A ROUGH SET APPROACH FOR CUSTOMER SEGMENTATION

Prabha Dhandayudam [*] ***and Ilango Krishnamurthi***

Department of CSE, Sri Krishna College of Engineering and Technology, Coimbatore, India
**Email:* prabhadhandayudam@gmail.com
Email: ik@skcet.ac.in

ABSTRACT

Customer segmentation is a process that divides a business's total customers into groups according to their diversity of purchasing behavior and characteristics. The data mining clustering technique can be used to accomplish this customer segmentation. This technique clusters the customers in such a way that the customers in one group behave similarly when compared to the customers in other groups. The customer related data are categorical in nature. However, the clustering algorithms for categorical data are few and are unable to handle uncertainty. Rough set theory (RST) is a mathematical approach that handles uncertainty and is capable of discovering knowledge from a database. This paper proposes a new clustering technique called MADO (Minimum Average Dissimilarity between Objects) for categorical data based on elements of RST. The proposed algorithm is compared with other RST based clustering algorithms, such as MMR (Min-Min Roughness), MMeR (Min Mean Roughness), SDR (Standard Deviation Roughness), SSDR (Standard deviation of Standard Deviation Roughness), and MADE (Maximal Attributes DEpendency). The results show that for the real customer data considered, the MADO algorithm achieves clusters with higher cohesion, lower coupling, and less computational complexity when compared to the above mentioned algorithms. The proposed algorithm has also been tested on a synthetic data set to prove that it is also suitable for high dimensional data.

Keywords: Customer Relationship Management, Customer segmentation, Clustering, Categorical data, Rough set theory

1 INTRODUCTION

Customer Relationship Management (CRM) is a business methodology used to build a relationship with long term profitable customers by analyzing customer needs and behaviors. It is an important technology in every business because business is customer centric. CRM helps business leaders gain insight into customer behavior and life time value to increase profit by acting according to the customer characteristics. Customer segmentation plays an important role in CRM. It divides customers into groups according to their purchasing behavior, allowing business leaders to design and establish different strategies for each group of customers and thus maximize their value (Ling & Yen, 2001). Recency (R), Frequency (F), and Monetary (M) are the attributes chosen for describing the purchasing customer characteristics. The RFM model works very effectively in customer segmentation (Wu & Lin, 2005). R indicates the time interval between the present and previous transaction dates of a customer. F indicates the number of transactions that the customer has made in a particular interval of time. M indicates the total value of the customer's transaction (Cheng & Chen, 2009). In this paper, the modified RFM model, called RFMP, is introduced. This model considers the customers' payment details. P indicates the average time interval between payment and purchase date. The customers' payment details are an important attribute because any two customers with the same R, F, M values but a different P value cannot be treated equally by the enterprise. The RFMP model ensures that the customer segmentation is done objectively. The values for R, F, M, and P attributes are continuous. These continuous values are normalized to categorical values as being very low, low, middle, high, and very high for effective analysis. The data available for customer segmentation is now categorical data. The data mining clustering technique is widely used to accomplish customer segmentation (Ngai, Xiu, & Chau, 2009).

The clustering technique is used to segment customers in such a way that the customers in one group behave similarly when compared to the customers in other groups based on their transaction details. The traditional approaches for clustering, such as partitioning and hierarchical algorithms, deal with numerical data whose inherent geometric properties can be exploited to naturally define distance functions between data points. Distance functions, such as Manhattan, Euclidean, and Minkowski, are used for allocating a data point to the appropriate clustering. However, customer data available for clustering is categorical so the above procedures are not feasible. The computation of similarity or dissimilarity is essential for categorical data (Chen, Chuang, & Chen 2008).

Simple matching, co-occurrence, probabilistic and distance hierarchy are the approaches for computing similarity or dissimilarity measures (Cao, Liang, Li, Bai, & Dang, 2011).

Simple matching is a common approach in which the comparison of two identical categorical values yields either zero or one. k-Modes, fuzzy k-Modes, and k-prototype algorithms are based on simple matching. These algorithms produce clusters with weak intrasimilarity and have stability problems (Cao, Liang, Li, Bai, & Dang, 2011). ROCK (Robust clustering using links) and CACTUS (Clustering categorical data using summaries) are algorithms based on the co-occurrence approach. ROCK uses the concept of a link to measure the similarity between categorical patterns. Here link is defined as the number of common neighbors between two patterns (Guha, Rastogi, & Shim, 2000). CACTUS calculates the frequency of two values appearing in the patterns together. It finds clusters in subsets of all attributes and thus performs subspace clustering. It generalizes the cluster definition of numerical data so that it is suitable for categorical data (Ganti, Gehrke, & Ramakrishnan, 1999). The limitations of ROCK are that it is sensitive to threshold value and sometimes does not produce the required number of clusters (Parmar, Wu, & Blackhurst, 2007). Probabilistic approaches use conditional probability estimation to define relations between clusters. COBWEB, AUTOCLASS, DECA, and COOLCAT are based on probabilistic models and require a long training time. COBWEB uses the category utility function, and AUTOCLASS uses a Bayesian method to derive probable class distribution. DECA is a discrete valued clustering algorithm, and COOLCAT is an entropy based algorithm. Distance hierarchy associates each link with a weight and requires the domain experts to incorporate knowledge (Cao, Liang, Li, Bai, & Dang, 2011). Rough set theory (RST) by Pawlak (1982) received a great deal of attention for dealing with categorical data in clustering algorithms due to its stable results and no requirement of domain expertise. It uses global data properties to establish similarity between the objects (Bean & Kambhampati, 2008).

The existing clustering algorithms (Parmar et al., 2007; Mazlack, He, Zhu, & Coppock, 2000; Kumar & Tripathy, 2009; Tripathy & Ghosh, 2011a; Tripathy & Ghosh, 2011b; Herawan, Deris, & Abawajy, 2010; Herawan, Ghazali, Yanto, & Deris, 2010) based on RST utilize the correlation of attributes. If the attributes are dependent, then the clustering algorithms based on the correlation of attributes produce correct results. The attributes R, F, M, and P chosen for describing the purchasing characteristics of customers are independent so that there is no proportional relationship between the attributes. This concept led to the proposal of MADO (Minimum Average Dissimilarity between Objects), a clustering algorithm based on RST. The MADO algorithm calculates the dissimilarity between objects without considering the dependency between attributes.

The rest of the paper is organized as follows: Section 2 discusses the basic concepts of rough set theory. Section 3 summarizes rough set theory based clustering algorithms. Section 4 explains the MADO clustering algorithm. Section 5 compares the clustering results obtained for real customer data and synthetic data. Finally, Section 6 provides concluding remarks.

2 BACKGROUND

Rough set theory (RST) by Pawlak classifies imprecise, uncertain, or incomplete information or knowledge expressed by data acquired from experience (Pawlak, 1982). It gained importance in the areas of machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, decision support systems, inductive reasoning, and pattern recognition (Pawlak, 1992). It is suitable for processing qualitative information that is difficult to analyze by standard statistical techniques. It manages vague and uncertain data or problems related to information systems (Shyng, Wang, Tzeng, & Wu, 2007).

The information system is a 4-tuple (quadruple) $S = (U, A, V, f)$, where U and A are a non-empty finite sets of objects and attributes, respectively, and V is the set containing the domain of each attribute, where V_a denotes the domain of attribute a that belongs to A. The function $f: U \times A \rightarrow V$ is a total function such that $f(u, a) \in V_a$, for every $(u, a) \in U \times A$ and is called the information function. RST is mainly based on the indiscernibility relation, equivalence class, lower approximation, and upper approximation (Pawlak & Skowron, 2007).

The indiscernibility relation $(Ind(B))$ is a relation on U. Given two objects, $x_i, x_j \in U$, they are indiscernible by the set of attributes B in A if and only if $a(x_i) = a(x_j)$ for every $a \in B$. That is, $(x_i, x_j) \in Ind(B)$ if and only if $\forall a \in B$ where $B \subseteq A$ and $a(x_i) = a(x_j)$. The equivalence class $([x_i]_{Ind(B)})$ is the set of objects x_i having the same values for the set of attributes in B. This is also known as an elementary set with respect to B. The lower approximation (X_{LB}) is the union of all the elementary sets with respect to B that are contained in X. The upper approximation (X_{UB}) is the

union of all the elementary sets with respect to *B* that have a non-empty intersection with *X*. Eqs. (1) and (2) calculate the lower and upper approximation, respectively.

$$X_{LB} = \{x_i \mid [x_i]_{Ind(B)} \subseteq X\} \quad (1)$$

$$X_{UB} = \{x_i \mid [x_i]_{Ind(B)} \cap X \neq \varphi\} \quad (2)$$

The lower approximation consists of all objects that definitely belong to the concept while the upper approximation contains all objects that possibly belong to the concept. The difference between the upper and the lower approximation constitutes the boundary region of the vague concept. Approximations are two basic operations in rough set theory; thus it expresses vagueness not by means of membership but by employing a boundary region of a set. If the boundary region of a set is empty, the set is crisp. Otherwise the set is rough (inexact) (Pawlak & Skowron, 2007). The ratio of the cardinality of the lower approximation and the cardinality of the upper approximation is defined as the accuracy of estimation, a measure of roughness (Pawlak, 1992). Many clustering algorithms based on RST have been developed, and they are overviewed in the next section.

3 RELATED WORK

Mazlack et al. (2000) proposed two techniques based on rough set theory to select clustering attributes. These are bi-clustering (BC) and total roughness (TR) techniques. BC is applicable only for bi-valued attributes. TR handles multi-valued attributes and is based on the total average of the mean roughness of an attribute with respect to the set of all attributes in an information system. The attribute with the higher TR is chosen as the clustering attribute. However, for partitioning, the method starts with binary valued attributes and uses the total roughness criterion only for multi-valued attributes. Therefore, partitioning is done on a multi-valued attribute only when all the binary valued attributes have already been partitioned. This reduces the efficiency of the algorithm because the partitioning is done on a binary valued attribute even when the total roughness value for the multi-valued attribute is high. The roughness, mean roughness, and total roughness in TR are calculated using Eqs. (3), (4), and (5), respectively.

$$R(X) = \frac{|X_{LB}|}{|X_{UB}|} \quad (3)$$

$$\text{Rough(i)} = \frac{\sum_{i=1}^{n} R(X_i)}{n} \quad (4)$$

$$\text{Total roughness(i)} = \frac{\sum_{i=1}^{m} \text{Rough(i)}}{m} \quad (5)$$

X indicates the set of objects in the data set; X_i indicates the set of objects in the sub-partition of attribute *I*; *n* indicates the number of sub-partitions of attribute *I*, and *m* indicates the number of attributes. In Eq. (3), X_{LB} and X_{UB} are calculated using Eqs. (1) and (2), respectively. In Eq. (4), Rough(i) is the mean roughness of all sub-partitions of attribute i. In order to choose the partitioning attribute i, the Total roughness(i) towards all the attributes is calculated using Eq. (5).

The attribute with the highest total roughness is chosen as the clustering attribute. However, for partitioning, the method starts with binary valued attributes and uses the total roughness criterion only for multi-valued attributes. This creates a disadvantage due to the fact that the partitioning is done on a binary attribute even though the total roughness for a multi-valued attribute is higher.

Parmar et al. (2007) proposed a new technique called min-min roughness (MMR) for multi-valued attributes. The MMR technique is based on the minimum value of mean roughness and does not require total roughness for calculating as in TR. The roughness and mean roughness in MMR is calculated using Eqs. (6) and (7), respectively.

Given a_i, $a_j \in A$ (set of attributes), $V(a_i)$ is the set of values of attributes a_i, and then $a_i \neq a_j$. *X* is a subset of objects having one specific value α for the attribute a_i,, that is X ($a_i = \alpha$). $(X_{a_j}(a_i = \alpha))_L$ is the lower approximation of

X with respect to $\{a_j\}$, and $(X_{a_j}(a_i = \alpha))_U$ is the upper approximation of *X* with respect to $\{a_j\}$. Thus, $R_{a_j}(X)$ is defined as the roughness of X with respect to $\{a_j\}$, which is given by Eq. (6). Also, the mean roughness on attribute a_i with respect to $\{a_j\}$ is defined in Eq. (7).

$$R_{a_j}(X \mid a_i = \alpha) = 1 - \frac{\left|(X_{a_j}(a_i = \alpha))_L\right|}{\left|(X_{a_j}(a_i = \alpha))_U\right|} \tag{6}$$

$$Rough_{a_j}(a_i) = \frac{R_{a_j}(X \mid a_i = \alpha_1) + \ldots + R_{a_j}(X \mid a_i = \alpha_{|V(a_i)|})}{|V(a_i)|} \tag{7}$$

The maximum value of roughness is one because the number of objects in the lower approximation is less than or equal to the number of objects in the upper approximation. The min-roughness (MR) of each attribute refers to the minimum of the mean roughness with respect to a single attribute. The min-min-roughness (MMR) is defined as the minimum MR of *n* attributes. The MMR value determines the splitting attribute. The node with the maximum number of elements is chosen for further splitting. From the experimental analysis in Parmar et al. (2007), MMR achieves better results when compared to BC, TR, fuzzy set based algorithms, and other dissimilarity approaches.

Kumar and Tripathy (2009) proposed MMeR by making a slight alteration in MMR. In their algorithm, the mean-roughness (MeR) of each attribute is calculated as the mean of the mean roughness. Then min-mean-roughness (MMeR) is defined as the minimum of MeR of n attributes. The MMeR value determines the splitting attribute. The node that has the maximum average distance between its elements is chosen for further splitting.

Tripathy and Gosh (2011a) proposed an algorithm for clustering categorical data where the standard-deviation-roughness (SDR) of each attribute is calculated as the standard deviation of the mean roughness. The attribute with the minimum SDR is chosen as the splitting attribute.

Tripathy and Gosh (2011b) also proposed an algorithm called SSDR. Here the standard deviation of SDR (SSDR) is calculated, and the splitting attribute is chosen based on this value. The node chosen for further splitting in the SDR and SSDR algorithm is the same as that of the MMeR algorithm. SSDR achieves better results than SDR for small data sets while for large data sets it achieves the same results as SDR. The roughness used in MMeR, SDR, and SSDR is calculated using the same formula as in MMR. All the algorithms TR, MMR, MMeR, SDR, and SSDR in the above examples (Parmar et al., 2007; Mazlack et al., 2000; Kumar & Tripathy, 2009; Tripathy & Ghosh, 2011a; Tripathy & Ghosh, 2011b) have been tested on data sets available in the UCI machine learning repository. The purity of clusters was used as a measure to test the quality of the clusters. The purity ratios of MMR, MMeR, SDR, and SSDR are in increasing order. All these clustering algorithms are based on the roughness of an attribute with other attributes.

Herawan et al. (2010a) proposed a new technique called maximal dependency attributes (MDA) for selecting clustering attributes. Based on MDA, Herawan et al. (2010b) proposed MADE for categorical data clustering. MDA selects the clustering attribute by determining the dependencies between attributes. The attribute with the maximum degree of dependency is selected as the partitioning attribute.

4 PROPOSED ALGORITHM

The calculation of roughness and determination of dependencies between attributes in real customer data are not appropriate due to the dynamic behavior of the customers. The MADO clustering algorithm overcomes this disadvantage by utilizing the equivalence class property of rough set theory. The splitting attribute is chosen based on the dissimilarity between the objects in the same equivalence class. Let *A* indicate the set containing the attributes in the data and *B* be another set containing the objects whose value for a particular attribute is same. Equivalence class is calculated for each attribute with a specific value. In each calculation, the set *B* contains the objects of that equivalence class. The dissimilarity between the objects within the set *B* is calculated using Eq. (8). The average dissimilarity between the objects in an equivalence class or set *B* is calculated using Eq. (9).

$$\text{dissim}(x_i, x_j) = 1 - \frac{\left|V(x_i, x_j)\right|}{n} \quad (8)$$

Here x_i, x_j belong to the same equivalence class; n is the number of attributes in A; and $\left|V(x_i, x_j)\right|$ is the number of attributes having same value for the objects x_i, x_j.

$$\text{avgdissim}(B) = \frac{\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \text{dissim}(x_i, x_j)}{m(m-1)/2} \quad (9)$$

Here B indicates an equivalence class; m is the number of objects in set B, and x_i, x_j belongs to set B. The equivalence class B of an attribute a_i having value v_j is represented as $[a_{ij}]$. Set B, which has the minimum average dissimilarity and has at least two objects, determines the splitting attribute a_i on its value v_j for the parent node. Initially, the parent node contains all the objects. After partition, the leaf node having more objects is selected as the parent node for further partitioning in subsequent iterations. The algorithm terminates when it reaches a pre-defined number of clusters. The procedure for the MADO clustering algorithm is given in Figure 1.

```
Procedure (U,k)
Line 1    Begin
Line 2    Set current number of cluster CNC = 1
Line 3  Set k as required number of clusters
Line 4    Set ParentNode = U
Line 5    Do
Line 6       For each a_i from A (i = 1 to n, where n is the number of attributes in A)
Line 7             For j=1 to l, where l is the number of different values in a_i
Line 8                   In the ParentNode, determine family of equivalence classes for a_i with
                       value j which is denoted as set B
Line 9                      Calculate avgdissim (B)
Line 10            Next
Line 11     Next
Line 12     Set Min-Avg-Dissim = Min (avgdissim (B)) for each |B|>1
Line 13     Determine splitting attribute a_i on its value v_j (a_ij) corresponding to the Min-Avg-Dissim
Line 14     CNC = CNC + 1
Line 15     ParentNode = ProcParentNode (CNC)
Line 16 While (CNC<k)
Line 17 End
Line 18 ProcParentNode (CNC)
Line 19 Begin
Line 20  For i = 1 to CNC
Line 21            Size (i) = Count (Set of Elements in Cluster i)
Line 22  Next
Line 23  Determine Max (Size (i))
Line 24  Return (Set of Elements in cluster i) corresponding to Max (Size (i))
Line 25 End
```

Figure 1. MADO algorithm

5 EXPERIMENTAL RESULTS

In this section, real data sets of customer transaction details are used for clustering or segmenting the customers. Customer transaction details for a period of six months have been collected from four different enterprises. Data set1 consists of 47891 records; data set2 consists of 29790 records; data set3 consists of 34035 records; and data set4 consists of 24191 records. For each transaction, party id, date of purchase, amount of purchase, and payment of purchase are used to define R, F, M, and P values. The distinct party id represents the individual customer. For each distinct party id, R is calculated as the interval between the time that the latest consuming behavior occurred and the present; F is calculated as the number of transaction records; M is calculated as the total purchase amount; and P is calculated as the average time interval (in terms of days) between the payment date and purchase date for

each transaction in the data set. The data set now has only four attributes, namely R, F, M, and P, for each customer. Dataset1 has 5062 customers; data set2 has 1420 customers; data set3 has 3811 customers; and data set4 has 2675 customers. The values of R, F, M, and P are normalized as given below:
For normalizing R or P:

1) Sort the data set in descending order of R or P.
2) Divide the data set into five equal parts with 20% of the records in each.
3) Assign categorical values as very low, low, middle, high, and very high to the first, second, third, fourth, and fifth parts of the records, respectively.

For normalizing F or M:

1) Sort the data set in ascending order of F or M
2) Divide the data set into five equal parts with 20% of the records in each.
3) Assign categorical values as very low, low, middle, high, and very high to first, second, third, fourth, and fifth part of the records, respectively.

The normalized data set is now used by the MADO clustering algorithm to segment the customers into various groups. MMR, MMeR, SDR, SSDR, and MADE algorithms are applied to the same four data sets so that the customers are divided into various groups. Criteria cohesion and coupling are used to measure the internal quality of the cluster (Santos, Heuser, Moreira, & Wives, 2011). Cohesion expresses the average similarity between the elements of a cluster. Coupling expresses the average similarity between all pairs of elements, where one element belongs to cluster C and the other does not. Ideally, cohesion should be high and coupling should be low (Kunz & Black, 1995). The formulas for cohesion and coupling for a cluster C are given by Eqs. (10) and (11), respectively. The total cohesion and total coupling of clusters are given by Eqs. (12) and (13), respectively.

$$\text{Cohesion}(C) = \frac{\sum_{i>j} \text{sim}(c_i, c_j)}{\frac{m*(m-1)}{2}} \tag{10}$$

$$\text{Coupling}(C) = \frac{\sum_{i,j} \text{sim}(c_i, q_j)}{m*n} \tag{11}$$

$$\text{Total cohesion} = \sum_{C=1}^{k} \text{Cohesion}(C) \tag{12}$$

$$\text{Total coupling} = \sum_{C=1}^{k} \text{Coupling}(C) \tag{13}$$

Here $\text{sim}(c_i,c_j)$ is the similarity score between elements c_i and c_j belonging to cluster C; $\text{sim}(c_i,q_j)$ is the similarity between element c_i from cluster C and element q_j from another cluster; m is the number of elements in C; n is the number of elements outside C; and k is the number of clusters. The results of MMR, MMeR, SDR, SSDR, MADE, and MADO algorithms are compared by varying the number of clusters to be produced from three to seven. In each case, the total coupling and the total cohesion of the clusters are calculated using Eqs. (12) and (13). The results of four data sets for all the five cases produced by the clustering algorithms are tabularized in Tables 1 through 8. The MADE algorithm produces the value zero for the degree of dependency calculated for each attribute. This is because the lower approximation of each attribute for each value is a null set. Therefore this algorithm could not be applied further to produce the clustering result for the considered data set. The results of SDR and SSDR are the same because for large data sets, SSDR achieves the same result as SDR (Tripathy & Ghosh, 2011b).

Table 1. Total cohesion value for data set1

Total cohesion	Number of clusters				
	3	4	5	6	7
MADO	1.254619	1.78055	2.278677	2.827566	3.493666
MMR	1.13443	1.595834	2.154488	2.818146	3.280673
MMeR	1.134043	1.595834	2.154488	2.782483	3.462215
SDR & SSDR	1.134043	1.595834	2.154488	2.782483	3.462215

Table 2. Total coupling value for data set1

Total coupling	Number of clusters				
	3	4	5	6	7
MADO	0.40142	0.557465	0.713996	0.902171	1.09174
MMR	0.407938	0.557517	0.739137	0.905447	1.099329
MMeR	0.407938	0.557517	0.739137	0.925793	1.113571
SDR & SSDR	0.407938	0.55717	0.739137	0.925793	1.113571

Table 3. Total cohesion value for data set2

Total cohesion	Number of clusters				
	3	4	5	6	7
MADO	1.379485	1.901231	2.499687	3.188883	3.85792
MMR	1.191954	1.680294	2.274037	3.026072	3.71652
MMeR	1.191954	1.680294	2.391836	2.985578	3.730305
SDR & SSDR	1.191954	1.680294	2.391836	2.985578	3.730305

Table 4. Total coupling value for data set2

Total coupling	Number of clusters				
	3	4	5	6	7
MADO	0.412799	0.573042	0.734237	0.947111	1.151452
MMR	0.436303	0.613043	0.790121	0.972689	1.18822
MMeR	0.436303	0.613043	0.81106	0.988138	1.184198
SDR & SSDR	0.436303	0.613043	0.81106	0.988138	1.184198

Table 5. Total cohesion value for data set3

Total cohesion	Number of clusters				
	3	4	5	6	7
MADO	1.249195	1.742696	2.296043	2.938652	3.580393
MMR	1.115192	1.576964	2.13819	2.867371	3.557482
MMeR	1.115192	1.576964	2.13819	2.83356	3.461231
SDR & SSDR	1.115192	1.576964	2.13819	2.83356	3.461231

Table 6. Total coupling value for data set3

Total coupling	Number of clusters				
	3	4	5	6	7
MADO	0.38837	0.527531	0.678554	0.872727	1.065775
MMR	0.409246	0.563443	0.719205	0.901638	1.090394
MMeR	0.409246	0.563443	0.719205	0.902418	1.088859
SDR & SSDR	0.409246	0.563443	0.719205	0.902418	1.088859

Table 7. Total cohesion value for data set4

Total cohesion	Number of clusters				
	3	4	5	6	7
MADO	1.277666	1.821888	2.33417	2.855034	3.577055
MMR	1.115602	1.570354	2.129685	2.834838	3.479356
MMeR	1.115602	1.570354	2.129685	2.848724	3.376209
SDR & SSDR	1.115602	1.570354	2.129685	2.848724	3.376209

Table 8. Total coupling value for data set4

Total coupling	Number of clusters				
	3	4	5	6	7
MADO	0.375785	0.551788	0.71688	0.905408	1.094585
MMR	0.413905	0.566269	0.730398	0.907193	1.100851
MMeR	0.413905	0.566269	0.730398	0.917279	1.103247
SDR & SSDR	0.413905	0.566269	0.730398	0.917279	1.103247

Tables 1 through 8 illustrate that the clusters produced by the MADO algorithm have on average high cohesion and low coupling values for all five cases with respect to the other algorithms. Because the clustering algorithm performance depends on producing clusters with high cohesion and low coupling values, the proposed algorithm out performs the other rough set based clustering algorithms for the considered four real customer data sets. The reason behind this is that the proposed algorithm considers the dissimilarity between the objects in the same equivalence class when choosing the splitting attribute instead of finding the correlation between attributes as the other rough set based clustering algorithms do.

The proposed clustering algorithm is also applicable for high dimensional data. It has been tested for soyabean and zoo data sets obtained from the UCI Machine Learning Repository. The soybean data set contains 47 objects with 35 categorical attributes. The zoo data set contains 101 objects with 18 categorical attributes. The purity of clusters is used to test the quality of the clusters if the class label is already known. In a real data set, because the class label is not known, cohesion and coupling are used to test the quality of the clusters. In the synthetic data set obtained from the repository, the class label is available in the data set and so purity is used as a measure to test the quality of the clusters. The purity of a cluster 'i' is defined by Eq. (14). The overall purity is defined by Eq. (15).

$$Purity(i) = \frac{n_r^i}{n_r} \tag{14}$$

$$Overall\ purity = \frac{\sum_{i=1}^{n} purity(i)}{n} \tag{15}$$

Here, n_r^i indicates the numbers of objects occurring in cluster i and its corresponding class r; n_r indicates the number of objects in the class r; and n indicates the number of clusters in the data set. A higher value of overall purity indicates a better clustering result, with perfect clustering yielding a value of 1. In the zoo data set, each object is classified as belonging to one of the 7 classes. The MMR, MMeR, SDR, SSDR, and proposed MADO clustering algorithms are executed to obtain 7 classes. Purity for each cluster is calculated using Eq. (14), and the overall purity is calculated using Eq. (15). The results obtained from the MMR, SSDR, and MADO algorithms are given in Tables 9, 10, and 11, respectively. From Table 9, it is observed that out of 101 objects, 92 objects are classified correctly. From Table 10, it is observed that out of 101 objects, 79 objects are classified correctly. From Table 11, it is observed that out of 101 objects, 95 objects are classified correctly. Thus the overall purity of the clusters obtained using MMR, SSDR, and the proposed algorithm is 0.91, 0.9079, and 0.9406 respectively. The overall purity of the clusters obtained using MMeR and SDR is 0.902 and 0.9079 respectively (Tripathy & Ghosh, 2011b). Thus, the proposed clustering algorithm produces good clusters for a high dimensional synthetic data set.

Table 9. MMR output for the zoo data set

Cluster Number	C1	C2	C3	C4	C5	C6	C7	Purity
1	0	0	3	0	3	0	0	0.50
2	39	0	0	0	0	0	0	1
3	0	0	1	0	1	0	0	0.50
4	0	0	1	13	0	0	0	0.93
5	0	0	0	0	0	2	10	0.83
6	2	0	0	0	0	6	0	0.75
7	0	20	0	0	0	0	0	1

Table 10. SSDR output for the zoo data set

Cluster Number	C1	C2	C3	C4	C5	C6	C7	Purity
1	19	3	0	0	0	0	2	0.7916
2	0	0	0	13	0	0	0	1
3	0	0	0	0	0	8	0	1
4	0	0	0	0	4	0	0	1
5	0	0	5	0	0	0	0	1
6	0	8	0	0	0	0	0	1
7	22	9	0	0	0	0	8	0.5641

Table 11. MADO output for the zoo data set

Cluster Number	C1	C2	C3	C4	C5	C6	C7	Purity
1	0	20	0	0	0	0	0	1
2	41	0	0	0	0	0	0	1
3	0	0	0	0	0	4	0	1
4	0	0	0	0	0	2	0	1
5	0	0	0	13	0	0	0	1
6	0	0	0	0	0	2	10	0.833
7	0	0	5	0	4	0	0	0.555

In the zoo data set, the MMR algorithm performs better when compared to the MMeR, SDR, and SSDR algorithms. Next, the MMR and proposed clustering algorithms are used for the soybean data set. In the soybean data set, each object is classified as belonging to one of the 4 diseases or 4 classes. The MMR and proposed clustering algorithms are executed to obtain 4 clusters. Purity for each cluster is calculated using Eq. (14). The results obtained from the MMR and the proposed algorithm are given in Tables 12 and 13, respectively. From Table 12, it is observed that out of 47 objects, 39 objects are classified correctly. From Table 13, it is observed that out of 47 objects, 46 objects are classified correctly. Thus the overall purity of the clusters obtained using the MMR and the proposed algorithm is 0.8298 and 0.9787, respectively.

Table 12. MMR output for soyabean data set

Cluster Number	Disease 1	Disease 2	Disease 3	Disease 4	Purity
1	0	10	0	0	1
2	10	0	0	0	1
3	0	0	8	17	0.68
4	0	0	2	0	1

Table 13. MADO output for soybean data set

Cluster Number	Disease 1	Disease 2	Disease 3	Disease 4	Purity
1	0	10	0	0	1
2	10	0	0	0	1
3	0	0	9	0	1
4	0	0	1	17	0.944

The results obtained for the synthetic data set show that the proposed algorithm produces improved clustering results in terms of purity when compared to other clustering algorithms. Therefore, the proposed clustering algorithm can cluster data in a better way irrespective of the number of attributes considered.

6 CONCLUSION

Customer segmentation for CRM is achieved using a data mining clustering technique. This technique is tested in four various real data sets. Data related to customers are categorical in nature so the usual hierarchical and partitioning algorithms are not applicable. The importance of rough set theory for categorical clustering is discussed, and the algorithms based on this technique are studied. The proposed MADO algorithm considers the dissimilarity between objects in the same equivalence class. The cluster quality for a real data set is measured using cohesion and coupling. The experimental results for the considered four customer data sets show that the MADO algorithm produces clusters with high cohesion and low coupling values for all four cases with respect to other algorithms. The applicability of the MADO algorithm to high dimensional data sets has been proven by testing it with synthetic data sets. The experimental results show that the MADO algorithm clusters high dimensional data in a better way when compared to other clustering algorithms. In the future, the behavior or characteristics of customers in each segment can be analyzed so that marketers can employ different strategies for each group. Furthermore, the life time value of a customer can be increased by adopting suitable techniques for each customer segment.

7 REFERENCES

Bean, C. & Kambhampati, C. (2008) Autonomous clustering using rough set theory. *International Journal of Automation and Computing, 5*(1), 90-102.

Cao, F., Liang, J., Li, D., Bai, L., & Dang, C. (2011) A dissimilarity measure for the k-Modes clustering algorithm. *Knowledge Based Systems, 26*, 120-127.

Chen, H., Chuang, K., & Chen, M. (2008) On data labeling for clustering categorical data. *IEEE Transactions on Knowledge and Data Engineering, 20*(11), 1458-1471.

Cheng, C. & Chen, Y. (2009) Classifying the segmentation of customer value via RFM model and RS theory. *Expert Systems with Applications, 36*(3), 4176-4184.

Ganti, V., Gehrke, J., & Ramakrishnan, R. (1999) CACTUS-Clustering categorical data using summaries. *Proceedings of 5th ACM SIGKDD International conference on Knowledge Discovery and Data Mining* (pp. 73-83). San Diego, CA, USA.

Guha, S., Rastogi, R., & Shim, K. (2000) ROCK: A robust clustering algorithm for categorical attributes. *Information Systems, 25*(5), 345-366.

Herawan, T., Deris, M.M., & Abawajy, J.H. (2010) A rough set approach for selecting clustering attribute. *Knowledge Based Systems, 23*(3), 220-231.

Herawan, T., Ghazali, R., Yanto, I.T.R, & Deris, M.M. (2010) Rough set approach for categorical data clustering. *International Journal of Database Theory and Application, 3*(1), 33-52.

Kumar, P. & Tripathy, B.K. (2009) MMeR: an algorithm for clustering heterogeneous data using rough set theory. *International Journal of Rapid Manufacturing, 1*(2), 189-207.
Kunz, T. & Black, J.P. (1995) Using automatic process clustering for design recovery and distributed debugging. *IEEE Transactions on Software Engineering, 21*(6), 515–527.

Ling, R. & Yen, D.C. (2001) Customer relationship management: An analysis framework and implementation strategies. *Journal of Computer Information Systems, 41*(3), 82–97.

Mazlack, L.J., He, A., Zhu, Y., & Coppock, S. (2000) A rough set approach in choosing clustering attributes. *Proceedings of the 13th international conference on Computer Applications in Industry and Engineering* (pp. 1-6). Honolulu, Hawaii, USA.

Ngai, E.W.T., Xiu, L., & Chau, D.C.K. (2009) Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications, 36*(2), 2592-2602.

Parmar, D., Wu, T., & Blackhurst, J. (2007) MMR:an algorithm for clustering categorical data using rough set theory. *Data and Knowledge Engineering, 63*(3), 879-893.

Pawlak, Z. (1982) Rough set. *International Journal of Computer and Information Sciences, 11*(5), 341-356.

Pawlak, Z. (1992) *Rough sets: Theoretical aspects of reasoning about data*, Norwell, MA, USA: Kluwer Academic Publishers.

Pawlak, Z. & Skowron, A. (2007) Rudiments of rough sets. *Information Sciences, 177*(1), 3 – 27.

Santos, J.B., Heuser, C.A., Moreira, V.P., & Wives, L.K. (2011) Automatic threshold estimation for data matching applications. *Information Sciences, 181*(13), 2685-2699.

Shyng, J.Y., Wang, F.K., Tzeng, G.H., & Wu, K.S. (2007) Rough set theory in analyzing the attributes of combination values for the insurance market. *Expert Systems with Applications, 32*(1), 56–64.

Tripathy, B.K. & Ghosh, A. (2011) SDR: An algorithm for clustering categorical data using rough set theory. *Proceedings of IEEE conference on Recent Advances in Intelligent Computational Systems* (pp. 867-872). Trivandrum, India.

Tripathy, B.K. & Ghosh, A. (2011) SSDR: An algorithm for clustering categorical data using rough set theory. *Advances in Applied Science Research, 2*(3), 314-326.

Wu, J. & Lin, Z. (2005) Research on Customer Segmentation Model by Clustering. *Proceedings of the ACM 7th international conference on Electronic commerce* (pp. 316-318). Xi-an, China.

(Article history: Received 26 March 2013, Accepted 19 February 2014, Available online 3 April 2014)