A Cluster Based MARDL Algorithm for Drifting Categorical Data

A Siddhartha Reddy[1], C V V N Varun[1], AnushaAre[2], P.V.V Prasad[2], K. Ruth Ramya[2]

[1](M.Tech student, Department of CSE, K L University, Vaddeswaram, A.P.-522502, India)
[2](Asst.Professor, Department of CSE, K L University, Vaddeswaram, A.P.-522502, India)

ABSTRACT:
Clustering is an important problem in data mining. Most of the earlier work on clustering focused on numeric attributes which have a natural ordering on their attribute values. Recently, clustering data with categorical attributes, whose attribute values do not have a natural ordering, has received some attention. However, previous algorithms do not give a formal description of the clusters they discover and some of them assume that the user post-processes the output of the algorithm to identify the final clusters. Sampling has been recognized as an important technique to improve the efficiency of clustering. However, with sampling applied those points that are not sampled will not have their labels after the normal process the problem of how to allocate those unlabeled data points into proper clusters remains as a challenging issue in the categorical domain. A mechanism named Maximal Resemblance Data Labeling for to kept the every unlabeled data point in to appropriate cluster , the MARDL will exhibits high execution efficiency and it preserves clustering characteristics that is high intra-cluster similarity and low inter-cluster similarity.

Keywords: Data mining, Data labeling, categorical Clustering.

1. INTRODUCTION

Clustering in the computer science world is the classification of data or object into different groups. It can also be referred to as partitioning of a data set into different subsets. Data cluster are created to meet specific requirements that cannot created using any of the categorical levels. One can combine data subjects as a temporary group to get a data cluster. Data clustering is an important technique for exploratory data analysis and has been the focus of substantial research in several domains for decades [8], [9]. The problem of clustering is defined as follows: Given a set of data objects, the problem of clustering is to partition data objects into groups in such a way that objects in the same group are similar while objects in different groups are dissimilar according to the predefined similarity measurement. Therefore, clustering analysis can help us to gain insight into the distribution of data. However, a difficult problem with learning in many real world domains is that the concept of interest may depend on some been explored in the previous works [1,2], hidden context, not given explicitly in the form of predictive features. In other words, the concepts that we try to learn from those data drift with time [7, 11, and 12]. For example, the buying preferences of customers may change with time, depending on the current day of the week, availability of alternatives, discounting rate, etc. As the concepts behind the data evolve with time, the underlying clusters may also change considerably with time [1]. Performing clustering on the entire time-evolving data not only decreases the quality of clusters but also disregards the expectations of users, which usually require recent clustering results. The problem of clustering time-evolving data in the numerical domain has [3, 4, 5, 6, 10, and 13]. However, this problem has not been widely discussed in the categorical domain with the exception of for Web log transactions. Actually, categorical attributes also prevalently exist in real data with drifting concepts. For example, buying records of customers, Web logs that record the browsing history of users, or Web documents often evolve with time. Previous works on clustering categorical data focus on doing clustering on the entire data set and do not take the drifting concepts into consideration. Therefore, the problem of clustering time evolving data in the categorical domain remains a challenging issue. As a result, a framework for performing clustering on the categorical time-evolving data is proposed in this paper. Instead of designing a specific clustering algorithm, we propose a generalized clustering

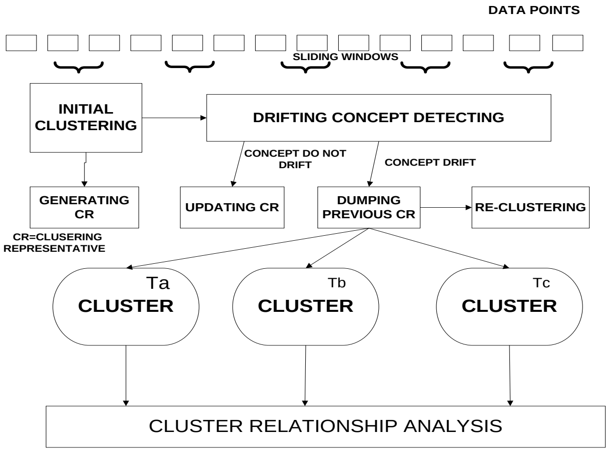


Fig. 1. The framework of performing clustering on the categorical time-evolving data.

framework that utilizes existing clustering algorithms and detects if there is a drifting concept or not in the incoming data.

The fig shows our entire framework of performing clustering on the categorical time-evolving data. In order to detect the drifting concepts, the sliding window technique is adopted. Sliding conveniently eliminate the outdated records, and the sliding windows technique is utilized in several previous works on clustering time-evolving data in the numerical domain [1,3, 4,12]. Therefore, based on the sliding window technique, we can test the latest data points in the current window if the characteristics of clusters are similar to the last clustering result or not.

1.1. Definition1 (*Node*)

A *node*, dt , is defined as *attribute name + attribute value* .The term *node* which is defined to represent attribute value in this paper avoids the ambiguity which might be caused by identical attribute values. If there are two different attributes with the same attribute value, e.g., the age is in the range *50~59* and the weight is in the range *50~59*, the attribute value *50~59* is confusing when we separate the attribute value from the attribute name. *Nodes [age=50~ 59]* and [w*eight=50~59]* avoid this ambiguity. Note that if the attribute name and the attribute value are both the same in the nodes d1 and d2, d1 and d2 are said to be equal. For example, in Figure 3 cluster c1, *[A1=a]* and *[A2=m]* are nodes.

1.2. Node Importance Representative (NIR)

We next describe the novel categorical cluster representative which is named *NIR* (standing for **N**ode **I**mportance **R**epresentative). The basic idea behind NIR is to represent a cluster as the distribution of the nodes, which is defined in Definition1.Moreover, in order to measure the representability of each node in a cluster, the importance of node is evaluated based on following two concepts: (1) The node is important in the cluster when the frequency of the node is high in this cluster. (2) The node is important in the cluster if the node appears prevalently in this cluster rather than in other clusters. The first concept characterizes the importance of the node in the cluster. The rationale for us to adopt the second concept to measure the importance of the node can be explained, where an attribute distribution in the two clusters is given. The node *b* is the most frequent node in the cluster 1. However, in the all data points which contain node *b*, there are only around 40% data points which belong to the cluster 1. In contrast, although the node *c* is less frequent than node *b* in the cluster 1, node c mostly occurs in the cluster 1. Only considering the first concept will cause the importance of node to be high simply because the node is frequent in the database. However, the representability of the node in this cluster is likely to be overestimated because the other clusters also contain this node with high frequency. Consequently, both the two concepts should be employed to evaluate the importance of the node.

Note that the good cluster criteria is high intra-cluster similarity, where the sum of distances between objects in the same cluster is minimized, and low inter-cluster similarity, where the distances between different clusters are maximized. Suppose that there is a node with high frequency in the cluster. This means that most of the data points in the cluster contain this node, and the intra-cluster similarity will be high. Hence, the first concept considers the distribution of the node in the cluster, which can be deemed as the intra-cluster similarity. In addition, suppose that a node occurs in one cluster and does not appear in other clusters. This means that most of the data points which contain this node only occur in this cluster. The distances between different clusters will be large. Hence, the second concept considers the distribution of the node between clusters, which can be deemed as the inter-cluster similarity. Therefore, NIR represents cluster by nodes and the importance of nodes ,which considers both the intra-cluster similarity and the inter-cluster similarity.

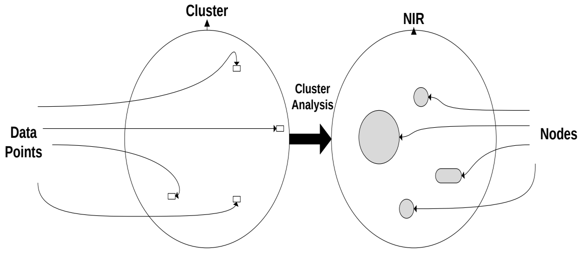


Fig. 2. The concept of NIR to represent a cluster.

As shown in Figure 2, the cluster is represented by NIR.The ellipses in the right side of Figure4 illustrate the nodes in the cluster, and the importance of the nodes is presented by the size of each ellipse. After the process of cluster analysis, a cluster with data points is represented by NIR. To achieve this, the theory of NIR technique is presented below.

Based on the foregoing, cluster ci can be represented by nodes. Each data point in the cluster ci is first decomposed into nodes, and then, the frequency of nodes in the cluster is calculated. The node decomposed from the data point may be equal to the node decomposed from the previous data points. In such cases, the frequency of this node is increased by one. After all the data points are decomposed into nodes in the cluster c_i, suppose that c_i contains t nodes, and each node dk which occurs in the cluster ci is abbreviated by d_{ik} , and, the frequency of node d_{ik} is $|d_{ik}$ |. Then, the node importance and NIR are defined as follows.

1.3. Definition 2 (node importance and NIR):

The node importance of the node (d_{ik}) is calculated as the following equations:

$$W(c_i, d_{ik}) = f(d_{ik}) \frac{|d_{ik}|}{\sum_{x=1}^{t} |d_{ix}|} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

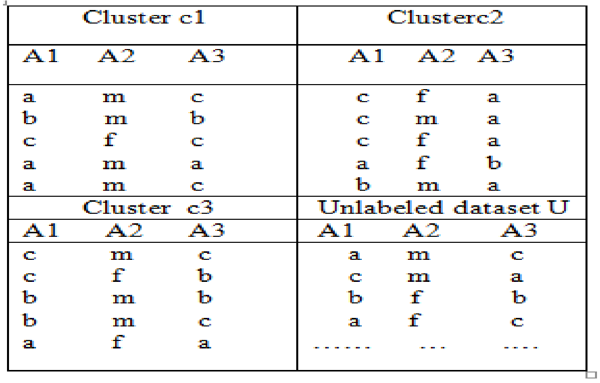
$$f(d_{ik}) = 1 - \frac{-1}{\log n} \times \sum_{y=1}^{n} p(d_{yk}) \log(p(d_{yk})),$$

$$\text{Where } p(d_{yk}) = \frac{|d_{yk}|}{\sum_{z=1}^{n} |d_{zk}|} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

The NIR of cluster c_i be represented as a table of the pairs (d_{ik}, w(c_i, d_{ik})) for the all nodes, i.e., $d_{i1} d_{i2}$..., d_{it}, in the cluster c_i.w(c_i, d_{ik}) represents the importance of node d_{ik} in cluster c_i with two factors, the probability of d_{ik} in c_i and the probability of d_{ik} in c_i and the weighting function f (d_{ik}). Based on the concepts of the importance of a node, the probability of d_{ik} in c_i is calculated to compute the frequency of $\mathrm{d_{ik}}$ in the cluster$\mathrm{c_i}$, and the weighting function is designed to measure the distribution of the node between clusters based on the information theorem. Entropy is the measurement of information and uncertainty on a random variable. Formally, if X is a random variable, S(X) is the set of values which X can take, and p(x) is the probability function of X, the entropy E(X) is defined as shown in Eq. (3).

E(X) = $-\sum_{x\in s(x)} P(x)\log(p(x))$…………..(3)

TABLE 1
An example dataset with three clusters and several unlabeled data points.



Cluster c1			Clusterc2		
A1	A2	A3	A1	A2	A3
a	m	c	c	f	a
b	m	b	c	m	a
c	f	c	c	f	a
a	m	a	a	f	b
a	m	c	b	m	a
Cluster c3			Unlabeled dataset U		
A1	A2	A3	A1	A2	A3
c	m	c	a	m	c
c	f	b	c	m	a
b	m	b	b	f	b
b	m	c	a	f	c
a	f	a	……	…	….

Explanation: Consider the data set in Table1. Cluster c_1 contains eight nodes ([A_1=a], [A_1=b], [A_1=c],[A_2==m], [A_2==f], etc.).

The node *[A_1=a]* occurs 3 times (|d1,[A_1=a]| = 3) in c_1 , once in c_2 , and once in c_3 .

The weight of the node *[A_1=a]*, f (d[A_1=a]) = $1-\frac{-1}{log3}(\frac{3}{5}\log\frac{3}{5}+\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5})$ = 0.135.

The importance of node *[A_1=a]* in cluster c_1 is: w(c_1 , [A_1 = a]) =0.135$*\frac{3}{15}$ = 0.027. Note that in the cluster c1, node *[A3=c]* also occurs three times. However, this node does not occur inc_2 but occurs twice inc_3.

Therefore, in cluster c_1 , the node *[A_3=c]* is more significant than node *[A_1=a]*.

Corresponding to the node importance, w (c_1, [A_3 = c]) = f (d [A_3=c]) $*\frac{3}{5}$ = 0.387 $*\frac{3}{15}$ = 0.077 > w(c1, [A_1 =a]) = 0.027.

TABLE 2
The NIR table of cluster c1, c2, and c3 in table1

Cluster c_1	Cluster c_2	Cluster c_3
d_{1j} W(d_{1j})	d_{2j} W(d_{2j})	d_{3j} W(d_{3j})
$[A_{1=a}]$ 0.027	$[A_{1=a}]$ 0.009	$[A_{1=a}]$ 0.009
$[A_{1=b}]$ 0.004	$[A_{1=b}]$ 0.004	$[A_{1=b}]$ 0.007
$[A_{1=c}]$ 0.005	$[A_{1=c}]$ 0.016	$[A_{1=c}]$ 0.011
$[A_{2=m}]$ 0.009	$[A_{2=m}]$0.005	$[A_{2=m}]$ 0.007
$[A_{2=f}]$ 0.005	$[A_{2=f}]$ 0.016	$[A_{2=f}]$ 0.011
$[A_{3=a}]$ 0.014	$[A_{3=a}]$0.056	$[A_{3=a}]$ 0.014
$[A_{3=b}]$ 0.004	$[A_{3=b}]$ 0.004	$[A_{3=b}]$ 0.007
$[A_{3=c}]$ 0.077		$[A_{3=c}]$ 0.052

Although these two nodes both occur three times in cluster c1, node *[A_3=c]* provides more information on cluster c1 than node *[A_1=a]* Finally, the *NIR* of cluster $\mathrm{c_i}$ can be represented as a table of the pairs ($\mathrm{d_{ik}}$, w($\mathrm{c_i}$, $\mathrm{d_{ik}}$)) for the all nodes in the cluster $\mathrm{c_i}$. The table 2 shows the *NIR* of the three clusters in Table1.

2. RELATED WORK

A survey on clustering techniques can be found in [14] Here, we focus on reviewing the techniques of cluster representative and data labeling on the categorical data, which are most related to our work.

Cluster representative is used to summarize and characterize the clustering result [11]. Since, in the categorical domain, the cluster representative is not well discussed, we review several categorical clustering algorithms and explain the sprite of cluster representative in each algorithm.

In k-modes [15], a cluster is represented by "mode", which is composed by the most frequent attribute value in each attribute domain in this cluster. Suppose that there are q attributes in the dataset. Only q attribute values, each of which is the most frequent attribute value in each attribute, will be selected to represent the cluster. Although this cluster representative is simple, only use one attribute value in each attribute domain to represent a cluster is questionable. For example, suppose that there is a cluster which contains 51% male and 49% female in attribute gender. Only using male to represent this cluster will lose the information from female, which is almost a half in this cluster.

In algorithm ROCK [16], clusters are represented by several representative points. This representative does not provide a summary of cluster, and thus cannot be efficiently used for the post-processing. For example, in the data labeling, the similarity between unclustered data points and clusters is needed to be measured. It is time consuming to measure the similarity between unclustered data points and each representative point, especially when a large amount of representative points is needed for the better representability.

In algorithm CACTUS [17], clusters are represented by the attribute values. The basic idea behind CACTUS is to calculate the co-occurrence for attribute-value pairs. Then, the cluster is composed of the attribute values with high co-occurrence.

However, this representative does not measure the importance of the attribute values. A cluster is represented only by several attribute values and each attribute value has the equally representability in the cluster. In this paper, we present NIR, which is based on the idea of representing the clusters by the importance of the attribute values because the summarization and characteristic information of a cluster can be obtained by the attribute values. Utilizing the summarization and characteristic information to execute data labeling is more efficient than utilizing the representative points.

Furthermore, data labeling is used to allocate an unlabeled data point into the corresponding appropriate cluster. The technique of data labeling has been studied in CURE [18]. However, CURE is a special numerical clustering algorithm to find non-spherical clusters. A specific data labeling algorithm is defined to assign each unlabeled data point into the cluster which contains the representative point closest to the unlabeled data point. In addition, ROCK [16], a categorical clustering algorithm, also utilizes data labeling to speed up the entire clustering procedure. The data labeling method in ROCK is independent of the proposed clustering algorithm, and is performed as follows. First, a fraction of points is obtained to represent each cluster. Then, each unlabeled data point is assigned to the cluster such that the data point contains the maximum neighbors in the fraction of points from the cluster. Two data points are said to be the neighbor of each other if the Jaccard-coefficient [19] is larger than or equal to the user defined threshold θ. However, the threshold θ in ROCK data labeling is difficult to be determined by users. Moreover, it is time consuming to compute the neighbor relationship between an unclustered data point and all representative points.

2.1. Maximal Resemblance Data Labeling

In this paper a mechanism, named **MA**ximal **R**esemblance **D**ata **L**abeling (abbreviated as MARDL), to allocate each categorical unclustered data point into the corresponding proper cluster. The allocating process is referred to as Data Labeling: to give each unclustered data point a cluster label. The unclustered data points are also called unlabeled data points. Figure 3 shows the entire framework on clustering a very large database based on sampling and MARDL. In particular, MARDL is independent of clustering algorithms, and any categorical clustering algorithm can in fact be utilized in this framework. In MARDL, those unlabeled data points will be allocated into clusters via two phases, namely, the Cluster Analysis phase and the Data Labeling phase. The work doing in each phase is described below.

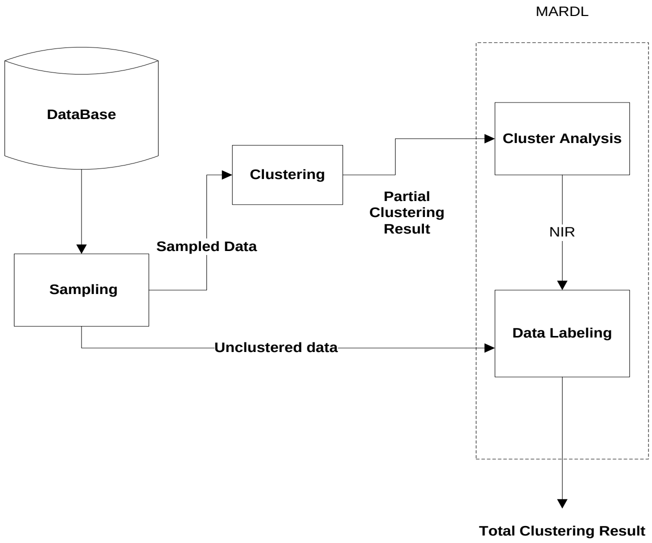


Fig.3. The framework of clustering a categorical very large database with sampling and MARDL.

2.2. Cluster Analysis Phase:

In the cluster analysis phase, a cluster representative is generated to characterize the clustering result. However, in the categorical domain, there Is no common way to decide cluster representative. Hence, a categorical cluster representative, named "**N**ode **I**mportance **R**epresentative" (abbreviated as *NIR*), is devised in this paper.NIR represents clusters by the attribute values, and the importance of an attribute value is measured by the following two concepts: (1) the attribute value is important in the cluster when the frequency of the attribute value is high in this cluster; (2) the attribute value is important in the cluster if the attribute value appears prevalently in this cluster rather than in other clusters. NIR identifies the significant components of the cluster by the important attribute values. Moreover, based on these two concepts to measure the importance of attribute values, NIR considers both the intra-cluster similarity and the inter-cluster similarity to represent the cluster.

2.3. Data Labeling Phase:

In the data labeling phase, each unlabeled data point is given a label of appropriate cluster according to NIR. By referring to the vector-space model [20], the similarity between the unlabeled data point and the cluster is designed analogously to the similarity between the query string and the document. According to this similarity measurement, MARDL allocates each unlabeled data point into the cluster which possesses the maximal resemblance. There are two advantages in MARDL: (1) high efficiency. MARDL is linear with respect to the data size. MARDL is efficient in essence and able to preserve the benefit of sampling on clustering very large database; (2) retaining cluster characteristics. MARDL gives each unlabeled data point a label of the cluster based on the partial clustering result obtained by clustering sampled data set. Since NIR considers the importance of the attribute value, MARDL will preserve clustering characteristic: high intra-cluster similarity and low inter-cluster similarity.

3. MARDL Algorithm

The algorithm MARDL is outlined below, where MARDL can be divided into two phases, the cluster analysis phase and the data labeling phase.

MARDL(C, U):

Clustering result C, unclustered data set U.

Procedure

main ():

The main procedure of MARDL

```
1. N Table=cluster analysis(C);
2. Data Labeling (N Table, U);
```

Procedure Cluster Analysis(C): analyze input clustering result and return the NIR hash table

```
Luster analysis(C)
1. While (C[next]! ='\0') {
2. p [i][j]=C[next];
3. divide Nodes (p [i][j]);
4. Update NF(c[i]);
5.}
6. For (N=$d_{i1}$; N<=$d_{it}$; N++)
7. Compute Weight f ($d_{ix}$);
8. For ( $C = c_1$ & C<=$c_n$; C++) {
9. For (N=$d_{i1}$&N<=$d_{it}$; N++) {
10. Calculate   n ($w_i$, $d_{ix}$)
11. Add NIR table NTable ($d_{ix}$,$w_i$, $d_{ix}$)}}
12. Return NTable;
```

Procedure Data Labeling (NTable,U): give each unclustered data point a cluster label

```
13. While (U[next]! ='\0') {
14. U [u}[j]=U[next};
15. Divide nodes (p[u][j]);
16. For (N=$c_1$ ;$N \leq c_n$;$N + +$)
17. Calculate Resemblance(C[m])
18. Give label c[m] to p[u][j] ;}.
```

The main purpose of the cluster analysis phase is to represent the prior clustering result with NIR. NIR represents cluster by a table which contains all the pairs of a node and its node importance. For better execution efficiency, the technique of hash can be applied on the represented table. Since the node names are never repeated, node is suitable to be a hash key for efficient execution. The main purpose of the data labeling phase is to decide the most appropriate cluster label for each unlabeled data point. Each unlabeled data point is labeled and then classified to the cluster which attains the maximal resemblance. The resemblance value of the specific cluster is computed efficiently by the sum of each node importance through looking up the NIR hash table q times. After all the resemblance values is computed and recorded, the maximal resemblance value is found, and the unlabeled data point is labeled to the cluster which obtains the maximal resemblance value. Note that after executing the data labeling phase, the labeled data point just obtains a cluster label but is not really added to the cluster. Therefore, NIR table will not be modified in the data labeling phase. This is because the MARDL framework does not cluster data, but rather, presents the original clustering characteristics to the incoming unlabeled data points.

4. CONCLUSION

This paper we formalized the definition of a cluster when the data consists of categorical attributes, and then introduced a fast summarization-based algorithm MARDL. To allocate each unlabeled data point into the appropriate cluster when the sampling technique is utilized to cluster a very large categorical database categorical cluster representative technique, named NIR, to represent clusters which are obtained from the sampled data set by the distribution of the nodes. The evaluation validates our claim that MARDL is of linear time complexity with respect to the data size, and MARDL preserves clustering characteristics, high intra-cluster similarity and low inter-cluster similarity. It is shown that MARDL is significantly more efficient than prior works while attaining results of high quality.

REFERENCES

[1] C. Aggarwal, J. Han, J. Wang, and P. Yu, "A Framework for Clustering Evolving Data Streams," Proc. 29th Int'l Conf. Very Large Data Bases (VLDB), 2003.

[2] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-Based Clustering over an Evolving Data Stream with Noise," Proc. Sixth SIAM Int'l Conf. Data Mining (SDM), 2006.

[3] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary Clustering," Proc. ACM SIGKDD '06, pp. 554-560, 2006

[4] Y. Chi, X.-D. Song, D.-Y. Zhou, K. Hino, and B.L. Tseng,"Evolutionary Spectral Clustering by Incorporating Temporal Smoothness," Proc. ACM SIGKDD '07, pp. 153-162, 2007..

[5] B.-R. Dai, J.-W. Huang, M.-Y. Yeh and M.-S. Chen, "Adaptive Clustering for Multiple Evolving Streams," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 9, pp. 1166-1180, Sept. 2006.

[6] M.M. Gaber and P.S. Yu, "Detection and Classification of Changesin Evolving Data Streams," Int'l J. Information Technology and Decision Making, vol. 5, no. 4, pp. 659-670, 2006.

[7] G. Hulten, L. Spencer, and P. Domingos, "Mining Time-ChangingData Streams," Proc. ACM SIGKDD, 2001.

[8] A. Jain and R. Dubes, Algorithms for Clustering Data. Prentice Hall,1988

[9] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," ACM Computing Surveys, 1999.

[10] O. Nasraoui and C. Rojas, “Robust Clustering for Tracking Noisy Evolving Data Streams,” Proc. Sixth SIAM Int’l Conf. Data Mining(SDM), 2006.

[11] H. Wang, W. Fan, P. Yun, and J. Han, “Mining Concept-Drifting Data Streams Using Ensemble Classifiers,” Proc. ACM SIGKDD,2003

.[12] G. Widmer and M. Kubat, “Learning in the Presence of Concept Drift and Hidden Contexts,” Machine Learning, 1996.

[13] M.-Y. Yeh, B.-R. Dai and M.-S. Chen, “Clustering over Multiple Evolving Streams by Events and Correlations,” IEEE Trans. Knowledge and Data Eng., vol. 19, no. 10, pp. 1349-1362, Oct. 2007.in Dynamic Web Sites,” IEEE Trans. Knowledge and Data Eng., vol. 20, no. 2, pp. 202-215, Feb. 2008.

[14] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, 2002.

[15] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining. Knowl. Discov., 1998.

[16] S. Guha, R. Rastogi, and K. Shim. ROCK: A Robust Clustering Algorithm for Categorical Attributes. InProc. of the 15th ICDE, 1999.

[17] V. Ganti, J. Gehrke, and R. Ramakrishnan.CACTUS stering Categorical Data Using Summaries. InProc. of ACM SIGKDD, 1999.

[18] S. Guha, R. Rastogi, and K. Shim. CURE: An Efficient Clustering Algorithm for Large Databases. InProc. of the ACM SIGMOD Conf., 1998.

[19] A. Jain and R. Dubes. Algorithms for Clustering Data.Prentiche Hall, 1988.

[20] R. Baeza-Yates and B. Riberiro-Neto. Modern InformationRetrieval. Addison-Wesley, 1999.

AUTHORS

AREGAKUTI SIDDHARTHA REDDY received his B.Tech degree in Computer Science and Engineering from Nalanda Engineering College, Andhra Pradesh, India, in 2010. He is Pursuing M.Tech in Computer Science and Engineering in K.L University, A.P, India during 2010-2012. His research invites Data Mining and Knowledge Discovery

CHAMARTHI VERRA VENKATA NAGA VARUN received his B.Tech in Information&Technnology and Engineering from Nalanda Engineering College, Andhra Pradesh, India, in 2009.He is Pursuing M.Tech in Computer Science and Engineering in K.L University, A.P, India during 2010-2012. His research invites Data Mining and Knowledge Discovery

- ANUSHAARE received her B.Tech degree in Computer Science and Engineering from SASTRA University, and M.Tech degree from K.L University .She is currently working as Assistant Professor in K.L University, She is actively engaged in research and publications in the areas of Computer Networks and Data Mining,