VOL. 7, NO. 9, SEPTEMBER 2012 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

AN INTEGRATED GHSOM-MLP WITH MODIFIED LM ALGORITHM FOR MIXED DATA CLUSTERING

D. Hari Prasad[1] and M. Punithavalli[2]
[1]Department of Computer Applications, Sri Ramakrishna Institute of Technology, Coimbatore, India
[2]Department of Computer Applications, Sri Ramakrishna Engineering College, Coimbatore, India
E-Mail: hari.research@yahoo.com

ABSTRACT

Data clustering is one of the common approaches used to carry out statistical data analysis, which is used in several fields, together with machine learning, data mining, customer requirement, trend investigation, pattern identification and image analysis. Even though many clustering approaches have been available, but most of them manage only the clustering of numerical data. On the other hand, the problem of clustering mixed data is more complicated and difficult as mixed data have nominal attributes. A large number of these algorithms deals with only on numeric data, a small number of algorithms take care of nominal data, and only a very least amount of algorithms can handle both numeric and nominal values. In order to provide an efficient mixed data clustering, there is a significant need for several approaches to handle mixed data clustering. The existing mixed data clustering techniques takes more time for clustering the usage of SOM has the inability to capture the inherent hierarchical structure of data. To overcome this, an integrated GHSOM-MLP with Modified LM Algorithm is proposed in this paper. The experimentation for the proposed technique is carried with the help of UCI Adult Data Set to compare the proposed approach with GHSOM, in terms of number of resultant clusters and mean square error.

Keywords: mixed data clustering, growing hierarchical self-organizing map (GHSOM), modified LM algorithm, attribute-oriented induction, data mining.

INTRODUCTION

One of the most popular data mining approaches which are adequate for numerous applications is clustering. The major reason for its wide range of application is the capability of clustering technique to work on datasets with least or no previous knowledge. This enables clustering convenient for many real world applications. In recent times, high dimensional data has stimulated the attention of database researchers because of its significant challenges brought to the research community. In huge dimensional space, the distance between a record to its adjacent neighbor can approach its distance to the outermost record [1]. In the framework of clustering, the difficulty causes the distance among two records of the same cluster to move toward the distance among two records of various clusters. Conventional clustering approaches possibly will be unsuccessful to recognize the accurate clusters.

Clustering is the unsupervised classification of patterns into groups. It is an essential data analyzing method that arranges a collection of patterns into clusters in accordance with certain similarities [2-4]. Clustering is one of the significant techniques in numerous exploratory pattern-analysis, grouping, decision-making, and machine learning application. Clustering techniques have been effectively exploited in wide ranges of areas comprising pattern recognition, biology, psychoanalysis, archaeology, geology, topography, marketing, image processing and information retrieval [5].

With the huge development in both the computer hardware and software, an enormous amount of data is produced and gathered every day. These data can be used very effectively only when the meaningful information can be extracted to find the hidden information. On the other hand, the considerable difficulty for acquiring the best information from data is owing to the limitations of the data itself [6]. These major difficulties of gathered data come from their enormous size and versatile domains. Consequently, data mining is to discover interesting patterns from huge collections of data within limited sources (i.e., computer memory and execution time) has turned out to be popular in recent years

Clustering is a significant area of research for both data investigation and machine learning applications. Since new difficulties emerges continuously with the growth in different kinds of data and new techniques have to be developed to take care of large amount of data, heterogeneous in nature (numerical, symbolic, spatial, etc.). Several approaches have been developed with the purpose of arranging, summarizing or to assembling a variety of data into a set of clusters, in such a way that data belonging to an identical cluster are similar and data from other clusters are dissimilar [7, 8].

On the other hand, most of the conventional clustering approaches are developed to focus either on numeric data or on categorical data [9]. The collection of data in real world dataset would typically have both numeric and categorical attributes. It is more complicated for applying conventional clustering approaches directly into these kinds of mixed data.

In practice, a common technique to cluster databases with nominal attributes (columns) is to transform them into numeric elements and exploiting a numeric clustering technique to carry out the clustering process. This is typically carried out by “exploding” the nominal element into a collection of new binary numeric

elements, one for each different value in the original element. This way of handling mixed data suffers from several limitations. Initially, the dimensionality of the problem is increased which is already high. This considerably reduces the quality of clustering outcome [10, 11]. Also, the computational cost incurred may be huge.

For dealing with mixed numeric and categorical data, only few techniques are available. One of the techniques is usage of Self-Organizing Map (SOM) [12, 13] and Extended Attribute-Oriented Induction (EAOI) for clustering mixed data type. This will take more time for clustering. To overcome this, an integrated GHSOM-MLP with Modified LM Algorithm [14] is proposed in this paper.

RELATED WORKS

This section of the paper presents an overview on these clustering algorithms put forth by various authors.

A novel method [15] was put forth by Juha *et al*., for clustering of Self-Organizing Map. According to the method proposed in this paper the clustering is carried out using a two-level approach, where the data set is first clustered using the SOM, and then, the SOM is clustered. The purpose of this paper was to evaluate if the data abstraction created by the SOM could be employed in clustering of data. The most imperative advantage of this procedure is that computational load decreases noticeably, making it possible to cluster large data sets and to consider several different preprocessing strategies in a restricted time. Obviously, the approach is applicable only if the clusters found using the SOM are analogous to those of the original data.

A robust and scalable clustering algorithm was put forth by Tom *et al*., in [16]. They employed this clustering algorithm for mixed type attributes in large database environment. In their paper, they proposed a distance measure that enables clustering data with both continuous and categorical attributes. This distance measure is derived from a probabilistic model that the distance between two clusters [17] is equivalent to the decrease in log-likelihood function as a result of merging. Calculation of this measure is memory efficient as it depends only on the merging cluster pair and not on all the other clusters. The algorithm is implemented in the commercial data mining tool Clementine 6.0 which supports the PMML standard of data mining model deployment. For data with mixed type of attributes, their experimental results confirmed that the algorithm not only generates better quality clusters than the traditional k-means algorithms [18], but also exhibits good scalability properties and is able to identify the underlying number of clusters in the data correctly.

Mark Girolami presents a Mercer Kernel-Based Clustering [19] algorithm in Feature Space. This paper presents a method for both the unsupervised partitioning of a sample of data and the estimation of the possible number of inherent clusters which generate the data. This work utilizes the perception that performing a nonlinear data transformation into some high dimensional feature space increases the probability of the linear separability of the patterns within the transformed space and therefore simplifies the associated data structure. In this case, the eigenvectors of a kernel matrix which defines the implicit mapping provides a means to estimate the number of clusters inherent within the data and a computationally simple iterative procedure is presented for the subsequent feature space partitioning of the data.

Jian *et al*., in [20] proposed an efficient algorithm for clustering mixed type attributes in large dataset. Clustering is an extensively used technique in data mining. At present there exist many clustering algorithms, but most existing clustering algorithms either are restricted to handle the single attribute or can handle both data types but are not competent when clustering large data sets. Few algorithms can do both well. In this article, they proposed a clustering algorithm [21] that can handle large datasets with mixed type of attributes. They first used CF*tree (just like CF-tree in BIRCH) to pre-cluster datasets. After that the dense regions are stored in leaf nodes, and then they looked every dense region as a single point and used the ameliorated k-prototype to cluster such dense regions. Experimental results showed that this algorithm is very efficient in clustering large datasets with mixed type of attributes.

METHODOLOGY

Growing hierarchical self-organizing map (GHSOM)

The Growing Hierarchical Self-Organizing Map comprises of a hierarchical structure of multiple layers in which each layer has several independent mounting Self-Organizing Maps. Starting from a top-level map, each map which is identical to the Growing Grid model develops in size to represent a gathering of data at a particular level of detail. After a definite enhancement concerning the granularity of data representation is achieved, the units are examined to observe whether they symbolize the data at a specific minimum level of granularity. Those units that symbolize too diverse input data are extended to create a new small growing SOM at a successive layer, where the relevant data shall be characterized in more detail. These new maps yet again grow in size until a specific enhancement of the quality of data representation is reached. Units indicating a previously rather homogeneous set of data, conversely, will not need any additional expansion into succeeding layers. The obtained GHSOM therefore is completely adaptive to reflect, by its very architecture, the hierarchical structure inherent in the data, assigning additional space for the illustrating of inhomogeneous areas in the input space.

A graphical demonstration of a GHSOM is provided in Figure-1. The map in first layer contains 3 X 2 units and offers a rather rough grouping of the chief clusters in the input data. The six independent maps in the second layer provide a more detailed view of the data. The input data for single map is the subset that has been mapped onto the respective unit in the upper layer. Two

units located in one of the second-layer maps have additional been extended into third-layer maps to offer adequately granular input data depiction. It has to be seen that the maps contains various sizes based on the structure of the data that mitigates the trouble of earlier defining the structure of the architecture. The layer 0 serves up as an illustration of the entire data set and is essential for the managing of the growth process.

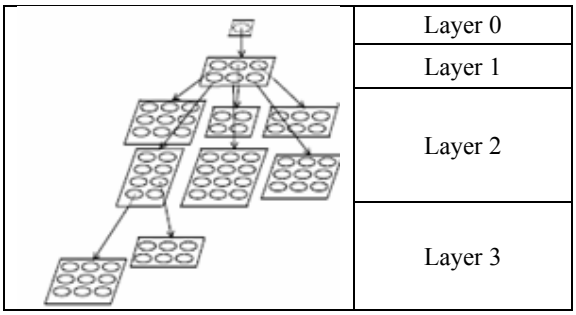

Figure-1. Trained GHSOM: The GHSOM evolves to a structure of SOMs reflecting the hierarchical structure of the input data.

Training and growth procedure of a growing GHSOM a newly generated map is trained based on the standard SOM training procedure. A high quantization error (qe) represents that an inhomogeneous part of the input space containing dissimilar data, or at least a rather large set of input data from a highly homogenous part of the input space is indicated by this unit. As a result, new units are required to offer more space for suitable data representation. The unit with the highest qe is therefore chosen and is indicated as the error unit. The error unit is mentioned as e. Subsequently, the majority of different adjacent unit d by means of input space distance is chosen. This is performed by means of contrasting the model vectors of every neighboring unit with the model vector of the error unit e. A new row or column of units is introduced among the error unit e and its most dissimilar neighbor d. The model vectors of the new units are kept as the average of their equivalent neighbors. There is lot of confusions exist in the training process of GHSOM, since it uses quantization error as major criteria for proceeding the training process [21]. Hence in this paper, GHSOM is incorporated with MLP (Multi Layer Perceptron) [22], trained with Modified Levenberg-Marquardt (LM) Learning.

GHSOM -MLP with modified LM learning technique

Due to the drawback of the existing GHSOM, an improved learning approach is introduced for MLP in this proposed approach. Modified LM approach is used for training MLP. Modified LM approach uses a technique which modifies the Learning parameter that resulted in decrease of both learning iteration and oscillation. A modification technique called Modified LM by altering the learning parameter has been used to accelerate LM algorithm. In addition, the error oscillation has been considerably reduced.

GHSOM-MLP with modified LM

The network's architecture is clearly shown in Figure-2 which has eight layers. The first layer with p nodes scales the data: it is the scaling interface between user and the system at the input side. The second and third layers comprise the GHSOM layer. The output of the scaling layer is fed as the input to the GHSOM layer. So, the second layer has p nodes. As discussed earlier for the GHSOM net, there are complete connections between layers 2 and 3.

The output layer of the GHSOM net possesses K number of nodes. So, there are K MLP networks, each of which receives inputs. As a result, the fourth layer has K_p nodes. These K_p nodes comprise the input layer of a set of K MLP networks. Without any loss of generalization, it is presumed that each of the K MLP networks contains only one hidden layer, even though it could be more than one and it can change for different MLP nets. The nodes in layer four is numbered as N_i, i = 1, 2,…, Kp. Nodes N_1 to N_p will be the input nodes of the first MLP (M_2); nodes N_{p+1} to will be input nodes of the second MLP (M_2); Similarly, nodes $N_{(K-1)\,p+1}$ to N_{np} will be the input nodes of Kth MLP, M_K. p = 9k as mentioned earlier. The jth input node of MLP M_i gets the jth normalized input (say, x_j) and passes it on to the first hidden layer of M_i.

The output of the node of the GHSOM (say, O_i) is linked to the output of every node of the last layer of M_i. The product of the MLP output and the GHSOM output then moves to layer 7. The product can be computed using an additional layer with two neurons for each MLP. Since only one of the GHSOM outputs will be one, and the remaining will be zero, only one of the MLPs will pass its output unattenuated to layer 7. The left behind (k-1) MLPs will transmit zero to layer 7. Because it is presumed that only one hidden layer, the nodes in layer six are the output nodes of the MLP nets. Each MLP, M_i will have two output nodes. These nodes are represented by O_{ij}^{6} where the index corresponds to the MLP, M_i and j = 1, 2. Layers 4-6 together constitute the MLP layer in Figure-2.

The outputs of this MLP-layer are then aggregated in layer seven which has just two nodes. These two nodes are represented as m and M. Now nodes O_{i1}^{6}, ∀ i = 1,2,…K are connected to node m and O_{i2}^{6}, ∀ i = 1,2,…K are connected to node M. All connection weights between layers 6 and 7 are fixed to unity and nodes m and M compute the weighted sum of all inputs as the output which is then passed to the scaling layer. It is observed that the network architecture guarantees that the aggregated output that is provided to the scaling layer is the output of the MLP equivalent to the winning node of the GHSOM net.

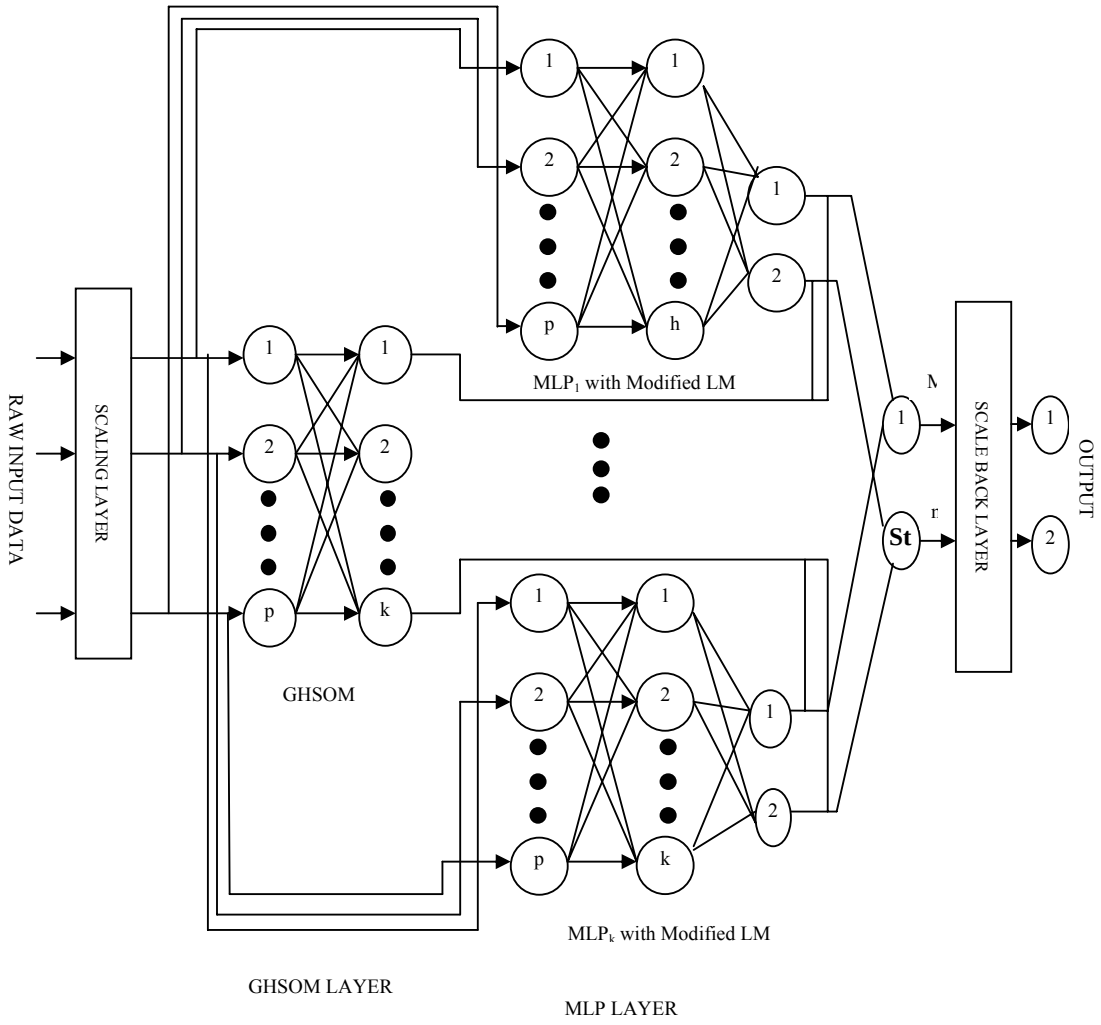

Figure-2. Architecture of GHSOM-MLP with modified LM for mixed data clustering.

The main reason for using GHSOM is that the prototypes formed by GHSOM preserve both topology and density. This density preservation attribute must be used. As the density matching property of GHSOM, if a specific region of the input space contains often occurring stimuli, it will be denoted by a larger area in the feature map than a region of the input space where the stimuli occur rarely. As a result, if there is a dense area in the input space, more prototypes will be positioned there by GHSOM. So, there will be more competitive MLPs for dense regions. Thus, finer aspects of the process can be modeled efficiently and this results in the overall enhancement of the performance.

The training phase of the hybrid GHSOM -MLP is described in this section. Input normalization (i.e., scaling) layer is used to normalize X_{tr}. Then the GHSOM is trained with the normalized X_{tr}. Once the GHSOM training is over, X_{tr} is partitioned into K subsets, $X_{tr}^{(l)}$ l = 1, 2,…, K as follows:

$$X_{tr}^{(l)} = \{x_i \in R^p \mid \|(x_i - v_l)\| = \min_j \|(x_i - v_j)\|\} \quad (1)$$

$X_{tr}^{(l)}$ can also be said as the set of input vectors for which the *l*th prototype, of the GHSOM becomes the winner. Let $Y_{tr}^{(l)}$ be the set of output vectors associated with vectors in $X_{tr}^{(l)}$. Now multilayer perceptron nets M_1, M_2,…, M_K, is trained with Modified LM algorithm.

In this paper, an effective learning algorithm called Modified LM Algorithm is used for learning to improve the overall performance of the proposed approach.

A Modified LM algorithm is used for training the neural network. Considering performance index is $F(w) = e^T e$ using the Newton method the equation obtained is as follows:

$$W_{K+1} = W_K - A_K^{-1} \cdot g_K \quad (2)$$

$$A_k = \nabla^2 F(w)\big|_{w=w_k} \quad (3)$$

$$g_k = \nabla F(w)\big|_{w=w_k} \quad (4)$$

$$[\nabla F(w)]_j = \frac{\partial F(w)}{\partial w_j} = 2\sum_{i=1}^{N} e_i(w).\frac{\partial e_i(w)}{\partial w_j} \quad (5)$$

The gradient can be written as:

$$\nabla F(x) = 2J^T e(w) \quad (6)$$

Where

$$J(w) = \begin{bmatrix} \frac{\partial e_{11}}{\partial w_1} & \frac{\partial e_{11}}{\partial w_2} & \cdots & \frac{\partial e_{11}}{\partial w_N} \\ \frac{\partial e_{21}}{\partial w_1} & \frac{\partial e_{21}}{\partial w_2} & \cdots & \frac{\partial e_{21}}{\partial w_N} \\ \vdots & & & \\ \frac{\partial e_{KP}}{\partial w_1} & \frac{\partial e_{KP}}{\partial w_2} & \cdots & \frac{\partial e_{KP}}{\partial w_N} \end{bmatrix} \quad (7)$$

$J(w)$ is called the Jacobian matrix.

Then, the Hessian matrix is to be found. The k, j elements of the Hessian matrix yields as:

$$[\nabla^2 F(w)]_{kj} = \frac{\partial^2 F(w)}{\partial w_k \partial w_j} = 2\sum_{i=1}^{N}\left\{\frac{\partial e_i(w)}{\partial w_k}\frac{\partial e_i(w)}{\partial w_j} + e_i(w).\frac{\partial^2 e_i(w)}{\partial w_k \partial w_j}\right\} \quad (8)$$

The Hessian matrix can then be represented as follows:

$$\nabla^2 F(w) = 2J^T(W).J(W) + S(W) \quad (9)$$

$$S(w) = \sum_{i=1}^{N} e_i(w).\nabla^2 e_i(w) \quad (10)$$

If $S(w)$ is small assumed, the Hessian matrix can be approximated as:

$$\nabla^2 F(w) \cong 2J^T(w)J(w) \quad (11)$$

Using equations (3) and (10), the Gauss-Newton method is obtained as follows:

$$\begin{aligned} W_{k+1} &= W_k - [2J^T(w_k).J(w_k)]^{-1}2J^T(w_k)e(w_k) \\ &\cong W_k - [J^T(w_k).J(w_k)]^{-1}J^T(w_k)e(w_k) \end{aligned} \quad (12)$$

The advantage of Gauss-Newton is that it does not need computation of second derivatives. The problem in the Gauss-Newton method is the matrix $H = J^T J$ may not be invertible. This can be overcome by using the following modification.

Hessian matrix can be written as:

$$G = H + \mu I \quad (13)$$

Suppose that the eigen values and eigenvectors of H are $\{\lambda_1, \lambda_2, \ldots\ldots, \lambda_n\}$ and $\{z_1, z_2, \ldots\ldots, z_n\}$. Then:

$$\begin{aligned} Gz_i &= [H + \mu I]z_i \\ &= Hz_i + \mu z_i \\ &= \lambda_i z_i + \mu z_i \\ &= (\lambda_i + \mu)z_i \end{aligned} \quad (14)$$

As a result, the eigenvectors of G are the identical as the eigenvectors of H, and the eigen values of G are $(\lambda_i + \mu)$. The matrix G is positive definite by enhancing μ until $(\lambda_i + \mu) > 0$ for all i therefore the matrix will be invertible.
This leads to LM algorithm:

$$w_{k+1} = w_k - [J^T(w_k)J(w_k) + \mu I]^{-1}J^T(w_k)e(w_k) \quad (15)$$

$$\Delta w_k = [J^T(w_k)J(w_k) + \mu I]^{-1}J^T(w_k)e(w_k) \quad (16)$$

As known, learning parameter, μ is illustrator of steps of real output movement to preferred output. In the standard LM method, μ is a steady number. This work modifies LM method using μ as:

$$\mu = 0.01e^T e \quad (17)$$

Where e is a $k \times 1$ matrix therefore $e^T e$ is a 1×1 therefore $[J^T J + \mu I]$ is invertible.

Consequently, if actual output is far than the preferred output or similarly, errors are huge so, it converges to preferred output with large steps. Similarly, when the measurement of error is small then, the actual output approaches to the preferred output with soft steps. Therefore, error oscillation reduces significantly.

Modified LM algorithm for Learning is used for learning of the GHSOM -MLP which provides significant performance in clustering mixed data. The proposed Modified LM algorithm converges to desired output and moreover error oscillation also reduces greatly. Thus, an effective neural network approach is used for clustering mixed data using Modified LM Algorithm.

GHSOM-MLP with modified LM algorithm.

Step 1: Raw input data is given as input to the scaling layer of the hybrid GHSOM _MLP networks (First Layer).
Step 2: The scaled data is given as input to the GHSOM layer (Second and Third Layer).
Step 3: MLP layer has K nodes. MLP networks receive p inputs (Fourth Layer).
Step 4: MLP is trained using Modified Levenberg-Marquardt algorithm
Step 5: K_p nodes constitutes the input layer of a set of K MLP networks.
Step 6: Nodes in fourth layer is numbered as N_i. $i = 1,2,\ldots..K_p$.
Step 7: Nodes N_1 to N_p is the input of the first MLP M_1.
Step 8: Each MLP, M_i has two output nodes m and M.
Step 9: The outputs of this MLP-layer are then aggregated in layer seven which has just two nodes one for the minimum and the other for the maximum object.

Extended attribute-oriented induction

To trounce the drawback of major values and numeric attributes, an extension to the conventional AOI is proposed in this paper. This provides the ability of exploring the major values and a choice for processing numeric attributes. For the exploration of major values, a parameter majority threshold β is introduced. If some values (i.e., major values) take up a major portion (exceedingβ) of an attribute, the Extended AOI (EAOI) preserves those major values and generalizes other non major values. If no major values exist in an attribute, the EAOI proceeds like the AOI, generating the same results as that of the conventional approach. Furthermore, if β is set to 1, the EAOI degenerates to the AOI.

For solving the problems of constructing subjectively numeric concept hierarchies and generalizing boundary values, an alternative for processing numeric attributes is proposed: Users can choose to compute the average and deviation of the aggregated numeric values instead of generalizing those values to discrete concepts. Under this alternative, only categorical attributes are generalized. The average and deviation of numeric attributes of the merged tuples are calculated and then replace the original numeric values. The computed deviation reveals the dispersion of numeric values; the less the deviation is, the more concentrated the values are; otherwise, the more diversified the values are.
The EAOI algorithm is outlined as follows:

Algorithm: An extended attribute-oriented induction algorithm for major values and alternative processing of numeric attributes
Input: A relation W with an attribute set A; a set of concept hierarchies; generalization threshold θ, and majority threshold β.
Output: A generalized relation P.
Method:
1. Determine whether to generalize numeric attributes.
2. For each attribute Ai to be generalized in W,
2.1 Determine whether Ai should be removed, and if not, determine its minimum desired generalization level Li in its concept hierarchy.
2.2 Construct its major-value set M_i according to θ and β.
2.3 For $v \in Dom(Ai), if\ v \epsilon Mi$ construct the mapping pair as $(v, v_{Li} - M_{Li})$ otherwise, as (v,v).
3. Derive the generalized relation P by replacing each value v by its mapping value and computing other aggregate values.

In Step-1, if numeric attributes are not to be generalized, their averages and deviations will be computed in Step-3. Step-2 aims at preparing the mapping pairs of attribute values for generalization. First, in Step-2.1, an attribute is removed either because there is no concept hierarchy defined for the attribute, or its higher-level concepts are expressed in terms of other attributes. In Step-2.2, the attribute's major-value set Mi is constructed, which consists of the first α (<θ) count leading values if they take up a major portion (≥β) of the attribute, where θ is the generalization threshold that sets the maximum number of distinct values allowed in the generalized attribute.

In Step-2.3, if v is one of the major values, its mapping value remains the same, i.e., major values will not be generalized to higher-level concepts. Otherwise, v will be generalized by the concept at level Li by excluding the values contained in both the major-value set and the leaf set of the v_{Li} subtree (i.e., v_{Li} - M_{Li} where M_{Li} = Leaf(v_{Li})∩ M_i). Note that, if there are no major values in A_i, M_i and M_{Li} will be empty. Accordingly, the EAOI will

behave like the AOI. In Step-3, aggregate values are computed, including the accumulated count of merged tuples, which have identical values after the generalization, and the averages and deviations of numeric attributes of merged tuples if numeric attributes are determined not to be generalized.

EXPERIMENTAL RESULTS

The proposed clustering technique is experimented with UCI Adult Data Set. The data set contains 15 attributes that include eight categorical, six numerical, and one class attributes. 10, 000 tuples from the 48, 842 tuples are chosen randomly for the evaluation.

For the attribute choosing, the method of relevance analysis based on information gain is utilized. The relevance threshold was set to 0.1, and seven qualified attributes are obtained: Marital-status, Relationship, Education, Capital_gain, Capital_loss, Age, and Hours_per_week. The first three are categorical, and the others are numeric.

The map size is 400 units. The training parameters are set to the same with that of the previous experiment.

Table-1. Number of resultant clusters for using GHSOM and GHSOM-MLP with modified LM algorithm with different distance criteria.

	GHSOM-MLP with modified LM algorithm		GHSOM	
	Cluster	Outliers	Cluster	Outlier
d = 0	65	-	75	-
$d \leq 1.414$	5	-	9	-
$d \leq 2.828$	2	-	4	-
$d \leq 3 \& Adj$	3	7	5	5

The number of resultant clusters by using GHSOM and GHSOM-MLP with Modified LM Algorithm with different distance criteria is provided in Table-1 and Figure-3. It can be seen that the proposed technique results in better categorization, since it avoids unnecessary clusters and hence the number of clusters are considerably reduced.

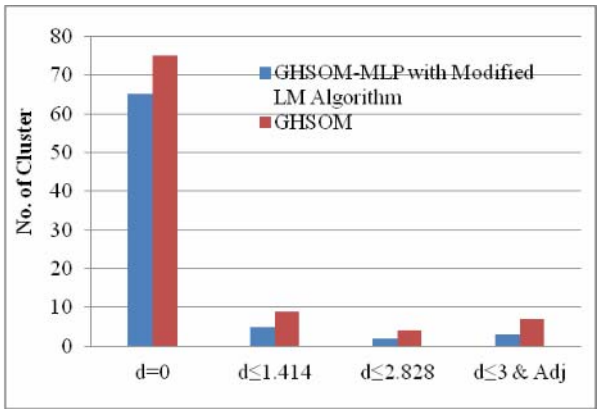

Figure-3. Number of resultant clusters for using GHSOM and GHSOM-MLP with modified LM algorithm with different distance criteria.

Mean square error

The mean square error (MSE) of the proposed GHSOM-MLP with Modified LM Algorithm is evaluated against the GHSOM. The MSE of the proposed approach are lesser than that of the GHSOM. The mean square error of the UCI Adult Data Set these two approaches are provided in Figure-4.

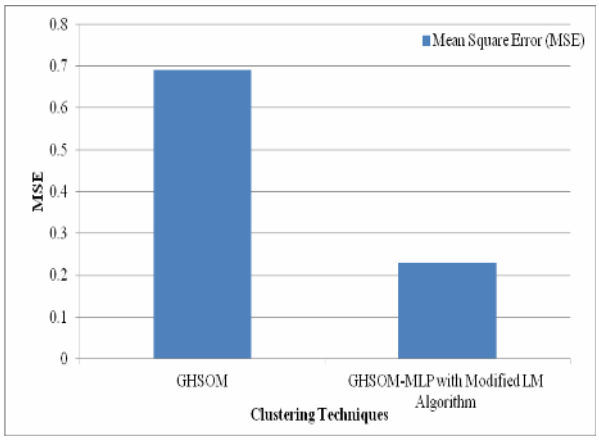

Figure-4. Comparison of mean square error.

CONCLUSIONS

The major intention of this paper is to propose an effective clustering technique to handle mixed category data. In fact, there exists a large numbers of mixed data clustering technique, but all those techniques does not provide better clustering results and has several limitations. In order to overcome these limitations, proposed a GHSOM-MLP with Modified LM Algorithm for clustering mixed data and it also uses Extended Attribute-Oriented Induction (EAOI). The experiment is performed on the UCI Adult Data Set and the result confirms that the better classification result is obtained for the proposed technique when compared to other existing approaches and also the mean square error of the proposed technique is very low.

REFERENCES

[1] K. Beyer, J. Goldstein, R. Ramakrishnan and U. Shaft. 1999. When is \nearest neighbor meaningful? Lecture Notes in Computer Science. 1540: 217-235.

[2] Hsu C.-C. 2006. Generalizing self-organizing map for categorical data. IEEE Transactions on Neural Networks. 17(2): 294–304.

[3] Hsu C.-C. and Wang S.-H. 2005. An integrated framework for visualized and exploratory pattern discovery in mixed data. IEEE Transactions on Knowledge and Data Engineering. 18(2): 161-173.

[4] Jain A. and Dubes R. 1988. Algorithms for clustering Data. Englewood Cliffs, NJ: Prentice-Hall.

[5] Rasmussen E. 1992. Clustering algorithms. In: William B. Frakes and Ricardo Baeza-Yates (Eds.), Information retrieval: Data structures and algorithms. Prentice Hall.

[6] Wiederhold G. and Foreword. 1996. In: Fayyad U., Shapiro G. P., Smyth P., Uthurusamy R. Editors Advances in Knowledge Discovery in Databases. California: AAAI/MIT Press.

[7] Halkidi M., Batistakis Y. and Vazirgiannis M. 2002. Cluster validity methods: Part I. SIGMOD Record (ACM Special Interest Group on Management of Data). 31(2): 40-45.

[8] Han J. and M. Kamber. 2006. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2nd Edition, New Delhi, India.

[9] M.K. Ng and J.C. Wong. 2002. Clustering Categorical Data Sets Using Tabu Search Techniques. Pattern Recognition. 35: 2783-2790.

[10] A. Hinneburg, C. C. Aggarwal and D. A. Keim. 2000. What Is the Nearest Neighbor in High Dimensional Spaces? In: Proceeding of 26th International Conference on Very Large Data Bases (VLDB'00). pp. 506-515.

[11] Hinneburg and D. A. Keim. 1999. Optimal Grid Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering. In: Proceeding of 25th International Conference on Very Large Data Bases (VLDB'99). pp. 506-517.

[12] T. Kohonen. 1997. The Self-Organizing Maps. 2nd ed. Berlin, Germany: Springer-Verlag.

[13] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero and A. Saarela. 2000. Self-Organization of a Massive Document Collection. IEEE Trans. Neural Networks. 11(3): 574-585.

[14] Amir Abolfazl Suratgar, Mohammad Bagher Tavakoli and Abbas Hoseinabadi. 2005. Modified Levenberg-Marquardt Method for Neural Networks Training. World Academy of Science, Engineering and Technology. 6: 46-48.

[15] Juha Vesanto and Esa Alhoniemi. 2000. Clustering of Self-Organizing Map. IEEE Transactions on Neural Networks. 11(3): 586-600.

[16] Tom Chiu, Dong Ping Fang, John Chen, Yao Wang and Christopher Jeris. 2001. A robust and scalable clustering algorithm for mixed type attributes in large database environment. International Conference on Knowledge Discovery and Data Mining. pp. 263-268.

[17] C.H. Cheng, A.W. Fu and Y. Zhang. 1999. Entropy-Based Subspace Clustering for Mining Numerical Data. Proc. ACM SIGMOD '99. pp. 84-93.

[18] M. Bouguessa, S. Wang and Q. Jiang. 2006. A K-Means-Based Algorithm for Projective Clustering. Proceeding of 18th IEEE International Conference on Pattern Recognition (ICPR '06). pp. 888-891.

[19] Mark Girolami. 2002. Mercer Kernel-based Clustering in Feature space. IEEE Transactions on Neural Networks. 13(3).

[20] Jian Yin, Zhi-Fang Tan, Jiang-Tao Ren and Yi-Qun Chen. 2005. An efficient clustering algorithm for mixed type attributes in large dataset. Proceedings of 2005 International Conference on Machine Learning and Cybernetics. 3: 1611-1614.

[21] Bello M. G. 1992. Enhanced training algorithms, and integrated training/architecture selection for multi layer perceptron networks. IEEE Trans. on Neural Net. 3: 864-875.

[22] D.W. Ruck, S. K. Rogers and M. Kbrisky. 1990. Feature selection using a multilayer perceptron. J. Neural Network Comput. 2(2): 40-48.