing techniques in data mining. So many existing methods are available for handling missing values like , ignoring the tuples, manual replacement, use of global constant etc., In this paper we used a cluster based mean and mode method for handling the missing values. This method clusters the instances based on the class labels , used mean value for numeric attributes replacement and mode method for categorical attributes replacement.

Table 1: Original Dataset for Preprocessing

Outlook	**Temperature**	**Humidity**	**Windy**	**Play**
Rainy	70	99	False	Yes
Rainy	68	80	False	Yes
Rainy	65	70	True	No
Overcast	57	65	True	Yes
Sunny	72	95	False	No
Sunny	69	70	False	Yes
Rainy	75	63	False	Yes

Process Steps

1. Read the dataset.
2. Cluster the dataset based on the class label and name it as cluster C1 and C2, if there a need add C3.
3. If C1 is a numerical attribute find the mean value of that attribute values and fill the missing values.
4. If C1 is a categorical attribute find the mode (the item which occurs more number of times) value of that attribute and replace the missing values.
5. Repeat 3 and 4 for Cluster C2 and Cluster C3.
6. Merge the cluster using bottom-up approach.

Table 2: Dataset with missing values

Outlook	**Temperature**	**Humidity**	**Windy**	**Play**
Rainy	70	99	False	Yes
Rainy	68	80	?	Yes
?	65	70	True	No
Overcast	?	65	True	Yes
Sunny	72	95	False	No
Sunny	69	70	False	Yes
Rainy	75	?	?	Yes

Table 3: Dataset after preprocessing (Clustering based mean and mode method)

Outlook	**Temperature**	**Humidity**	**Windy**	**Play**
Rainy	70	99	False	Yes
Rainy	68	80	**False**	Yes
Rainy	65	70	True	No
Overcast	**56.4**	65	True	Yes
Sunny	72	95	False	No
Sunny	69	70	False	Yes
Rainy	75	**62.8**	**False**	Yes

Process description:

Read the dataset which consists of both categorical and numerical values. Cluster the instances based on class labels into C1, C2 and if there is a need, cluster it to C3. Each cluster contains both categorical and numerical data. The mode value is calculated for numerical data types and mode value mode values are calculated for categorical data types and replacement for missing values takes place. Calculation of mode values (M) is done by finding the item which occurs more often in the attribute.

Conversion Using Modified Co-Occurrence Method

Conversion techniques help to have good clusters, many algorithms work for either on categorical data or numerical. To handle a mixed dataset a modified co-occurrence method is proposed that helps to convert categorical values to numerical values. BIRCH can handle only numerical data but it has many important features, to utilize that this conversion methodology has been proposed to make BIRCH to perform well on mixed data types. All categorical attribute values are converted to numeric according to the similarity. In this paper the items in categorical attributes are processed to construct the similarity or relationships among them based on the ideas of co-occurrence; then all categorical attributes can be converted into numeric attributes based on these constructed relationships.

Algorithm

1. Normalize the numeric attribute.
2. Find the attribute with the high number of items to be base attribute and items are base items.
3. Count the frequency of co-occurrence between the items in the base attribute and the categorical attribute (CA1). Divide it by number of base items and store it Matrix T.
4. Repeat the step 3 until Categorical item equals to 0. Take the next categorical attribute compare it with the base attribute and Repeat step 3.
5. Find the numeric attribute which has the minimum mean value and assign the values to the categorical values in the base item and find the mean value of each base item.
6. Calculate the information stored in matrix T and find the numeric values for each categorical item.

Table 3: Weather dataset

Outlook	**Temperature**	**Humidity**	**Windy**
Rainy	70	99	False
Rainy	68	80	False
Rainy	65	70	True
Overcast	57	65	True
Sunny	72	95	False
Sunny	69	70	False
Rainy	75	63	False

From the above Table 3, the base attribute is ‘**outlook**’ because the number of items in outlook is three like rainy, overcast and sunny and the base items are rainy, overcast and sunny. The frequency of co-occurrence between the base item and

Categorical items are obtained and the items are divided by number of base items (i.e. 3) are represented in a matrix T.

Matrix T:

True False True False

Rainy 1/3 3/3 Rainy 0.333 1

Overcast 1/3 0 = Overcast 0.333 0

Sunny 0 2/3 Sunny 0 0.666

Conversion of base item: Calculate the mean value for all numerical attributes and find out the one which has the minimum mean value. Temperature attribute has the mean value 68.428 and Humidity attribute has the mean value 78.285.Temperature attribute has the minimum mean value ,so select the temperature attribute and assign its value to the items in the base attributes.

Example : Rainy (70+68+65+75)/4 = 69.5 (Rainy occurs 4 times). Similarly, for Overcast the value assigned is 60 and for sunny the value assigned is (72+69)/2 = 70.5.

Table 4: Data set after the base item is converted

Outlook	**Temperature**	**Humidity**	**Windy**
69.5	70	99	False
69.5	68	80	False
69.5	65	70	True
60	60	65	True
70.5	72	95	False
70.5	69	70	False
69.5	75	69	False

Conversion of other Categorical attributes except base attribute: Since all the base item are assigned to a numeric values, all other categorical items are assigned to numeric values by as follows,

$$F(x)= \sum_{i=1}^{d} \mathrm{ai} * \mathrm{vi} \text{ \_\_\_\_\_\_\_\_ Eq} \qquad (1)$$

a_i - The similarity between item *x* and i th base item.

v_i - The quantified value of i^{th} base item.

d - The number of base item.

The item False in the above table will be assigned to the value

F (False) = 1*69.5+0*60+0.666*70.5 = 116.45

Similarly for all categorical items,

F (True) = 0.333*69.5+0.333*60+0*70.5= 43.124.

Table 6: Numerical dataset after Conversion

Outlook	**Temperature**	**Humidity**	**Windy**
69.5	70	99	116.45
69.5	68	80	116.45
69.5	65	70	43.124
60	60	65	43.124
70.5	72	95	116.45
70.5	69	70	116.45
69.5	75	69	116.45

Clustering Using BIRCH Algorithm

BIRCH is designed for clustering a large amount of numerical data and describes a hierarchical clustering method, which uses a new data structure similar to a B-tree, called a CF-tree, to store a small amount of information about each cluster in order to dynamically update clusters in a linear scan of the dataset. It introduces like cluster feature and cluster feature tree (CF Tree) which is used to summarize the cluster in the way in which it is representations. The data space in BIRCH is not equally occupied; hence not every data point is equally important for clustering purposes. It utilizes the whole part of the memory which is available to derive the best possible sub clusters. An efficient data clustering method based on new in-memory data structure called CF-tree. This method could produce high quality dataset with far less instances that sufficiently represent all of the instances in the original dataset

Process Steps:

1. Find the cluster feature (CF) for the attribute.
   CF= (N, LS, SS)
   N-Data points in the cluster (Number)
   LS-Linear Sum of the N data points
   SS-Square Sum of data points
2. Calculate radius , centroid, centroid Euclidian distance , centroid Manhattan distance, average inter-cluster distances, average intracluster distance and variance of cluster increases by distance.
3. Repeat step 1 and step2 until attribute equal to NULL.
4. Assume the branching factor B and the threshold value T.
5. Construct Cluster Feature Tree (CF Tree).

Build the CF tree with the above information calculated and it consists of two parameters. They are (i) Threshold (T) and (ii) Branching Factor (B) where,

B – Maximum number of children per non leaf node.

T – Maximum diameter of sub clusters stored at the leaf nodes of the tree.

Each non-leaf node contains at most B entries of the form [CF_i, $child_i$], where $child_i$ is a pointer to its i^{th} child node and CF_i is the CF of the subcluster represented by this child. So, a non-leaf node represents a cluster made up of all the sub clusters represented by its entries. A leaf node contains at most *L* entries, each of them of the form [CF_i], where i = 1, 2 . . . L. It also has two pointers, *prev* and *next*, which are used to chain all leaf nodes together for efficient scans. A leaf node also represents a cluster made up of all the sub clusters represented by its entries. But all entries in a leaf node must satisfy a *threshold*, with respect to a threshold value T: the diameter has to be less than T [1]. The leaf(s) contains actual clusters and the size of any cluster in a leaf is not larger than T.

Experimental Results

Insurance set was considered for our analysis and this dataset contains 5,000 instances and 18 attributes and clustering takes place based on the class label. It contains some missing values also. The attributes like categorical and nominal are available in this dataset.

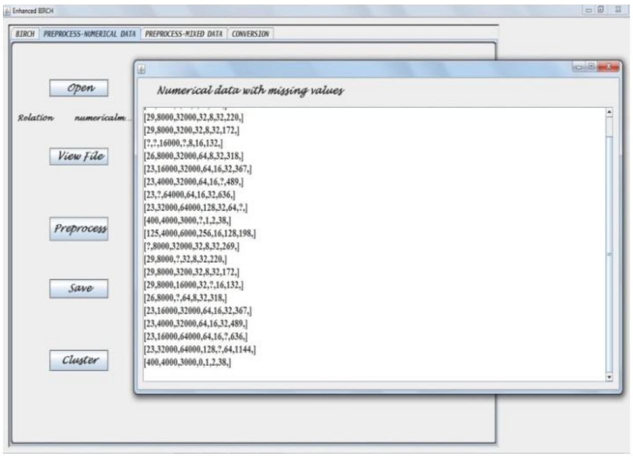
Comparison of BIRCH Algorithm with K-Means and HAC Algorithm:

Before Preprocessing and Conversion

BIRCH algorithm is compared with k-means algorithm and HAC (Hierarchical Agglomerative Clustering) with different number of records for both numbers of clusters and outliers. Only numerical attributes are considered for analysis.

Table 7: Clusters Obtained Before Conversion and Preprocessing

Number of Records	BIRCH Algorithm	k-means Algorithm	HAC Algorithm
1000	4	3	2
2000	6	3	2
3000	7	5	3
4000	8	6	7
5000	10	6	9



Numerical data with missing values

```
[29,8000,32000,32,8,32,220,]
[29,8000,3200,32,8,32,172,]
[?,?,16000,?,8,16,132,]
[26,8000,32000,64,8,32,318,]
[23,16000,32000,64,16,32,367,]
[23,4000,32000,64,16,?,489,]
[23,?,64000,64,16,32,636,]
[23,32000,64000,128,32,64,?,]
[400,4000,3000,?,1,2,38,]
[125,4000,6000,256,16,128,198,]
[?,8000,32000,32,8,32,269,]
[29,8000,?,32,8,32,220,]
[29,8000,3200,32,8,32,172,]
[29,8000,16000,32,?,16,132,]
[26,8000,?,64,8,32,318,]
[23,16000,32000,64,16,32,367,]
[23,4000,32000,64,16,32,489,]
[23,16000,64000,64,16,?,636,]
[23,32000,64000,128,?,64,1144,]
[400,4000,3000,0,1,2,38,]
```

Figure 2: Numerical Data set with Missing Values

Table 8: Outliers Detected Before Conversion and Preprocessing

Number of Records	BIRCH Algorithm	k-means Algorithm	HAC Algorithm
1000	0	10	250
2000	0	30	300
3000	30	70	450
4000	70	100	570
5000	100	150	700

After preprocessing and conversion

After preprocessing and conversion the Enhanced BIRCH algorithm is compared with k-means algorithm and HAC (Hierarchical Agglomerative Clustering) with different number of records and for number of clusters, outliers and execution time. The dataset which was considered for this analysis consist of numerical and categorical values(after converting to numerical values).

Table 9: Clusters Obtained After Conversion and Preprocessing

Number of Records	Enhanced BIRCH Algorithm	k-means Algorithm	HAC Algorithm
1000	2	3	2
2000	3	3	2
3000	5	5	3
4000	6	6	7
5000	6	6	9

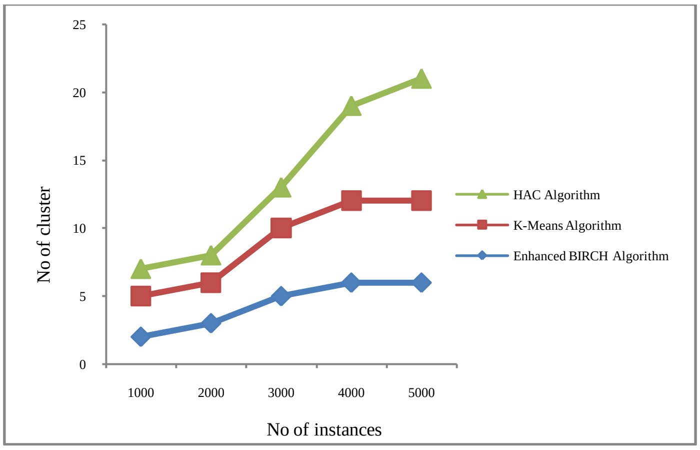


Figure 3: EBIRCH Vs k-means and HAC in terms of Clusters

The Enhanced BIRCH algorithm is compared with HAC algorithm, k-means algorithm for number of clusters. In figure 3 X-axis represents number of clusters and Y-axis represents number of instances.

Table 10: Performance of Clustering Algorithms in terms of Outliers

Number of Records	Enhanced BIRCH Algorithm	k-means Algorithm	HAC Algorithm
1000	0	10	250
2000	0	30	300
3000	0	70	450
4000	0	100	570
5000	30	150	700

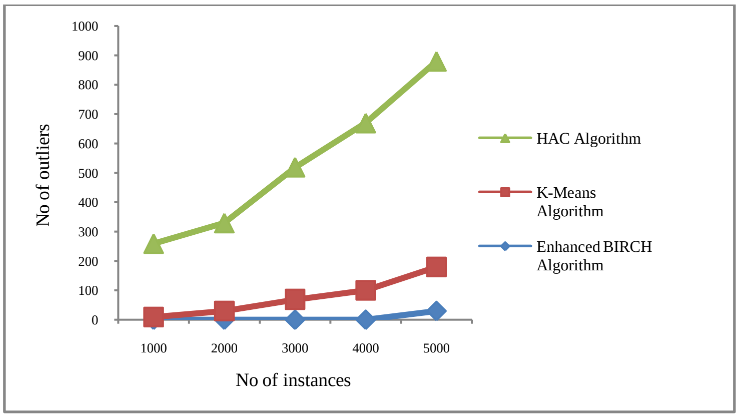


Figure 4: EBIRCH Vs k-means and HAC in terms of Outliers

Table 11: Performance of Clustering Algorithms in terms of Execution Time

Number of Records	Enhanced BIRCH Algorithm (ms)	k-means Algorithm (ms)	HAC Algorithm (ms)
1000	310	210	3560
2000	432	390	11360
3000	441	570	38140
4000	452	890	58150
5000	460	1000	72420

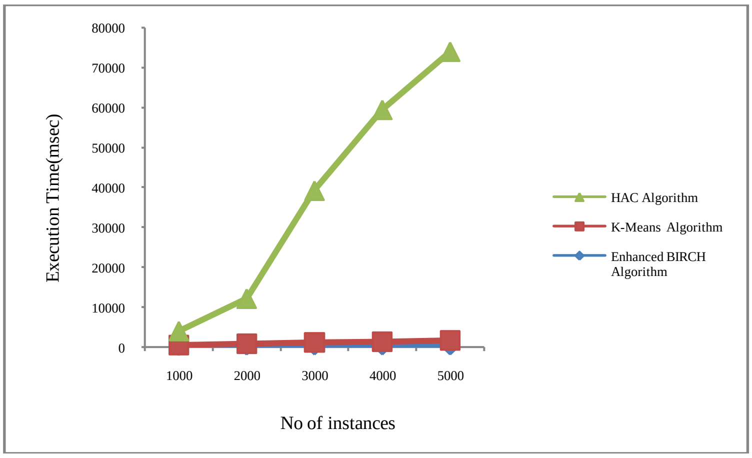
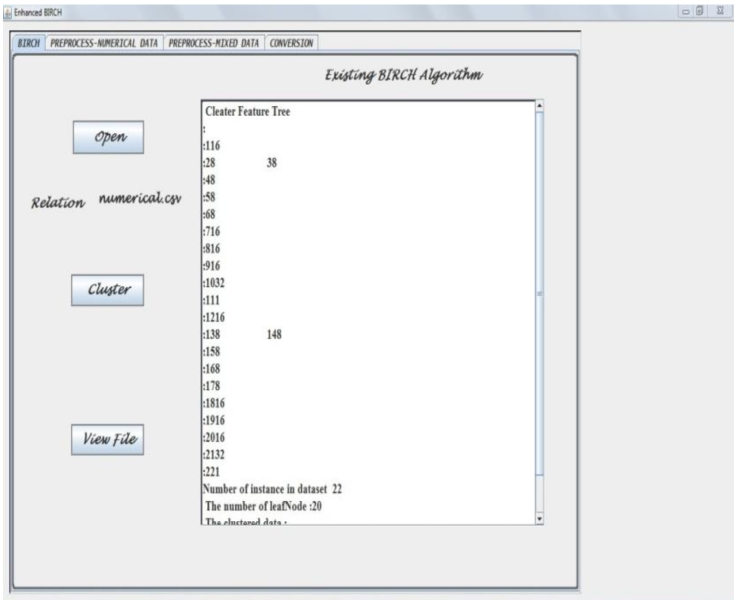


Figure 5: EBIRCH Vs k-means and HAC in terms of Execution Time



```
Cleater Feature Tree
:
:116
:28          38
:48
:58
:68
:716
:816
:916
:1032
:111
:1216
:138         148
:158
:168
:178
:1816
:1916
:2016
:2132
:221
Number of instance in dataset  22
 The number of leafNode :20
 The clustered data :
```

Figure 6: Construction of CF tree

Conclusions

The main advantage of BIRCH algorithm in data mining is its efficiency in clustering large data sets but it is limited to numeric values. The modified co-occurrence conversion method which presented in this paper has rectified this limitation and preserves its efficiency. This enhancement helps to cluster mixed data types. Another limitation of clustering algorithm is handling missing values well, which was done by mean mode method without compromising on cluster quality. The conversion and replacement of missing values is a combined approach that allows to cluster numerical values directly with the help of BIRCH algorithm. Scalability is an important requirement of clustering algorithms which can be handled by BIRCH efficiently .The results demonstrated shows that the Enhanced BIRCH algorithm is indeed scalable to large and complex data sets in terms of outliers and the number of clusters. In fact the results of EBIRCH is closer to *k*-means algorithm in cluster numbers but the outliers are more in number in both k means and HAC than that of EBIRCH .In future there is a plan to develop and implement Enhanced BIRCH to cluster data sets with millions of instances.

References

[1] T. Zhang, R. Ramakrishnan and M. Livny, 1996. BIRCH: An efficient clustering method for very large databases, Proceeding of the ACM SIGMOD Workshop on Data Mining and Knowledge Discovery, pp:103-114.

[2] Ming-Yi Shih, Jar-Wen Jheng and Lien-Fu Lai,2010. A Two-Step Method for Clustering Mixed Categorical and Numeric Data**,** Tamkang Journal of Science and Engineering, Vol. 13, No. 1, pp: 11-19

[3] Dr. S. Vijayarani , Ms. P. Jothi ,2013. An Efficient Clustering Algorithm for Outlier Detection in Data Streams , International Journal of Advanced Research in Computer and Communication Engineering Vol. 2.

[4] Tian Zhang, Raghu Ramakrishnan, Miron Livny, BIRCH: A New Data Clustering Algorithm and Its Applications, Small Journal Name, 1-40 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

[5] Tian Zhang, Raghu Ramakrishnan and Miron Livny, 1997.BIRCH: A New Data Algorithm and its Applications, Data Mining and Knowledge Discovery, 1, pp:141-182

[6] D.Pramodh Krishna, A.Senguttuvan and T.Swarna Latha, 2012. Clustering on Large Numeric Data Sets Using Hierarchical Approach: Birch, Global Journal of Computer Science and Technology.

[7] B. Yedukondalu and G. Srinivasa Rao, 2012.Efficiently Improving Associations Among Items & Weakness in Cluster –TMCM Algorithm, IJCST Vol. 3, Issue 3,pp:1130-1132.

[8] C. Li, G. Biswas, 2002. Unsupervised learning with mixed numeric and nominal data, IEEE Transactions on Knowledge and Data Engineering 14 (4) ,pp: 673–690.

[9] D.W. Goodall, 1996.A new similarity index based on probability, Biometric 22, 882–907.

[10] S. Guha, R. Rastogi, K. Shim, 1998.CURE: An efficient clustering algorithm for clustering large databases, in: Proceedings of the Symposium on Management of Data (SIGMOD),

[11] Jain A.K, M. N. Murty and P. J.Flynn,1999 Data Clustering: A Review. ACM Comput. Surv. Vol 31(3),pp:264- 323

[12] Karypis G., Han E. H. and Kumar V. (1999), CHAMELEON: A hierarchical clustering algorithm using dynamic modeling, Computer 32(8),pp: 68-75.

[13] Li.C, G. Biswas, 2002.Unsupervised learning with mixed numeric and nominal data, IEEE Transactions on Knowledge and Data Engineering14 (4) pp:673–690.

[14] Goodall D.W,1996. A new similarity index based on probability, Biometric 22 pp:882–907.

[15] www.cs.waikato.ac.nz/ml/weka.

[16] He.Z, X. Xu, S. Deng, 2001, Squeezer: An efficient algorithm for clustering categorical data, Journal of Computer Science and Technology 17 (5) pp: 611–624.

[17] Anil K. Jain Data clustering: 50 years beyond K-means Pattern Recognition Letters 31(2010) 651–666.

[18] Yiu-ming Cheung a,b,n, Hong Jia , 2013.Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number Pattern Recognition ,pp:2228–2238

[19] Acuna, E. & Rodriguez, C., 2004 .The Treatment of Missing Values and its Effect on Classifier Accuracy, In: Classification, Clustering, and Data Mining Applications, pp: 639-647.

[20] Amir Ahamad, Lipika Dey, 2007.A k-mean clustering algorithm for mixed numeric and categorical data, Science Direct, Data & Knowledge Engineering pp: 503–527.

[21] UCI Machine learning Repository-Datasets.

[22] Jiawei Han, Micheline Kamber, Jian Pei, 2011. Data Mining: Concepts and Techniques. Morgan Kaufmann Series