A Novel Similarity Measure for Clustering Categorical Data Sets

Rishi Sayal
HOD & Professor, Dept. of Computer Science & Engineering,
Guru Nanak Engineering College, Ibrahimpatnam,
Andhra Pradesh, India.

Dr. V. Vijay Kumar
Dean & Professor, Dept. of Computer Science & Engineering,
GIET, Rajahmundry, Andhra Pradesh, India.

ABSTRACT

Measuring similarity between two data objects is a more challenging problem for data mining and knowledge discovery tasks. The traditional clustering algorithms have been mainly stressed on numerical data, the implicit property of which can be exploited to define distance function between the data points to define similarity measure. The problem of similarity becomes more complex when the data is categorical which do not have a natural ordering of values or can be called as non geometrical attributes. Clustering on relational data sets when majority of its attributes are of categorical types makes interesting facts. No earlier work has been done on clustering categorical attributes of relational data set types making use of the property of functional dependency as parameter to measure similarity. This paper is an extension of earlier work on clustering relational data sets where domains are unique and similarity is context based and introduces a new notion of similarity based on dependency of an attribute on other attributes prevalent in the relational data set. This paper also gives a brief overview of popular similarity measures of categorical attributes. This novel similarity measure can be used to apply on tuples and their respective values. The important property of categorical domain is that they have smaller number of attribute values. The similarity measure of relational data sets then can be applied to the smaller data sets for efficient results.

Keywords: Data Clustering, Similarity measures, Context based similarity, Categorical attributes and functional dependency

1. INTRODUCTION

We Data clustering has attracted a lot of research attention in the field of computation statistics and datamining. The clustering techniques can be applied and used to perform similarity clusters and search, pattern recognition, trend analysis and so forth. Clustering [10] is the technique of grouping a set of physical or abstract objects into different clusters, such that objects with in a cluster are more similar to one another and are dissimilar to the objects in other clusters. A good clustering algorithm generates high quality clusters to yield low inter cluster similarity and high intra cluster similarity.

1.1 Attributes

In general, there are two types of attributes associated with input data in clustering algorithm i.e. numerical attributes and categorical attributes[9]. Numerical attributes are those with a finite or infinite number of ordered values such as the height of a person or the x – coordinate of a point on a 2D domain. On the other hand, categorical attributes are those with finite unordered values, such as the occupation or the blood type of a person. In most related studies, the dissimilarity between two clusters is defined as the distance between their centroids or the distance between two closests(or farthest) data points. However all these measures are prone to outliers and removal of outliers precisely is yet another difficult task.

In most conventional clustering problem, the similarity measurement mainly takes the numerical attributes into considerations, like the k-means algorithm is one of the most popular clustering algorithms because of its efficiency in clustering large data sets. However, k-means clustering algorithm fails to handle data sets with categorical attributes because it minimizes the cost function that is numerically measured. Most of the algorithms have been focused in clustering of numerical data sets. However, in many data mining applications the categorical attributes are what users are concerned about. Data points which are similar to one another in their categorical attributes may be scattered geometrically. The traditional approach to converting categorical data into numeric values does not necessarily produce meaningful results in the case where categorical domains are not ordered.

In this present study, the focus is only on the clustering of the categorical data. Many data clustering algorithms have been proposed on categorical data in the past which uses different similarity measures. Categorical databases do not contain numeric data and instead the domain of the attributes are small, unordered sets of values. An important instance can be market basket database containing record of purchase of customers, mostly where attributes are of product kind and rows represent customers. Our paper takes an instance of Television channel database where most attributes are of categorical kind and presents a similarity measure which is an extension to context based similarity [5], [8] where similarity between components is determined by checking the contexts in which they appear. For Example two products(attributes) are considered similar if their respective set of customers are similar. In view of this, a novel similarity measure for categorical data has been proposed in this research work.

1.2 Plan of the paper

This paper is organized as follows: We start by critically reviewing in Section 2 the approaches and similarity measures available in the literature. Section 3 provides a detailed description of the new similarity measure for clustering categorical data. The behavior of the similarity measure has been shown on sample relational data set in Section 4. Finally, Section 5 draws conclusions and highlights extensions to similarity measure, which are worth further research.

2. RELATED WORK

This section deals with all categorical clustering algorithms and their similarity approaches in finding the best of clusters and also deals with the previous work of finding and dealing similarity between one set of attributes with respect to other set known as context based similarity. Most of the earlier work has been done with k-means as the stepping platform to generate clusters on categorical attributes.

2.1 The *k*-modes Algorithm

The *k*-modes algorithm [14] is an extension to k-means algorithm to cluster categorical data by removing the drawback that was put by *k*-means by using a simple matching dissimilarity measure or the hamming distance for categorical data objects and replacing means of clusters by their modes. This algorithm has made three major modifications to the k-means: firstly it uses a different similarity or rather dissimilarity measure known as chi- square distance as mentioned below, it has replaced k-means to k modes and lastly uses a frequency based method to update modes. The dissimilarity measure of k-modes algorithm follows

2.1.1 Dissimilarity Measures

Let *A, B* be two categorical objects described by n categorical attributes. The dissimilarity measure between these two objects *A* and B can be thence be defined by the all and total mismatches of the corresponding attribute categories of the two objects. The smaller the number of mismatches is, the more similar the two objects. Formally,

$$d(A,B) = \sum_{i=1}^{m} \delta\ (a_i,\ b_i)$$

$$\text{where } \delta\ (a_i,\ b_i) = \begin{bmatrix} 0(a_i = b_i) \\ 1(a_i \neq b_i) \end{bmatrix}$$

$d(A,B)$ here gives more or less equal importance to each category of an attribute. The frequency of categories if taken into account can then lead to a dissimilarity measure as defined below

$$d_{x^2}\ (\mathrm{X,Y}) = \sum_{j=1}^{m} \frac{(n_{xj} + n_{yj})}{n_{xj} n_{yj}} \delta(x_j, y_j)$$

where n_{xj}, n_{yj} are the numbers of objects in the data set that have categories x_j and y_j for attribute *j*. Because $d_{x2}\ (X,\ Y)$ is similar to the *chi-square distance*. The dissimilarity measure here uses and prioritizes rare categories than to frequent ones.

2.2 K-Representative Algorithm

K-modes algorithm [14] has its own set of drawbacks because of its instability due to non-uniqueness of the modes i.e., the results of the clusters depend largely and strongly on the selection of modes during the clustering process. Huang combined k-modes with k-means to give k-prototype algorithm [15] but because of the K-mode problem, limitations remained same. K-representative algorithm, [7] works on the principal of "cluster centers" called representatives for categorical objects. Arithmetic operations are completely absent in the initialization and setting of categorical objects, it applies the notion of fuzzy logic in defining representatives instead of means for clusters. With this theory, it can formulate the clustering problem of categorical objects as a partitioning problem in the way similar to k-means clustering. The dissimilarity measure of this algorithm is as follows.

2.2.1 Dissimilarity Measure

The dissimilarity between a categorical object and the representative of a cluster is defined based on simple matching as follows.

Let C = {A_1, . . . ,A_p} be a cluster of categorical objects, with A_i = (a_i, *1*, . . . , $a_{i,m}$), 1 ≤ *i* ≤ p, denote by Dj the set formed from categorical attributes (a_{1j},, . . $a_{p,j}$.) and A = (a_1, . . . , $a_{i,m}$) be a categorical object. Here A may or may not belong to *C*. The assumption is that $Q = (q_1, \ldots, q_m)$, with $q_j = \{(c_j, f_{cj}) \mid c_j\ \varepsilon$ *Dj*}, is a representative of cluster C. k-representative defines the dissimilarity between object A and representative Q by

$$d(\mathrm{A},\ Q) = d(X,Q) = \sum_{j=1}^{m} \sum_{c_j \varepsilon D_j} f_{cj}\, \delta\ (x_j,\ c_j)$$

The dissimilarity *d(A, Q)* is mainly dependent on the relative frequencies of categorical values within the cluster and simple matching between categorical values. Here it is important to note that the simple matching dissimilarity measure between categorical objects can be considered as a categorical counterpart of the squared Euclidean distance measure. It can be observed that

$$d(\mathrm{A},\ Q) \quad = d(X,Q) = \sum_{j=1}^{m} \sum_{c_j \varepsilon D_j} f_{cj}\, \delta\ (x_j, c_j)$$

$$= d(X,Q) = \sum_{j=1}^{m} \sum_{c_j \varepsilon D_j, \neq x_j} f_{cj}$$

$$= \sum_{j=1}^{m} (1 - f_{xj})$$

where f_{xj} is the relative frequency of category x_j within *C*.

2.3 K-Histograms Algorithm

The *k*-histograms algorithm [13] extends the *k*-mean algorithm to categorical data by replacing the means of clusters with histogram. It dynamically updates histograms in the clustering process and is found to be very high in accuracy. In general, this algorithm is very similar to the *k*-modes algorithm except that it uses the histogram data structure to describe a categorical data cluster instead of mode.

2.3.1 Dissimilarity Measures

The dissimilarity measure is defined here in terms of a histogram H=(h_1 ,h_2…..h_m) which is compact representation of dataset D and an object Y.

$$d(H,Y)=\frac{\sum_{j=1}^{m}\psi(h_j,y_j)}{n}$$

2.4 RAHCA Algorithm

The Rough Set-Based Agglomeration Hierarchy Clustering Algorithm, RAHCA, [3] for categorical data proposes a new categorical similarity measure based on Euclidean distance so as to better solve the problem of difficult measurement of categorical data because of the non-numerical data nature. RAHCA describes the clusters using the notations: U is universe, one element x_i € U is called as an object, A is the attribute set and d is the introduced decision attribute

Definition 1: In *(U,A*U*{d}), A={a1,...,as}, P=U/R{d}={D1,...,Dr}, n=*|*U*|, ∀ *f, h* ℰ *{1,...,r}*, the similarity between two clusters D_f and D_h in *P* can be defined as follows:

$$SIM(f,h)=\frac{1}{S}\sum_{K=1}^{S}\{1-[\frac{1}{n}\sum_{i=1}^{n}(\mu_k(i,f)/|D_f|-\mu_k(i,h)/|D_h|)^2]^{1/2}\}$$

The Definition 1: mentions Euclidean distance to describe the similarity among clusters so that more the value of distance is, the smaller the value of similarity is. Not only the similarity defined using Euclidean distance measure, [3] the numbers of the same attributes between two clusters and differences in the numbers of dissimilar attributes, but also the fact lies in expressing the degrees of the similar and dissimilar attributes, whose nature is to do numerical processing of categorical attributes. In above equation, μ_k *(i,f) /* |D_f| *and* μ_k*(i,h)/*|D_h| indicate the centers of cluster D_f and cluster D_h respectively. If D_f and D_h contain one object, then |D_f| = |D_h|=*1*. Thence Definition 1 can be adapted to the situation such that it expresses the similarities among the cluster versus the cluster, and the object versus the cluster as well as the object versus the object.

2.5 ROCK Algorithm

ROCK, RObust Clustering using links, [6] an adaptation of an agglomerative hierarchical clustering algorithm, proposes a new concept of links to measure the similarity between a pair of data points and helps to overcome the problems with L_p distance metrics and Jaccard coefficient. It uses links and not distances when merging clusters and also extends to non-metric similarity measures that are relevant in situations where a domain expert/similarity table is the only source of knowledge.

ROCK optimizes a criterion function defined in terms of the number of "links" between tuples. The number of links between two tuples is the number of common neighbors (Given a *similarity function*, two tuples in the dataset are said to be *neighbors* if the similarity between them is greater than a certain threshold value) they have in the dataset. Starting with each tuple in its own cluster, they repeatedly merge the two closest clusters till the required number (say, *K*) of clusters remains. Since the complexity of the algorithm is *cubic* in the number of tuples in the dataset, they cluster a sample randomly drawn from the dataset, and then partition the entire dataset based on the clusters from the sample. Beyond that the set of all "clusters" together may optimize a criterion function, the set of tuples in each individual cluster is not characterized.

2.6 STIRR Algorithm

STIRR, Sieving Through Iterated Relational Reinforcement, [2] is an iterative algorithm which clusters attribute values and does not define a distance measure between attribute values and produces just two clusters of values. It is based on non-linear dynamical system [2] over multiple copies of a hypergraph of weighted attribute values, until a *fixed point* is reached. Each copy of the hypergraph contains two groups of attribute values, one with positive and another with negative weights, which define the two clusters. They represent each attribute value as a weighted vertex in a graph. Multiple copies b_1… b_m, called *basins*, of this set of weighted vertices are maintained; the weights on any given vertex may differ across basins. b_1 is called the *principal* basin; $b_{2,\dots\dots,}$ b_m are called *non-principal* basins. The process starts with a set of weights on all vertices (in all basins) and the system is "iterated" until a fixed point is reached. The weights in one or more of the basins $b_{2\dots}$ b_m isolate two groups of attribute values on each attribute, when the fixed point is reached. The first with large positive weights and the second with small negative weights, and that these groups correspond intuitively to projections of clusters on the attribute.

2.7 CLOPE Algorithm

CLOPE, Clustering with sLOPE, algorithm [12] proposes an approach based on histograms: The goodness of a cluster is higher if the average frequency of an item is high, as compared to the number of items appearing within a transaction. The algorithm is particularly suitable for large high- dimensional databases, but it is sensitive to a user-defined parameter (the repulsion factor), which weights the importance of the compactness/sparseness of a cluster. A better cluster is reflected graphically if higher height to weight ratio is achieved. CLOPE uses histograms of a cluster C with items as the X –axis decreasingly ordered by their occurrences and occurrences as y-axis. A larger height means a heavier overlap among the items in the cluster and thus more similarity among transactions in the cluster.

2.8 COOLCAT Algorithm

Categorical clustering can also be tackled by using information-theoretic principles and the notion of entropy to measure closeness between objects. The basic intuition is that groups of similar objects have lower entropy than those of dissimilar ones. Thus, the COOLCAT algorithm [1] proposes a scheme where data objects are processed incrementally, and a suitable cluster is chosen for each tuple such that at each step, the entropy of the resulting clustering is minimized. This algorithm is a scalable algorithm that optimizes the objective function as the entropy [1] of the clustering and depends on sampling. It is non-hierarchical and starts with a sample of points and identifies a set of k initial tuples such that the minimum pair wise distance among them is maximized. These serve as representatives of the k clusters. All remaining tuples of the data set are placed in one of the clusters such that, at each step, the increase in the entropy of the resulting clustering is minimized. COOLCAT [1] uses the fact that entropy (measure of amount of disorder in a system) can serve as a measure of similarity along with any set of vectors not just two, unlike jaccard coefficent which takes two objects.

2.9 LIMBO Algorithm

The scaLable InforMation BOttleneck, LIMBO, algorithm [8] gives a notion of entropy to catch the similarity between objects and defines a clustering procedure that minimizes the information loss. The algorithm builds a Distributional Cluster Features (DCF) tree to summarize the data in *k* clusters, where each node contains statistics on a subset of tuples. Then, given a set of k clusters and their corresponding DCFs, a scan over the data set is performed to assign each tuple to the cluster exhibiting the closest DCF. LIMBO algorithm that builds on the *Information Bottleneck (IB)* framework [8] for quantifying the relevant information preserved when clustering and it has the advantage that it can produce clustering of different sizes in a single execution.

It uses the IB framework to define a distance measure for categorical tuples and it also presents a novel distance measure for categorical attribute values. Categorical data is characterized by the fact that there is no inherent distance between attribute values. Two attribute values are similar if the contexts in which they appear are similar. It defines the context as the distribution these attribute values induce on the remaining attributes. This approach has the advantage that it allows for the definition of distance between clusters of values, which can be used to perform intra-attribute value clustering.

2.10 CACTUS Algorithm

CACTUS algorithm [4] is an agglomerative algorithm using strong connection and similarity to cluster categorical data and it is a fast summarization based algorithm which exploits the small domain size of categorical attributes. The basic idea is that summary information constructed from the dataset is sufficient for discovering well defined datasets. The summary information is of two types; inter attribute summaries [4] consisting of all strongly connected values from different attributes value pairs where each pair has attribute values from different attribute and intra attribute summaries consisting of similarities between attribute values of the same attributes.

The support for an attribute value pair (a_i, a_j), where a_i is in the domain of attribute A_i and a_j in the domain of attribute A_j is defined as the number of tuples that have these two values. The two attributes a_i; a_j are strongly connected if their support exceeds the value expected under the attribute-independence. This concept is then extended to sets of attributes. A cluster in CACTUS is defined as a area of attributes that are pair wise strongly connected, no sub-region has the property, and its support is greater than the expected support under the attribute-independence assumption. Similarity concept is based on connecting attribute values (a_1; a_2) of the same attribute A_i, and it measures how many “neighboring" values *x* belonging to other attributes exist, such that a_1; *x* and a_2; *x* have positive support. Using support and similarity, CACTUS defines inter-attribute and intra-attribute summaries and similarities which tells how related values from different and the same attribute are respectively and it uses these summaries to compute the so-called cluster projections on individual attributes and use these projections to obtain candidate clusters on a pair of attributes, extending then to three and more attributes. CACTUS, thence, bases its clustering results on the “neighboring" concept of similarity.

3. PROPOSED WORK

In this section, to explain context based similarity, [5], [8] consider the problem of defining (dis)similarity between attributes of a relation which can have various applications in forming clusters of attributes. In absence of any knowledge we can use standard similarity measures such as correlation and Euclidean distance but in case of a super market scenario, we might say that white butter and yellow butter are dissimilar because they may not have the common customers because of taste. But in reality, both are butters and should be similar. Such similarities can be explained on context based similarity where two products are similar if the taste or buying patterns of the customer are similar. Hence yellow butter and white butter are sub relations of the data base and we can relate this similarity between attributes to similarity between certain sub relations (context).

Without a measure of distance between data values, it is very difficult to find the similarity measure when the data is categorical and relational. In relational database, since data values are conveyed through their respective attributes, the similarity measure of one tuple may always be expressed in terms of other tuple i.e., attributes can be taken as context to measure the similarity of other attribute. Most of the context based similarity measures do not use the property of relational data set where two values of the attributes if are same, let’s say (X) and are dependent on some other attribute (Y) then they can be grouped into a dependency form Y → X, value of X is always determined by Y or Y determines X or X dependent on Y.

The above notion of relational data set can be taken as a similarity measure in the context based similarity. Our main objective is to determine a similarity measure in relational data set that contains most of categorical attributes and obeying the law of relational datasets where if two values of a attribute are similar with respect to another value of other attribute then functional dependency [11] between these two attributes exists and can be taken as a measure for similarity based on context.

Let α and β be the two attributes of a relation R such that the notion of functional dependency $\alpha\rightarrow\beta$ holds good, for all pairs of tuples t_1 and t_2 such that $t_1[\alpha] = t_2[\alpha]$ then it is also the case of $t_1[\beta] =t_2[\beta]$

Using the functional dependencies notation we say that K is a super key of R if K → R i.e., K is a super key if when $t_1 [K] = t_2 [K]$, it is also the case that $t_1[R] = t_2[R]$ in other words $t_1= t_2$.

Table 1: Instance of sample relation r

X	A	B	C	D	M	N
X_1	a_1	b_1	c_1	d_1	M_1	N_1
X_2	a_1	b_2	c_1	d_2	M_1	N_2
X_3	a_2	b_2	c_2	d_2	M_2	N_2
X_4	a_2	b_3	c_2	d_3	M_2	N_3
X_5	a_3	b_3	c_2	d_3	M_3	N_3

In Table: 1 there are two tuples that have an a_1 value for attribute A and they have the same value c_1 for attribute C and the functional dependencies exists are

A→C, B→D

AB → D since there are no pairs distinct tuples t_1 and t_2 such that $t_1[AB] = t_2[AB]$

Therefore, if $t_1[AB] = t_2[AB]$, then $t_1 = t_2$ and $t_1[D] = t_2[D]$ and hence satisfies AB → D.

Table 1 contains two clusters namely M and N with their subclusters.The context based similarity is based on the fact that there is no inherent distance between attribute values of categorical data, for example in the Table 1 for the values X_1 and X_2, it is not clear as how to assess their similarity, hence they are placed on a context. These values are similar if the context in which they appear are similar. Context is defined as the distribution of these attributes values that have inducing effect on remaining attributes. X_1 and X_2 are considered similar if they induce a similarity context which is here functional dependency, hence X_1 and X_2 belong to the cluster M_1 obeying the functional dependency A → C and X_3 and X_4 belong to the cluster M_2 as they obey functional dependency but context changes. Similar case is applied on cluster N where B → D. X_2 and X_3 belong to the same cluster N_2 as the distribution of d_2 is high and they obey functional dependency B → D. In the same context X_4 and X_5 belongs to cluster N_3 as distributions of the values d_3 is high inducing an effect on the attributes X.

Applying the theory of context based similarity, X_1 and X_2 are similar in the context to A & C as for every repeated value of A, the value of C is repeated hence X_1 and X_2 are similar and belongs to the same cluster with respect to context based similarity. Subcluster M_3 and N_1 indicate value other than the one based on similarity that do not obey in context though they obey functional dependency A → C and B → D

4. EXPERIMENTAL RESULTS

This proposed similarity measure was experimented on Indian Televison Channel data base to gauge similarity between various Television Channels broad casting popular programs. The data set is a real one containing around 50 tuples but for our test cases and for evaluation, the sample data base was reduced to an instance of 6 tuples. The clusters in the dataset are named U and V. The experiments are shown as a test a cases below.

4.1 Test on television channel dataset

The similarity based on functional dependency [11] and context based was tested on TV channel database where each channel shows a different program at a different slot. The programs may be directed by the same director on a different channel depending on the slot available. The similarity measure proposed here measures the similarity between various channels in context with other categorical attributes viz director, actor by first taking functional dependency into account and then finding the (dis)similarity between various channels.

Table 2: An instance of Television Channel Dataset

Channel	Director	Actor	Slot	Program	U	V
ETV	Shri	AK	Morning	Serial	U_3	V_2
MAA	Kapoor	AB	Evening	Movie	U_1	V_1
STAR	Kapoor	AB	Morning	Movie	U_1	V_1
SONY	Venkat	AK	Noon	Serial	U_3	V_2
ZEE	Bhalla	HR	Noon	Drama	U_2	V_3
ZOOM	Bhalla	KB	Noon	Drama	U_2	V_3

The Television Channel data base consists various Indian Channels Broadcasting movie, Drama, Serials and Music among other programs as part of the schedule. This programs are allotted various slots as per the popularity of the program like Drama and Serials are shown in afternoon slot for the household woman chore. Further each program is directed by channels favourite director (In case of movie, popular movies are shown directed by famous directors)

To find similarity between various channels there seem to be no distance or numerical factor involved to find out the similarity based on distance.

Here context based similarity can very well work with the concept of functional dependency. The functional dependency realistically are

$$\text{Director(D)} \rightarrow \text{Program(P)} \quad (1)$$

$$\text{Actor(A)} \rightarrow \text{Program} \quad (2)$$

$$D / A \rightarrow P \quad (3)$$

Considering only (1) and (2) as for our case, star and MAA channels are similar in context with director and program(D and P attributes are part of functional dependency) and applying context based theory, the distribution of the same is high leading to sub cluster U_1 of U .Similar case is applied to other directors and programs leading to subcluster U_2. U_3 is the sub cluster which obeys functional dependency but probabilistic approach in context based similarity is less as distribution is less for respective director and programs. Similar case can be applied for Subcluster, V_1 for (2) functional dependency.

Thence the usage of concept of mix approach of functional dependency and context works very well for smaller datasets leading to better sub clusters and clusters.

5. CONCLUSION AND FUTURE WORK

In this paper, a novel similarity measure for categorical attributes of relational data sets has been proposed based on the intuitive idea of functional dependency and Context based similarity. The idea is generalized with a functional dependency that also uses context based of transactions in a cluster, and thus the resulting number of clusters. Our application shows that this similarity measure is quite effective in finding interesting clustering of relational data sets.

How reasonable, realistic the proposed similarity measure works with the real data containing both categorical and numerical attributes has to be investigated. Most of the current clustering algorithms are numerical based using common distance measure as a similarity measures. This proposed similarity measure should have a theoretical study on the impact of these clustering algorithms.

6. ACKNOWLEDGMENTS

The authors would like to thank Sardar Tavinder Singh, Chairman and Mr. Gagandeep Singh Kohli, Vice Chairman, Guru Nanak Institutions for their encouragement and support for this research work. The authors would like to express their gratitude to the reviewers for their valuable suggestions and comments. The work is (partially) supported by research grants from the R&D of Guru Nanak Engineering College.

7. REFERENCES

[1] Daniel Barbara´ , J. Couto, and Y. Li, “COOLCAT: An Entropy-Based Algorithm for Categorical Clustering,” Proc.11th ACM Conf. Information and Knowledge Management (CIKM ’02), pp. 582-589, 2002.

[2] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Clustering categorical data: An approach based on dynamical systems. In Proceedings of the 24th InternationalConference on Very Large Databases, pages 311– 323, New York City, New York, August 24-27 1998.

[3] Duo Chen, Du-Wu Cui, Chao-Xue Wang, Zhu-Rong Wang “A Rough Set-Based Hierarchical Clustering Algorithm for Categorical Data” International Journal of Information Technology, Vol.12, No.3, 2006

[4] Ganti V, J. Gehrke and R.Ramakrishnan. “CACTUS: Clustering Categorical data using summaries.” In Proc Int Conf Knowledge Discovery and Data Mining, 1999, pp.73-88

[5] Gautam Das, Heikki Mannila “Context-Based Similarity Measures for Categorical Databases.” PKDD 2000: 201-210.

[6] Guha S, R Rastogi & K. Shim “ROCK: A robust clustering algorithm for categorical attributes.” In Proc. IEEE Int. Conf. on Data Engineering ,1999 pp 512-521

[7] Ohn Mar San, Van-Nam Huynh, and Yoshiteru Nakamori “An Alternative Extension of The K-Means Algorithm For Clustering Categorical Data” International Journal of Appl. Math. Comput. Sci., 2004, Vol. 14, No. 2, 241–247

[8] P. Andritsos, P. Tsaparas, R. Miller, and K. Sevcik, “LIMBO: Scalable Clustering of Categorical Data,” Proc. Ninth Int’l Conf. Extending Database Technology (EDBT ’04), pp. 123-146, 2004.

[9] Rishi Sayal, D. Durga Bhavani, P. Harsha and Dr. V. Vijaya Kumar “Study of Hierarchical and Partitional Clustering Techniques ”International Conference on Soft Computing & Intelligent Systems” ICSCIS-07, pp. 74-80, 2008.

[10] Rui Xu, Donald Wunsch II, “Survey of Clustering Algorithms”, IEEE in Neural Networks 16(3)(2005).

[11] Silberschatz, Korth, “Data Base System Concepts”, Mc Graw hill, V Edition.

[12] Y. Yang, X. Guan, and J. You, “CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data,” Proc. Eighth ACM Conf. Knowledge Discovery and Data Mining (KDD ’02), pp. 682-687, 2002

[13] Zengyou He, Xiaofei Xu, Shengchun Deng, Bin Dong “K-Histograms: An Efficient Clustering Algorithm for Categorical Dataset” http://arxiv.org/abs/cs/0509033

[14] Zhexue Huang “A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining”

[15] Zhexue Huang, “Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values,” Data Mining and Knowledge Discovery, vol. 2, no. 3, pp. 283-304, 1998.