International Journal of Approximate Reasoning 98 (2018) 11–24

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

A three-way clustering approach for handling missing data using GTRS

Mohammad Khan Afridi [a], Nouman Azam [a,*], JingTao Yao [b], Eisa Alanazi [c]

[a] National University of Computer and Emerging Sciences, Pakistan
[b] Department of Computer Science, University of Regina, Regina, SK, S4S 0A2, Canada
[c] Department of Computer Science, Umm Al-Qura University, Makkah, Saudi Arabia

ARTICLE INFO

Article history:
Received 3 November 2017
Received in revised form 19 February 2018
Accepted 1 April 2018
Available online 5 April 2018

Keywords:
Clustering
Three-way decisions
Game-theoretic rough sets
Missing data
Uncertainty

ABSTRACT

Clustering is an important data analysis task. It becomes a challenge in the presence of uncertainty due to reasons such as incomplete, missing or corrupted data. A three-way approach has recently been introduced to deal with uncertainty in clustering due to missing values. The essential idea is to make a deferment decision whenever it is not clear and possible to decide whether or not to include an object in a cluster. A key issue in the three-way approach is to determine the thresholds that are used to define the three types of decisions, namely, include an object in a cluster, exclude an object from a cluster, or delay (defer) the decision of inclusion or exclusion from a cluster. The existing studies do not sufficiently address the determination of thresholds and generally use its fix values. In this paper, we explore the use of game-theoretic rough set (GTRS) model to handle this issue. In particular, a game is defined where the determination of thresholds is approached based on a tradeoff between the properties of accuracy and generality of clusters. The determined thresholds are then used to induce three-way decisions for clustering uncertain objects. Experimental results on four datasets from UCI machine learning repository suggests that the GTRS significantly improves the generality while keeping similar levels of accuracy in comparison to other three-way and similar models.

1. Introduction

Clustering techniques aim to group similar objects into clusters. It has been widely used in the fields of computer science, engineering, medical sciences, social sciences and others [13–15,29,34]. Clustering techniques are broadly categorized as hierarchical clustering and partitional clustering techniques [5,11]. Hierarchical clustering group data objects by examining hierarchical relations among the objects where the levels in hierarchy are determined based on the degree of required granularity. Partitional clustering groups data objects into some predefined number of clusters [30]. In this study, we consider partitional clustering.

The clustering of data containing missing values one of the key issues in clustering [24,28,40]. Two strategies are commonly used in this regards [8,19,27]. The first strategy is to apply some preprocessing on the data. The second strategy is to incorporate additional mechanisms in clustering model for handling data with missing values. Commonly used preprocess-

* Corresponding author.
E-mail address: nouman.azam@nu.edu.pk (N. Azam).

https://doi.org/10.1016/j.ijar.2018.04.001
0888-613X/

ing techniques include deleting the tuple containing missing attribute values or imputing (replacing) the missing attribute values [19]. Extensions of these basic preprocessing techniques include pairwise deletion [7], mean substitution [6], regression imputation [3], expectation-maximization imputation [4], hot deck imputation [18], cold deck imputation [12], last observation carried forward [21] and multiple imputations [23]. It has been reported that imputation in preprocessing jeopardizes the quality and reliability of the classification results [27]. Another study reported that preprocessing based missing data treatments have imperfections, as they are rooted in specific statistical assumptions [20]. A more appropriate and suitable strategy is to equip the clustering model to handle data with missing values. The essential idea is to handle the data with missing values at the model level instead of modifying the data itself. Different attempts have been made under this strategy. For instance, assigning objects with missing values to a cluster having high number of missing values [27], creating a separate cluster for containing all objects with missing values [10], using the neighbors of objects with missing values to decide their assignment to clusters [8]. Other relevant studies are discussed in [16,22,25,39].

Yu recently introduced a three-way approach for clustering of data containing missing values [35]. The key idea was to make three-way decisions for each object corresponding to a particular cluster, i.e., accepting an object as belonging to a cluster, rejecting an object as belonging to a cluster or deferring the decision of acceptance or rejection. The deferment decision option is exercised whenever it is not possible and clear to accept or reject an object. The three-way decisions and the quality of the resulting three regions are critically controlled and defined based on a pair of thresholds. However, the automated determination of suitable values for these thresholds has not been sufficiently addressed by the studies of Yu and Yu et al. [35–37]. Their main focus remained on formulation of three-way approach for clustering and they considered the use of fixed thresholds in their studies. The use of fix or restricted thresholds does not allow to fine tune the regions in order to further improve the quality of clustering. In this study, we provide a three-way clustering approach based on game-theoretic rough sets (GTRS) for automatic determination of thresholds.

The GTRS model utilizes game-theoretic formulation to implement games between multiple criteria in the aim of reaching an effective tradeoff based solution. We have formulate a game in GTRS between two important properties of accuracy and generality of three-way clustering. The overall objective of this game is to determine suitable thresholds that are used to induce three-way clustering based on a tradeoff and balance between the two properties. In general, configuring the thresholds to increase accuracy affects generality and modifying the thresholds to improve generality affects accuracy. The aim of the game is to obtain thresholds based on a compromise between the two properties. Experimental results on different datasets from UCI machine learning repository suggest that on average an increase between 36% to 65% in generality can be achieved while keeping similar levels of accuracy for clustering data with missing values [17]. The reported results suggest that the proposed approach can effectively cluster data with missing values.

2. A critical review of three-way clustering for handling missing data

In this section, we motivate the present study by pointing out a limitation in existing three-way clustering approaches. Later in Section 3, we present a game-theoretic rough sets based three-way clustering approach for resolving this limitation. For the sake of completeness, we briefly review the key notions of the three-way clustering in the next section.

2.1. Basic notions of three-way clustering

The three-way clustering receives its motivation from theory of three-way decisions outlined by Yao [32,33]. Let there be a set $U = \{o_1, o_2, o_3, ...\}$, called universe of objects. A clustering scheme will result in a family of sets denoted as $\{c_1, c_2, c_3, ...\}$, where each c_k is a set of objects contained in that cluster. Each object o_i has A attributes, i.e., $o_i = (o_i^1, ..., o_i^A)$, where o_i^a denotes the value of the ath attribute corresponding to ith object.

In conventional clustering, a cluster is frequently represented by a single set. This representation essentially means that the objects belonging to the set definitely belong to a cluster and the objects not in a set definitely does not belong to a cluster [36]. From decision making perspective this is a two-way decision, i.e., an object either belongs to a cluster or it does not belong to a cluster. Two-way decisions are not always feasible especially in situations that are characterized by uncertainty and lack of information [32]. A more practical and reasonable approach is to consider three-way decisions where instead of two we have three decision choices. In particular, we may decide whether an object belongs to a cluster, or it does not belong to a cluster or we are unable to decide whether or not an object belongs to a cluster. This leads to the notion of three-way clustering.

A general framework of three-way clustering was introduced in [35]. In contrast to conventional approaches where a single set is used to represent a cluster which leads to two-way decisions, the three-way framework uses a couple of sets to represent a cluster and thereby lead to three-way decisions. In particular, each cluster c_k is represented by two sets, i.e.,

$$c_k = \{In(c_k),\ Pt(c_k)\}, \tag{1}$$

where, $In(c_k)$ and $Pt(c_k)$ are subsets of U. The two sets are used to create three regions corresponding to a cluster as follows.

$$Inside(c_k) = In(c_k), \tag{2}$$

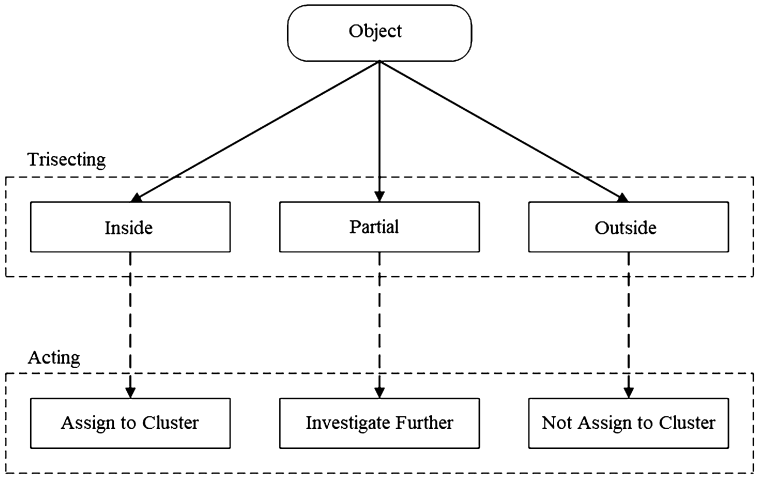

Fig. 1. Three-way clustering.

$$Partial(c_k) = Pt(c_k), \tag{3}$$

$$Outside(c_k) = U - In(c_k) - Pt(c_k). \tag{4}$$

The $Inside(c_k)$ contains objects that belong to the cluster, the $Partial(c_k)$ contains object which may belong to the cluster and the $Outside(c_k)$ region contains objects that does not belong to the cluster. It is important to note that there is a slightly different three-way framework used in [36]. In particular, they called the inside and partial regions as the lower and upper approximations, thereby relating it to the rough sets formulation. Following the three-way framework, the family of clusters is represented as,

$$\{\{In(c_1),\ Pt(c_1)\},\ \{In(c_2),\ Pt(c_2)\}, ...\}. \tag{5}$$

In order to obtain the three regions namely, inside, partial and outside, an evaluation function and a pair of thresholds may be used [35]. The evaluation function quantifies the relationship between an object and a cluster, and the thresholds define the bounds on the relationship for inclusion in different regions. Consider $e(c_k, o_i)$ to be an evaluation function representing the relationship or association between a certain cluster c_k and a particular object o_i, with (α, β) be some thresholds. The three regions are defined as follows.

$$Inside(c_k) = \{o_i \in U \mid e(c_k, o_i) \geq \alpha\}, \tag{6}$$

$$Partial(c_k) = \{o_i \in U \mid \beta < e(c_k, o_i) < \alpha\}, \tag{7}$$

$$Outside(c_k) = \{o_i \in U \mid e(c_k, o_i) \leq \beta\}. \tag{8}$$

This means that an object is included in the $Inside(c_k)$ when its evaluation is above or equal to threshold α. Similarly, an object is included in the $Outside(c_k)$ when its evaluation is below or equal to threshold β. The object is included in the $Partial(c_k)$ when its evaluation is between the two thresholds.

It is worth mentioning that the above three-way clustering framework may also be interpreted based on the general trisecting and acting framework of three-way decisions [33]. Fig. 1 shows a three-way interpretation of clustering based on the trisecting and acting framework. In the trisection step, we partition the objects into three disjoint regions corresponding to a particular cluster. In the acting step, we decide whether or not to include an object in a cluster.

The thresholds (α, β) control the inclusion in different regions and its different settings lead to different regions. How to determine the thresholds automatically is an important research issue in this context. This issue is however being ignored in the current literature. We, further highlight this issue in the next section.

2.2. Limitation in existing three-way clustering approach

In this section, we use a demonstrative example to highlight the issue of lack of consideration towards the determination of suitable thresholds for three-way clustering. We consider a three step approach for applying the three-way framework for handling data with missing values, reported in [33].

In the first step, the set of objects U is divided into set C and set M, where set C contains objects with no missing values and set M contain objects with missing values. The objects in set C are clustered using one of the conventional algorithms such as K-means. It is assumed that since these objects do not contain missing values, therefore the level of uncertainty will be low, therefore the conventional approaches will be more appropriate for clustering such objects. In the second step,

Table 1
Sample dataset with missing data, assumed missing values are marked with a *.

	A_1	A_2	A_3	A_4		A_1	A_2	A_3	A_4		A_1	A_2	A_3	A_4
o_1	5.9	3.2	4.8	2	o_{11}	5.6*	2.9	4.1*	1.5	o_{21}	6.3	2.7*	4.9	1.8*
o_2	6.1	2.8*	4.2	1.5*	o_{12}	5.5	2.5	4	1.5	o_{22}	6.2	2.8	4.8	1.8
o_3	6.4	2.8	4.6	1.3	o_{13}	5.5	2.6	4.4	1.4	o_{23}	5.9	3	5.1	1.8
o_4	6.4*	2.5	4.3*	1.4	o_{14}	6.1*	2.7	4.6*	1.4	o_{24}	6.4	2.8	5.6	2.1
o_5	6.3	2.3	4.4	1.5	o_{15}	5.8	2.6	4	1.4	o_{25}	6.5	3	5.5	1.8
o_6	6.3	2.8*	4.9	1.6*	o_{16}	5.8	2.7	5.1	1.9	o_{26}	6.3	2.8	5.1	1.5
o_7	5.5	2.4	3.8	1.3	o_{17}	5.7	2.5	5	2	o_{27}	6.1*	2.7	5.6*	1.5
o_8	5.8	2.7*	4	1.4*	o_{18}	6.1	2.8*	5.6	2.2*	o_{28}	6.4	3.1	5.5	1.8
o_9	5.5	2.4	3.7	1.2	o_{19}	6	2.2	5	1.5	o_{29}	6*	2.9	4.8*	1.6
o_{10}	6	2.8	4.5	1.4	o_{20}	5.6	2.8	4.9	2	o_{30}	5.9	3	5.1	1.8

an incomplete data set is constructed from C, where the rate of missing values is kept similar to the rate of missing values in dataset U. This means that if the original dataset contains 30% of objects with missing values, than approximately 30% of objects will be randomly selected from C for induced missing values. This leads to a division of C into two more sets, i.e., the constructed dataset containing objects with missing values denoted as U_m and the remaining objects in C with no missing values denoted as U_c. This step will help in selecting suitable values for (α, β) thresholds that will objects with clustering the objects having missing values. In the third step, the objects with missing values, denoted by M are being decided in the three-way framework outlined in Section 2.1. In particular, the association of an object with each cluster (determined in step one) is checked in a three-way framework. These three steps may be termed as the training, validation and testing steps. It is sufficient to consider the first two steps of this approach to highlight a limitation in existing studies.

Consider Table 1 that contains information about 30 objects. The rows of the table correspond to objects which are represented as $o_1, o_2, o_3, ..., o_{30}$ and the columns corresponds to 4 attributes which are represented as A_1, A_2, A_3 and A_4. We further assume that this data represent the set C and the missing rate in the original dataset, i.e., U was 30%. Therefore, we also randomly considered 30% of the objects having missing values from C. The missing values are assumed to be the values with a * on top of them. The induced missing values will be used to compute the (α, β) thresholds which may be later on applied on the objects in M to determine three-way clustering for those objects.

In the first step, we apply K-mean clustering on C with K = 2, which leads to the formation of two clusters, namely, $c_1 = \{o_1, ..., o_{15}\}$ and $c_2 = \{o_{16}, ..., o_{30}\}$. In the second step, based on the objects with induced missing values, we aim to determine suitable thresholds that will do a good job of clustering these objects. The three-way clustering approach introduced in Section 2.1, is used for this purpose.

To apply three-way clustering on the data with missing values, we need to compute the evaluation function $e(c_k, o_i)$ described in Equation (9). The evaluation function quantifies the relationship between an object o_i and cluster c_k and may be defined in different ways. We consider the evaluation function based on the relative number of nearest neighbors for object o_i belonging to cluster c_k. More formally, we define it as,

$$e(c_k, o_i) = \frac{\text{Number of } o_i \text{ neighbors belonging to } c_k}{\text{Total neighbors of } o_i}. \tag{9}$$

It may be noted that a slightly different evaluation function was considered in [35]. They considered the nearest neighbors that fall in some distance limit from the object o_i. To compute the neighbors, we need a certain distance metric. In this example, we consider the following distance metric,

$$d_{(i,\,j)} = \sqrt{\sum_{a=1}^{A} (o_i^a - o_j^a)^2}, \tag{10}$$

where o_i^a is the value of the ath attribute of the ith object. Moreover, we ignore the attributes with missing values while computing the distance. For instance, the distance between object o_2 and o_1 is determined as,

$$\begin{aligned} d_{(2,\,1)} &= \sqrt{\sum_{a=1}^{A} (o_2^a - o_1^a)^2} \\ &= \sqrt{(6.1-5.9)^2 + (*-3.2)^2 + (4.2-4.8)^2 + (*-2)^2} \\ &= \sqrt{(6.1-5.9)^2 + (4.2-4.8)^2} = 0.63 \end{aligned} \tag{11}$$

Using the above distance metric, we can compute the distances of each o_i with missing values, from all the objects in U_c. By sorting these distances, we can compute the nearest neighbors for each o_i. For instance, the distances of o_2 from all objects in U_c are $d(o_2, o_1) = 0.63$, $d(o_2, o_3) = 0.5$, ..., $d(o_2, o_{30}) = 0.92$. By sorting these distances, we find that the nearest

Table 2
Evaluation function $e(c_k, o_i)$ values for objects in U_m.

	o_2	o_4	o_6	o_8	o_{11}	o_{14}	o_{18}	o_{21}	o_{27}	o_{29}
c_1	0.86	1	0.43	1	0.57	0.86	0	0.29	0.71	0
c_2	0.14	0	0.57	0	0.43	0.14	1	0.71	0.29	1

Table 3
(Accuracy, Generality) values for different threshold values.

		α			
		1	0.85	0.7	0.5
β	0	(1.00, 0.40)	(1.00, 0.50)	(0.86, 0.60)	(0.70, 0.69)
	0.15	(1.00, 0.50)	(1.00, 0.60)	(0.86, 0.70)	(0.71, 0.80)
	0.3	(0.92, 0.60)	(0.93, 0.61)	(0.85, 0.80)	(0.83, 0.90)
	0.5	(0.86, 0.70)	(0.86, 0.80)	(0.84, 0.91)	(0.80, 1.00)

neighbors, say seven nearest neighbors of o_2 are o_5, o_{10}, o_{15}, o_3, o_{22}, o_1 and o_{12}. It should be noted that for the sake of simplicity, we used a basic distance metric used in Equation (10). Other distance measures can also be used [8,35].

Once the neighbors are determined, we can compute the evaluation function $e(c_k, o_i)$. For instance, considering cluster c_1, based on the seven neighbors of o_2, the evaluation function $e(c_1, o_2)$ based on the Equation (9), is given by,

$$e(c_1, o_2) = \frac{\text{Number of } o_2 \text{ neighbors belong to } c_1}{\text{Total neighbors of } o_2} = 6/7 = 0.86, \tag{12}$$

This means that 86% neighbors of o_2 belongs to cluster c_1. In the same way, the evaluation function $e(c_2, o_2)$ is given by,

$$e(c_2, o_2) = \frac{\text{Number of } o_2 \text{ neighbors belong to } c_2}{\text{Total number of } o_2 \text{ neighbors}} = 1/7 = 0.14. \tag{13}$$

The evaluation functions corresponding to the two clusters for all objects in U_m having missing values are given in Table 2. Once the evaluation functions are computed, we may use Equations (6)–(8) for inclusion of objects into one of the three regions. For instance, if we assume thresholds $(\alpha, \beta) = (1, 0)$, the object o_2 will be in the $Partial(c_1)$ and $Partial(c_2)$. This will mean that object o_2 is not being clustered. However, if we set thresholds $(\alpha, \beta) = (0.7, 0.25)$, than the object o_2 will belong to cluster c_1 and it will be in the outside region of the cluster c_2.

From the evaluation function results in Table 2, it may be noted that different threshold settings will lead to different regions. For instance, if we set thresholds $(\alpha, \beta) = (1, 0)$, than only objects o_4, o_8, o_{18} and o_{29} will be clustered. In particular, o_4 and o_8 will be in the $Inside(c_1)$ and o_{18} and o_{29} will be in $Inside(c_2)$. Since o_4 and o_8 belong to cluster c_1 and o_{18} and o_{29} belong to c_2 (see Equations (6)–(8)), this means that we have accurately clustered these objects. However, we were only able to cluster 4 out of 10 objects. This suggests that with the thresholds setting of $(\alpha, \beta) = (1, 0)$ we are able to cluster only 4 out of 10 or 40% objects with all of these 4 objects being correctly placed in their appropriate clusters thereby leading to 100% accuracy. On the other hand, if we set $(\alpha, \beta) = (0.5, 0.5)$, we will be able to cluster all the objects, however, 8 out of these 10 objects will be appropriately placed in their respective clusters thereby leading to 80% accuracy.

Let us consider the formal definitions for the accuracy and generality of clustered objects.

$$Accuracy(\alpha, \beta) = \frac{\text{Correctly clustered objects}}{\text{Total clustered objects}}, \tag{14}$$

$$Generality(\alpha, \beta) = \frac{\text{Total clustered objects}}{\text{Total objects in } U}. \tag{15}$$

Accuracy means how much accurately we cluster the objects with missing values and generality refers to percentage of objects that were actually being clustered. Based on the Table 1, we may compute the accuracy and generality for different thresholds setting. This is summarized in Table 3.

In general, modifying the thresholds to improve the generality or the number of clustered points may affect the accuracy and improving the accuracy may affect the generality. How to determine the thresholds in order to achieve a balance between accuracy and generality is a critical issue in this context. This issue has been overlooked in the current literature. In particular, they ignore the threshold determination. We provide a possible solution in this regard in Section 4. In the next section, we provide a visual representation of the same issue of accuracy versus generality and their impact on the thresholds.

2.3. A visual representation of the limitation

Fig. 2 considers three clusters corresponding to the objects in universe. The objects are represented by dots and the * represents the objects with missing values. The straight lines represent the boundaries of the clusters. There are circles

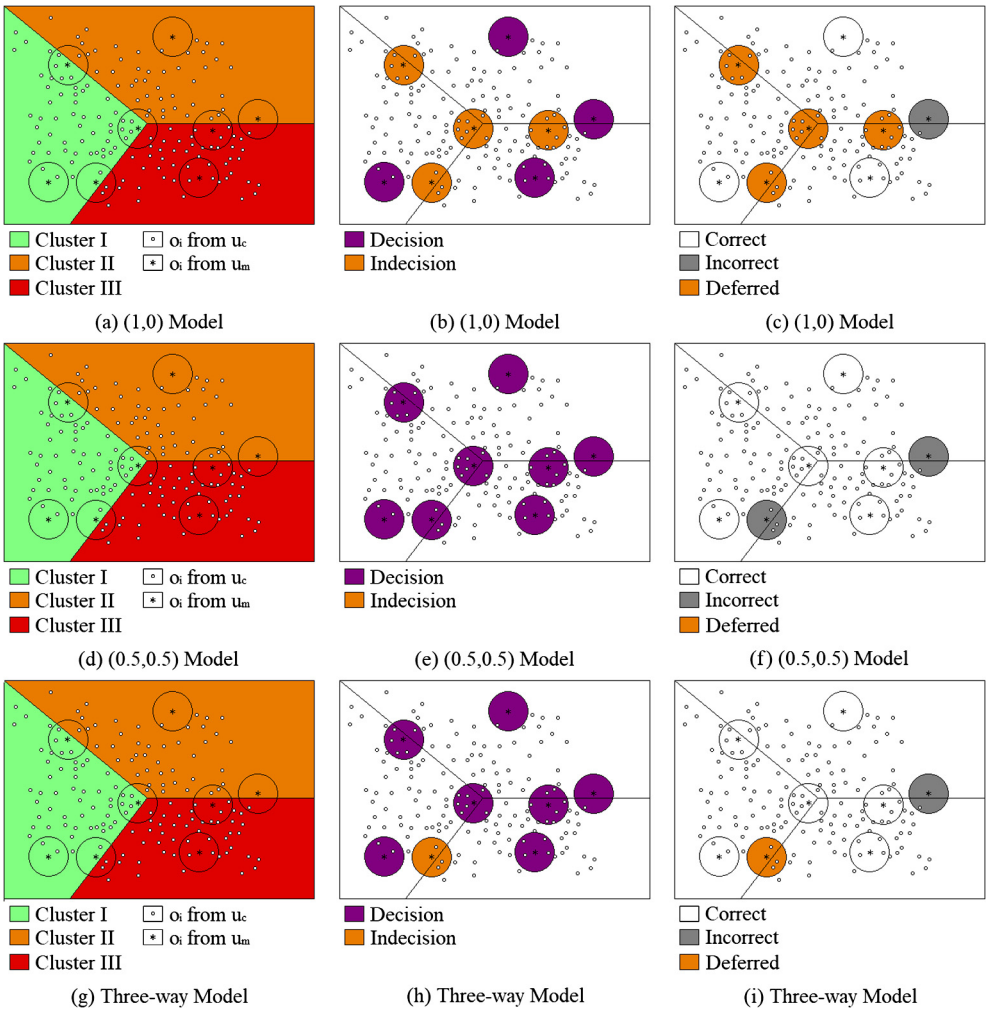

Fig. 2. Accuracy versus generality. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

around the objects with missing values. The objects inside these circles represent the neighbors of the objects with missing values that are in some specified distance range from it. Moreover, we consider an evaluation function which is defined in terms of the relative number of neighbors in the specified distance. In particular, $e(c_k, o_i)$ equals 1 when all the neighbors belong to the cluster c_k and $e(c_k, o_i)$ equals 0 when none of the neighbors belong to c_k.

Fig. 2(a)–(c) correspond to the thresholds setting of $(\alpha, \beta) = (1, 0)$. This means that we will be able to cluster an object with missing values only if all of its neighbors belong to the same cluster, i.e., $e(c_k, o_i) = 1$. Fig. 2(b) shows that with this thresholds setting, we are only able to cluster 4 out of 8 objects. Moreover, Fig. 2(c) shows that 3 out of 4 objects are clustered correctly.

Fig. 2(d)–(f) correspond to another extreme setting of $(\alpha, \beta) = (0.5, 0.5)$. According to this setting we are able to cluster an object with missing value only if 50% or more of its neighbors belong to the same cluster. In other words, the evaluation function $e(c_k, o_i) \geq 0.5$. Fig. 2(e) reveals that all the objects are clustered with this threshold configuration. Moreover, Fig. 2(f) shows that 6 out of 8 objects are clustered correctly.

Fig. 2(g)–(i) corresponds to a three-way clustering that is based on determination of suitable (α, β) thresholds, say $(\alpha, \beta) = (0.7, 0.3)$. This means that we will be able to cluster an object with missing value when 70% or more of its neighbors belong to the same cluster. Fig. 2(h) shows that with this thresholds setting, we are only able to cluster 7 out of 8 objects. Moreover, Fig. 2(i) shows that 6 out of 7 objects will be clustered correctly.

From Fig. 2, it may be noted that the two properties of accuracy and generality are critically dependent on the choice of the thresholds. In one extreme thresholds configuration of $(\alpha, \beta) = (1, 0)$ we have maximum accuracy but not very effective generality. Similarly, in another extreme thresholds settings of $(\alpha, \beta) = (0.5, 0.5)$, we have maximum generality but not necessarily an effective accuracy. The three-way approach provides a threshold setting between these two extreme cases based on a compromise and tradeoff between accuracy and generality. We examine the use of GTRS for determining such a tradeoff.

Table 4
A typical two-player game in GTRS.

		P_2		
		s_1	s_2	...
P_1	s_1	$u_1(s_1, s_1)$, $u_2(s_1, s_1)$	$u_1(s_1, s_2)$, $u_2(s_1, s_2)$	...
	s_2	$u_1(s_2, s_1)$, $u_2(s_2, s_1)$	$u_1(s_2, s_2)$, $u_2(s_2, s_2)$	...
	...	...	...	...

3. Game theoretic rough sets

The game-theoretic rough sets provides a game-theoretic environment for reading a tradeoff solution between multiple criteria that are realized as game players [9,31]. It formulates strategies for players in the form of changes in thresholds in order to improve the overall quality of three-way decisions. More specifically, each player participates in the game by configuring the thresholds with the aim to maximize its benefits and utilities. The overall objective of a game in GTRS is to select suitable thresholds for three-way decisions, based on the available criteria.

A typical game in GTRS is defined as a tuple {P, S, u}, where [26],

- P is a finite set of n players,
- $S = S_1 \times \ldots \times S_n$, where S_i is a finite set of strategies available to each player i. Each vector $s = (s_1, ..., s_n) \in S$ is called a strategy profile where player i plays strategy s_i,
- $u = (u_1, ..., u_n)$ where $u_i : S \mapsto \Re$ is a real-valued utility or payoff function for player i.

Let us denote the strategy profile of all the players in the game except player i as $s_{-i} = (s_1, ..., s_{i-1}, s_{i+1}, ..., s_n)$. This means that we can write $s = (s_i, s_{-i})$. Thus, all the players except i are committed to play s_{-i} and player i choosing s_i. Nash equilibrium is generally used to determine game solution or game outcome in GTRS. A strategy profile $(s_1, ..., s_n)$ is a Nash equilibrium, when,

$$u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i}), \text{ where } (s_i' \neq s_i) \tag{16}$$

In a typical GTRS game, the players are considered as different criteria that highlight various quality related aspects of three-way decisions such as accuracy, generality, precision recall, uncertainty or cost [2]. The strategies are formulated as different level of changes in the thresholds defining three-way decisions. Each criterion is affected differently when various strategies are considered. The strategies are therefore formulated as changes in the thresholds. Finally, suitable measures are defined for evaluating each criterion. The values of these measures reflect the payoffs of different players or criteria.

Table 4 shows a typical two player game in GTRS. The players in the game are denoted by P_1 and P_2. Cells in the table correspond to strategy profiles. Each cell contains a pair of payoff functions based on their strategy profile. For example the top right cell corresponds to a strategy profile (s_1, s_2) which contains payoff functions $u_1(s_1, s_2)$ and $u_2(s_1, s_2)$.

Playing the game results in the selection of Nash equilibrium which is utilized in determining a possible strategy profile and the associated thresholds. In the next section, we propose a GTRS based approach for determining thresholds of three-way clustering.

4. Three-way clustering with GTRS

Earlier in Section 2.2, we demonstrated a relation between the threshold pair (α, β) and the properties of accuracy and generality. A key observation was that the configuration of the threshold pair (α, β) controls the tradeoff between accuracy and generality. In this section, we propose an approach based on the GTRS, which considers the tradeoff between accuracy and generality and automatically determines the thresholds.

4.1. Formulating a game in GTRS for three-way clustering

From GTRS description in Section 3, we noted that in order to analyze problems with GTRS, we need to formulate them as games. Three components needs to be identified for this purpose, i.e., the players, the strategies and the payoff or utility functions.

The players should reflect the overall intention and purpose of the game. The objective in this game is to improve the quality of clustering data with missing values. In Section 2.2, we noted that this objective may be approached from a viewpoint of tradeoff between accuracy and generality of the clustering. The players in this game are therefore considered as the properties of accuracy and generality of the clustering. The player accuracy will be denoted by A and player generality will be denoted by G. The player set is given as $P = \{A, G\}$.

The strategies represent different actions of a player in a game. Each player chooses a strategy in order to maximize his benefits. In Section 2.2, we noted that the properties of accuracy and generality are affected differently when various thresholds are being selected. We therefore consider modification in thresholds as possible strategies. In particular, we

Table 5
Payoff table for the game.

		G		
		$s_1 = \alpha\downarrow$	$s_2 = \beta\uparrow$	$s_3 = \alpha\downarrow\beta\uparrow$
A	$s_1 = \alpha\downarrow$	$u_A(s_1, s_1)$, $u_G(s_1, s_1)$	$u_A(s_1, s_2)$, $u_G(s_1, s_2)$	$u_A(s_1, s_3)$, $u_G(s_1, s_3)$
	$s_2 = \beta\uparrow$	$u_A(s_2, s_1)$, $u_G(s_2, s_1)$	$u_A(s_2, s_2)$, $u_G(s_2, s_2)$	$u_A(s_2, s_3)$, $u_G(s_2, s_3)$
	$s_3 = \alpha\downarrow\beta\uparrow$	$u_A(s_3, s_1)$, $u_G(s_3, s_1)$	$u_A(s_3, s_2)$, $u_G(s_3, s_2)$	$u_A(s_3, s_3)$, $u_G(s_3, s_3)$

consider three strategies, namely, decrease in threshold α (denoted as $\alpha\downarrow$), increase in threshold β (denoted as $\beta\uparrow$) and decrease α and increase β simultaneously (denoted as $\alpha\downarrow\beta\uparrow$).

A payoff function is used to measure the results of selecting a certain strategy. Please note that these strategies are being formulated by considering initial thresholds $(\alpha, \beta) = (1, 0)$. Other ways of formulating are described in [1]. These functions are defined to reflect the possible performance gains or benefits of a particular player in choosing a certain strategy. As discussed earlier, the two players A and G are affected based on different threshold values (which in this case are game strategies). For a particular strategy profile, say (s_m, s_n) that leads to thresholds (α, β), the associated payoffs of the players are defined as,

$$u_A(s_m, s_n) = Accuracy(\alpha, \beta), \tag{17}$$

$$u_G(s_m, s_n) = Generality(\alpha, \beta), \tag{18}$$

where u_A and u_G are the payoff functions of players A and G, respectively and $Accuracy(\alpha, \beta)$ and $Generality(\alpha, \beta)$ are defined in Equations (14) and (15). For both the players, a value of 1 means maximum payoff and a value of 0 means minimum payoff.

4.2. Realization of accuracy versus generality tradeoff as a game

We consider the constructed game as a competition between accuracy and generality of clustering. Table 5 is used to highlight this. The rows correspond to strategies for player A and the columns correspond to strategies of player G. Each cell represents a strategy profile of the form (s_m, s_n) where s_m is the strategy of player A and s_n is the strategy of player G. Each player aims to select a strategy that will configure the thresholds in order to improve his respective utility. The payoffs corresponding to the strategy profile (s_m, s_n) are given by $u_A(s_m, s_n)$ and $u_G(s_m, s_n)$ for players A and G, respectively.

A logical thing to do for a player in a game is to prefer a strategy having higher payoff over strategies with lower payoffs. According to the definition of Nash equilibrium in Equation (16), for the considered two player game, a strategy profile will be the Nash equilibrium if,

$$\text{For Accuracy: } \forall s_m \in S_A,\ u_A(s_m, s_n) \geq u_A(s'_m, s_n),\ \text{where } (s'_m \neq s_m), \tag{19}$$

$$\text{For Generality: } \forall s_n \in S_G,\ u_G(s_m, s_n) \geq u_G(s_m, s'_n),\ \text{where } (s'_n \neq s_n). \tag{20}$$

This means that no player will benefit from changing their strategy other than the strategy specified by the profile (s_m, s_n).

We now examine how to determine the changes in the thresholds based on a certain strategy profile. From the game description in Section 4.1, we noted that there are four ways for changing the thresholds, namely,

$$\alpha- = \text{a single player suggests to decrease } \alpha, \tag{21}$$

$$\alpha-- = \text{both players suggest to decrease } \alpha, \tag{22}$$

$$\beta+ = \text{a single player suggests to increase } \beta, \tag{23}$$

$$\beta++ = \text{both players suggest to increase } \beta. \tag{24}$$

The above definitions can be used to associate threshold pairs with a certain strategy profile. For instance, a strategy profile with (s_1, s_1) which equals to $(\alpha\downarrow, \alpha\downarrow)$ is represented as $(\alpha--, \beta)$, since both the players suggest to decrease threshold α (see Equation (22)). In the next section, we examine how to obtain the values of the four variables in Equations (21)–(24) based on an interactive game.

4.3. Iterative threshold learning in GTRS

A single run of the game has limited application for searching suitable thresholds. Modifying the thresholds iteratively with the aim to improve the payoffs for the players will lead to a learning mechanism. The learning rule or criterion in this case is based on the relationship between modification in thresholds and its impact on the utilities of the players. We utilize this relationship in order to define the variables $(\alpha-, \alpha--, \beta+, \beta++)$. An iterative game is defined for this purpose.

Algorithm 1 GTRS based threshold learning algorithm.

Input: K as number of clusters, U as a dataset and initial values of $\alpha-$, $\alpha--$, $\beta+$ and $\beta++$.
Output: Three-way clustering of objects.
1: Initialize $\alpha = 1.0$, $\beta = 0.0$.
2: Divide U into C and M. # C is the set of objects with no missing values and M is the set of objects with missing values.
3: Apply K-mean clustering on C.
4: Randomly remove values from C by following the percentage of missing values in M.
5: Divide C into U_c and U_m. # U_c is the set of objects with no missing values and U_m is the set of objects with simulated missing values.
6: **Repeat**
7: Calculate the utilities of players by using Equations (14) and (15).
8: Populate the payoff table with calculated values.
9: Calculate equilibrium in a payoff table by using Equations (19) and (20).
10: Determine selected strategies and corresponding thresholds (α', β').
11: $(\alpha, \beta) = (\alpha', \beta')$.
12: **Until** $Accuracy(\alpha, \beta) \leq Generality(\alpha, \beta)$ or $\alpha \leq 0.5$ or $\beta \geq 0.5$ or Maximum iterations reached.
13: Evaluate objects in M using Equation (9).
14: Use (α, β) determined in Line 11, with three-way framework of Equations (6)–(8) for assigning objects to different regions of a clusters.

Table 6
Example payoff table of the game.

		Generality		
		$s_1 = \alpha\downarrow$	$s_2 = \beta\uparrow$	$s_3 = \alpha\downarrow\beta\uparrow$
Accuracy	$s_1 = \alpha\downarrow$	(0.86, 0.60)	(1.0, 0.60)	(0.86, 0.70)
	$s_2 = \beta\uparrow$	(1.0, 0.60)	(0.92, 0.60)	**(0.93, 0.70)**
	$s_3 = \alpha\downarrow\beta\uparrow$	(0.86, 0.70)	(0.93, 0.70)	(0.86, 0.80)

Let (α, β) be the starting thresholds in a certain iteration of a repeated game. The Nash equilibrium will be used to compute and determine the game solution and the corresponding thresholds, say, (α', β'). By considering the initial thresholds to be (α, β) and the computed thresholds based on the game solution to be (α', β'), the four variable in Equations (21)–(24) are defined as,

$$\alpha- = \alpha - (\alpha \times (Generality(\alpha', \beta') - Generality(\alpha, \beta)), \tag{25}$$

$$\alpha-- = \alpha - c(\alpha \times (Generality(\alpha', \beta') - Generality(\alpha, \beta)), \tag{26}$$

$$\beta+ = \beta - (\beta \times (Generality(\alpha', \beta') - Generality(\alpha, \beta)), \tag{27}$$

$$\beta++ = \beta - c(\beta \times (Generality(\alpha', \beta') - Generality(\alpha, \beta)). \tag{28}$$

This means that we consider the changes in the threshold proportional to the improvement in generality or applicability. The constant c in the Equations (26) and (28) is used to control the level of change in thresholds. When c is set to a small value, we can fine tune the thresholds at a cost of more computational overhead. When c is set to a high value, we have lesser computations however, fine tuning of thresholds based on the data may not be possible. The iterative process stops when either the $Generality(\alpha, \beta)$ exceeds $Accuracy(\alpha, \beta)$ or boundary region becomes empty or runs a specified number of times or when either $\alpha \leq 0.5$ or $\beta \geq 0.5$. Algorithm 1 describes the iterative GTRS based learning mechanism for the determination of thresholds. A dataset is provided to the algorithm which then computes a threshold pair (α, β) for the three-way clustering.

4.4. Demonstrating the application of GTRS

We determine the role of GTRS for determining the thresholds of three-way clustering by considering the example introduced earlier in Section 2.2. Considering a game between accuracy and generality discussed in Section 4, with three strategies for each player. To simplify the example, consider a non-repetitive one time game. Moreover, consider simplified strategies in the form of increases or decreases of 15% or 0.15 in the thresholds. The game is being played based on initial thresholds setting of $(\alpha, \beta) = (1, 0)$. A particular strategy say s_1 requiring a decrease in α, will be interpreted as a 15% decrease in α leading to $\alpha = 0.85$. When both the players suggest an increase or decrease in a threshold value, then the new value will be determined as the sum of both the changes.

Table 6 shows the payoff table corresponding to this game based on the data in Table 1. The Nash game solution based on Equations (19) and (20), is highlighted by bold values i.e., **(0.93, 0.70)** which corresponds to a strategy profile (s_2, s_3). The game solution in this case means that, from the total objects with missing values, we are able to cluster 70% of the objects at an accuracy of 93%. Earlier in Section 2.2, we noted that with the extreme case of $(\alpha, \beta) = (1, 0)$ we were only able to cluster 40% of the objects with missing values at an accuracy of 100%. The GTRS in comparison is able to cluster 30% more objects at a cost of only 7% decrease in accuracy.

Table 7
Results from an iterative GTRS game.

Iteration	α	β	Accuracy	Generality
Initial	1.0000	0.0000	0.9756	0.3624
1	0.9200	0.2197	0.9741	0.8596
2	0.8355	0.2197	0.9673	0.9073
3	0.7983	0.3046	0.9680	0.9225
4	0.7713	0.3046	0.9593	0.9324
5	0.7120	0.3246	0.9563	0.9606

Table 8
Wisconsin Breast Cancer dataset.

Missing	(1, 0) Model		GTRS Model			(0.5, 0.5) Model	
	Accuracy	Generality	(α, β)	Accuracy	Generality	Accuracy	Generality
5%	0.9914	0.2565	(0.71, 0.29)	0.9797	0.9818	0.9251	0.9812
10%	0.9961	0.2621	(0.71, 0.24)	0.9737	0.9753	0.9287	0.9888
15%	0.9868	0.2984	(0.71, 0.31)	0.9748	0.9802	0.9326	0.9986
20%	0.9917	0.3029	(0.76, 0.29)	0.9769	0.9751	0.9398	0.9984
25%	0.9869	0.3078	(0.72, 0.36)	0.9723	0.9759	0.9379	0.9981
30%	0.9878	0.3216	(0.76, 0.26)	0.9740	0.9705	0.9391	0.9983
Average:	0.9901	0.2915	(0.73, 0.29)	0.9752	0.9765	0.9322	0.9939

5. Experimental results and discussion

In this section, we present detailed experimental results of the proposed GTRS based three-way clustering approach. In particular, the performance of the proposed approach is evaluated on four datasets from the UCI machine learning repository and are compared with and six other commonly used approaches for handling missing data [17]. Below is a brief description of these datasets.

5.1. Datasets description

Wisconsin Breast Cancer: This dataset is obtained from the University of Wisconsin hospital and contains 10 attributes. The total number of instances are 683 which are divided into two classes namely malignant and benign, with 444 and 239 instances respectively.

Iris: This dataset contains information on Iris flowers of three related species, namely Setosa, Virginica and Versicolor. There are 50 instances for each of these species. Moreover, there are 4 attributes.

Pen-Based Recognition of Handwritten Digits: This dataset is a collection of handwriting samples from 44 writers. A sample of 250 digits belongs to each writer. There are 16 attributes with total instances of 10,992.

Seed: This data set contains information about the kernels belonging to 3 different varieties of wheat, namely Kama, Rosa and Canadian. There are 70 instances belonging to each variant and a total of 7 attributes.

5.2. Experimental results of GTRS-based approach

In order to apply the GTRS based approach, we need to execute Algorithm 1 outlined in Section 4.3. The input to the algorithm is a dataset and the input parameter K, which controls the number of clusters. Each experiment was performed multiple times and the average values is reported in the tables. To simulate a dataset with missing values, we randomly removed attribute values for some of the objects. A fixed percentage of objects were randomly selected from the list of objects to have missing values and were added to the list U_m. In the experiments, we use different percentages of missing values, i.e., 5%, 10%, 15%, 20%, 25% and 30%. In all the experiments, we use the initial values of $\alpha-$ equals to 0.95 and $\alpha--$ equals to 0.90 and $\beta+$ equals to 0.05 and $\beta++$ equals to 0.1. Please be noted that the initial values of these variables do not significantly change the final output of the algorithm and one can use their different values while obtaining similar results. The number of iterations is not fixed, however maximum number of iterations are set to stop the algorithm from running in an infinite loop in case the algorithm does not converges. The maximum iteration in our algorithm was set to 20, but the algorithm usually converged between 5 to 9 iterations, depending on the dataset and the value of constant c (constant c was usually set to 1.2).

Table 7 shows how the thresholds are changed in different iterations of the game and its impact on the accuracy and generality. This table is being constructed for the game implemented on the Wisconsin Breast Cancer dataset. Before the game starts, the initial thresholds are being set to $(\alpha, \beta) = (1, 0)$ which leads to an accuracy of 0.9756 and generality of 0.3624. In subsequent iterations, we note an aggressive increase in generality with some decrease in accuracy. In iteration 1, both the thresholds are being updated. In iteration 2, threshold α is decreased, while threshold β stayed the same. The accuracy had a slight decrease while generality increase from 0.8596 to 0.9073. In iteration 3, threshold α decrease while

Table 9
Iris dataset.

Missing	(1, 0) Model		GTRS model			(0.5, 0.5) Model	
	Accuracy	Generality	(α, β)	Accuracy	Generality	Accuracy	Generality
5%	1	0.3743	(0.55, 0.29)	0.9402	0.9394	0.8991	0.9914
10%	0.9908	0.3453	(0.56, 0.21)	0.9473	0.9360	0.8951	0.9920
15%	0.9831	0.3827	(0.55, 0.25)	0.9407	0.9382	0.8844	0.9941
20%	0.9844	0.3927	(0.57, 0.20)	0.9375	0.9187	0.8865	0.9923
25%	0.9733	0.3903	(0.57, 0.30)	0.9354	0.9151	0.8895	0.9932
30%	0.9746	0.4062	(0.58, 0.25)	0.9280	0.9156	0.8877	0.9933
Average:	0.9844	0.3819	(0.56, 0.24)	0.9382	0.9272	0.8904	0.9927

Table 10
Pen-based recognition of handwritten digits dataset.

Missing	(1, 0) Model		GTRS model			(0.5, 0.5) Model	
	Accuracy	Generality	(α, β)	Accuracy	Generality	Accuracy	Generality
5%	0.9888	0.1217	(0.59, 0.23)	0.9574	0.9545	0.8917	0.9943
10%	0.9946	0.1947	(0.58, 0.20)	0.9513	0.9494	0.9001	0.9914
15%	0.9944	0.1892	(0.61, 0.23)	0.9478	0.9455	0.9040	0.9981
20%	0.9955	0.1947	(0.62, 0.23)	0.9455	0.9440	0.9013	0.9843
25%	0.9946	0.1975	(0.62, 0.21)	0.9443	0.9399	0.9060	0.9943
30%	0.9931	0.2013	(0.62, 0.23)	0.9411	0.9414	0.9020	0.9886
Average:	0.9935	0.1832	(0.61, 0.22)	0.9479	0.9458	0.9008	0.9918

Table 11
Seeds dataset.

Missing	(1, 0) Model		GTRS model			(0.5, 0.5) Model	
	Accuracy	Generality	(α, β)	Accuracy	Generality	Accuracy	Generality
5%	1	0.3590	(0.57, 0.26)	0.9575	0.9550	0.9180	1
10%	0.9973	0.3448	(0.57, 0.28)	0.9552	0.9519	0.9071	0.9986
15%	0.9898	0.3045	(0.57, 0.21)	0.9501	0.9539	0.9093	0.9968
20%	0.9897	0.3552	(0.58, 0.32)	0.9525	0.9476	0.9092	0.9962
25%	0.9868	0.3229	(0.58, 0.23)	0.9469	0.9438	0.9061	0.9944
30%	0.9879	0.3584	(0.58, 0.22)	0.9518	0.9475	0.9080	0.9933
Average:	0.9919	0.3408	(0.57, 0.25)	0.9523	0.9500	0.9096	0.9965

threshold β increased. This increased generality while accuracy remained the same. In iteration 4, threshold α decrease while threshold β stayed the same, resulting in a slight decrease in accuracy and a slight increase in generality. In iteration 5 the algorithm converged due to condition generality $\geq$ accuracy being met. The result at convergence or accuracy is 0.9563 and generality is 0.9606.

Table 8 presents the experimental results on the Wisconsin Breast Cancer dataset. For the sake of comparisons, we also include the results for extreme threshold settings, i.e., $(\alpha, \beta) = (1, 0)$ and $(\alpha, \beta) = (0.5, 0.5)$. We will refer to these thresholds settings as (1, 0) model and (0.5, 0.5) model. The accuracy for the (1, 0) model averages around 99%. However, clustering was possible for only 29% of the objects with missing values. Arguably, one would like the clustering of more objects. This is achieved with the (0.5, 0.5) model. In particular, this model was able to cluster around 99.4% of the objects with an accuracy of 93%. Where as, the GTRS model is able to achieve 97.5% accuracy for 97.6% of the objects. We may note that in comparison to $(\alpha, \beta) = (1, 0)$ model, the GTRS improved generality by 68.5% at the cost of 1.5% decrease in accuracy. Similarly, in comparison to (0.5, 0.5) model, the GTRS improved accuracy by 4.3% at the cost of only 1.7% decrease in generality. Overall, the GTRS provides acceptable accuracy with significant improvement in generality. This observation will become more obvious in the experimental results on the other datasets.

Experimental results on Iris dataset are presented in Table 9. The (1, 0) model is able to achieve an accuracy of 98.4% for 38% of the objects. In comparison to Wisconsin Breast Cancer dataset, the generality of (1, 0) model in this case is comparatively better. The accuracy for (0.5, 0.5) model is 89%, with a very high generality of 99.3%. This means that in comparison to (1, 0) model, it provides 60% improvement in generality. The GTRS model set $(\alpha, \beta) = (0.56, 0.24)$. The accuracy for GTRS is 92.8%, with generality of 92.7%. The GTRS model improved generality by 54.5% at the cost of 5.6% accuracy when compared with (1, 0) model. In comparison to the (0.5, 0.5) model, the GTRS improves accuracy by 4.8% at the cost of merely 6.5% generality.

Results on the experiments on Pen-Based Recognition of Handwritten Digits and Seeds dataset are presented in Table 10 and Table 11, respectively. The accuracy with (1, 0) model is 99% for both the datasets and the generalities are 18% and 34% respectively for the two datasets.

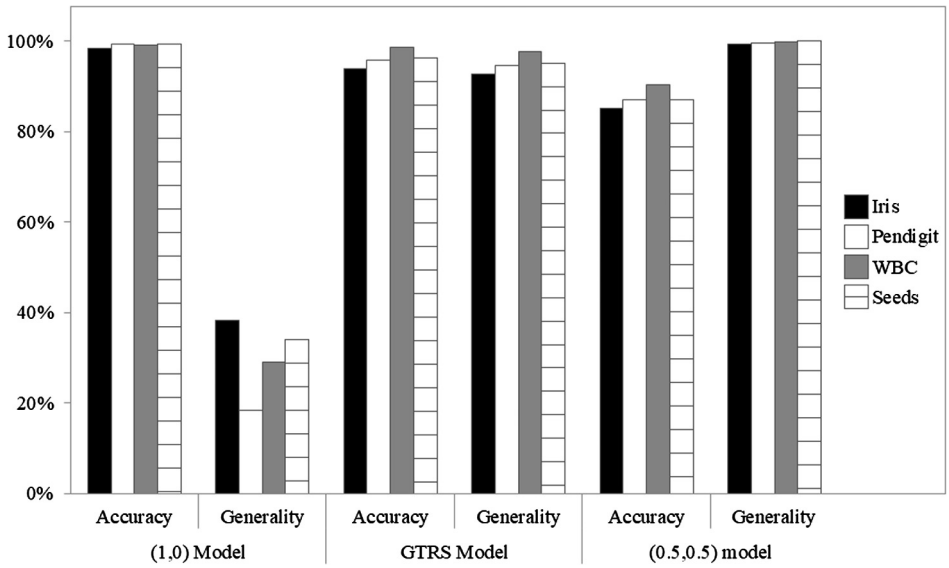

Fig. 3. Accuracy and generality results.

Table 12
Comparison results of accuracy.

GTRS model	Iris			Pendigit		
	10%	15%	20%	10%	15%	20%
	0.947	0.941	0.938	0.951	0.948	0.945
Three-way Clustering for Incomplete Data [35]	0.914	0.913	0.893	0.753	0.764	0.737
Whole Data Strategy [8]	0.583	0.468	0.464	0.331	0.323	0.319
Partial Distance Strategy [8]	0.898	0.892	0.889	0.663	0.689	0.676
Optimal Completion Strategy [8]	0.883	0.858	0.867	0.539	0.464	0.369
Nearest Prototype Strategy [8]	0.869	0.845	0.807	0.630	0.581	0.530
Nearest Neighbor Interval [38]	0.900	0.889	0.811	0.489	0.481	0.421

On the other hand, the accuracy for (0.5, 0.5) model is 90% and 91% with the generalities 99% and 99.6%, for the two datasets, respectively. The determined thresholds with GTRS for the two datasets were $(\alpha, \beta) = (0.61, 0.22)$ and $(\alpha, \beta) = (0.57, 0.25)$, respectively. The accuracies for the two datasets with GTRS are 94.8% and 95% and the generalities are 94.6% and 95%, respectively. The GTRS model improved generality by 76.3% and 61% at a cost of 4.6% accuracy when compared to (1, 0) model. Similarly, in comparison to (0.5, 0.5) model, the GTRS model improved accuracy by 4.8% and 4.3% at the cost of only 4.6% decrease in generality for both the datasets. Fig. 3 visually summarizes the experimental results for the considered four datasets.

5.3. Comparisons with other approaches

To gain more insights, we compare the GTRS results with existing three-way approaches for clustering that is based on fixed thresholds [35,36]. Moreover, we also include some of the other existing methods whose results were also being reported in [35]. Since out of the four datasets we considered, the study in [35] report the results for the Iris and the Pendigit datasets, therefore we only report comparisons on these two datasets. Table 12 shows the comparisons of the GTRS based approach with the other methods. For the Iris dataset, the GTRS model achieves 94.2% accuracy, outperforming the other existing methods including the fixed thresholds. The difference between the second best and the GTRS accuracy turns out to be 4.1%. Similarly, for the Pendigit dataset, the GTRS model achieves 94.8% accuracy, outperforming all other methods. Improvement in accuracy with GTRS model compared to the second best model is 17.1%. These results further indicate that the GTRS model may be a more suitable approach for handling data with missing values in clustering. The results reported in this section suggest that the GTRS is able to successfully configure the thresholds in order to improve the overall quality of clustering the objects with missing values. In general, the improvement with GTRS in generality is between 34% to 65%, while reduction in accuracy ranges from 1.7% to 6.5%. It is further observed that in some cases we can maintain accuracy

within 1% of the maximum accuracy and still achieves upto 20% increase in generality. These results suggest that the GTRS may be considered as a useful alternative approach for handling data with missing values in clustering.

6. Conclusion

This paper considers clustering in the presence of uncertainty due to missing values. The three-way approach provides an effective solution for handling missing values in clustering. A key issue in the application of three-way decisions is the determination of suitable thresholds that defines the three types of decisions. We approach this issue as a GTRS based competitive game is proposed between accuracy and generality that automatically determines the thresholds based on the data itself. Experiments are being performed on four different data sets from the UCI machine learning repository. The comparison of the GTRS results with another three-way model of (1, 0) suggests that the GTRS significantly improves the generality between 34% to 65% while maintaining similar levels of accuracy. In comparison to the (0.5, 0.5) model, the GTRS improves accuracy by upto 5% at a cost of some decrease in generality. In contrast to exiting three-way approach for clustering missing data which is based on arbitrary selected thresholds, the GTRS improves the accuracy on Iris dataset and Pendigit dataset by 3% and 20% respectively. Moreover, the GTRS also outperforms some of the existing approaches for handling missing data on the these two datasets. The GTRS model can be considered as a useful alternative for clustering objects with missing values.

The proposed approach may be further enhanced by considering different evaluation functions for quantifying the relationship between an object and a cluster. Moreover different stopping conditions in the iterative GTRS based game may also produce fruitful results.

Acknowledgements

This work was partially supported by Higher Education Commission of Pakistan and NSERC discovery grant Canada.

References

[1] N. Azam, J.T. Yao, Formulating game strategies in game-theoretic rough sets, in: Proceedings of 8th International Conference on Rough Sets and Knowledge Technology, RSKT 2013, in: Lecture Notes in Computer Science, vol. 8171, 2013, pp. 145–153.
[2] N. Azam, J.T. Yao, Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets, Int. J. Approx. Reason. 55 (1) (2014) 142–155.
[3] C.H. Brown, Asymptotic comparison of missing data procedures for estimating factor loadings, Psychometrika 48 (2) (1983) 269–291.
[4] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. B 39 (1) (1977) 1–38.
[5] B.S. Everitt, S. Landau, M. Leese, D. Stahl, Cluster Analysis, Wiley Series in Probability and Statistics, Wiley, 2011.
[6] A.V. Eye, Statistical Methods in Longitudinal Research: Principles and Structuring Change, Statistical Modeling and Decision Science, vol. 1, 2014.
[7] V. Haitovsky, Missing data in regression analysis, J. R. Stat. Soc. 30 (1968) 67–82.
[8] R.J. Hathaway, J.C. Bezdek, Fuzzy c-means clustering of incomplete data, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 31 (5) (2001) 735–744.
[9] J.P. Herbert, J.T. Yao, Game-theoretic rough sets, Fundam. Inform. 108 (3–4) (2011) 267–286.
[10] N. Iam-On, T. Boongeon, S. Garrett, C. Price, A link-based cluster ensemble approach for categorical data clustering, IEEE Trans. Knowl. Data Eng. 24 (3) (2012) 413–425.
[11] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[12] W.D. Kalsbeek, A conceptual review of survey error due to nonresponse, in: Proceedings of the Section on Survey Research Methods, American Statistical Association, 1980, pp. 131–136.
[13] H.F. Köhn, D. Steinley, M.J. Brusco, The p-median model as a tool for clustering psychological data, Psychol. Methods 15 (1) (2010) 87–95.
[14] G. Kou, Y. Peng, G.X. Wang, Evaluation of clustering algorithms for financial risk analysis using MCDM methods, Inf. Sci. 275 (2014) 1–12.
[15] R.M. Leech, S.A. McNaughton, A. Timperio, The clustering of diet, physical activity and sedentary behavior in children and adolescents: a review, Int. J. Behav. Nutr. Phys. Act. (Online) 11 (1) (2014) 4.
[16] D. Li, H. Gu, L.Y. Zhang, A hybrid genetic algorithm – fuzzy c-means approach for incomplete data clustering based on nearest-neighbor intervals, Soft Comput. 17 (10) (2013) 1787–1796.
[17] M. Lichman, UCI machine learning repository, http://archive.ics.uci.edu/ml, 2013 (Retrieved: 2017-02-09).
[18] R.J.A. Little, D.B. Rubin, Statistical Analysis with Missing Data, John Wiley and Sons, Incorporated, New York, USA, 1986.
[19] T.D. Little, K.M. Lang, W. Wu, M. Rhemtulla, Missing Data, John Wiley and Sons, Incorporated, 2016.
[20] D.A. Newman, Missing data, Organ. Res. Methods 17 (4) (2014) 372–411.
[21] A.C. Pegis, Cosmogony and knowledge the dilemma of composite essences, J. Thought 19 (1944) 269–290.
[22] G. Peters, F. Crespo, P. Lingras, R. Weber, Soft clustering – fuzzy and rough approaches and their extensions and derivatives, Int. J. Approx. Reason. 54 (2) (2013) 307–322.
[23] D.B. Rubin, Multiple imputations in sample surveys – a phenomenological Bayesian approach to nonresponse, in: Proceedings of the Section on Survey Research Methods, American Statistical Association, 1978, pp. 20–34.
[24] J.L. Schafer, J.W. Graham, Missing data: our view of the state of the art, Psychol. Methods 7 (2) (2002) 147–177.
[25] E. Schubert, A. Koos, T. Emrich, A. Züfle, K.A. Schmid, A. Zimek, A framework for clustering uncertain data, Proc. VLDB Endow. 8 (12) (2015) 1976–1979.
[26] Y. Shoham, Computer science and game theory, Commun. ACM 51 (8) (2008) 74–79.
[27] H. Timm, C. Döring, R. Kruse, Different approaches to fuzzy clustering of incomplete datasets, Int. J. Approx. Reason. 35 (3) (2004) 239–249.
[28] I.R. White, J.P.T. Higgins, A.M. Wood, Allowing for uncertainty due to missing data in meta-analysis—part 1: two-stage methods, Stat. Med. 27 (5) (2008) 711–727.
[29] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Ann. Data Sci. 2 (2) (2015) 165–193.
[30] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678.
[31] J.T. Yao, J.P. Herbert, A game-theoretic perspective on rough set analysis, J. Chongqing Univ. Posts Telecommun. (Nat. Sci. Ed.) 20 (3) (2008) 291–298.
[32] Y.Y. Yao, An outline of a theory of three-way decisions, in: Proceedings of 8th International Conference on Rough Sets and Current Trends in Computing, RSCTC 2012, in: Lecture Notes in Computer Science, vol. 7413, 2012, pp. 1–17.

[33] Y.Y. Yao, Rough sets and three-way decisions, in: Proceedings of 10th International Conference on Rough Sets and Knowledge Technology, RSKT 2015, in: Lecture Notes in Computer Science, vol. 9436, 2015, pp. 62–73.
[34] S. Yin, Z. Huang, Performance monitoring for vehicle suspension system via fuzzy positivistic c-means clustering based on accelerometer measurements, IEEE/ASME Trans. Mechatron. 20 (5) (2015) 2613–2620.
[35] H. Yu, A framework of three-way cluster analysis, in: Proceedings of 2nd International Joint Conference on Rough Sets, IJCRS 2017, in: Lecture Notes in Computer Science, vol. 10313, 2017, pp. 300–312.
[36] H. Yu, T. Su, X.H. Zeng, A three-way decisions clustering algorithm for incomplete data, in: Proceedings of 9th International Conference on Rough Sets and Knowledge Technology, RSKT 2014, in: Lecture Notes in Computer Science, vol. 8818, 2014, pp. 765–776.
[37] H. Yu, C. Zhang, G. Wang, A tree-based incremental overlapping clustering method using the three-way decision theory, Knowl.-Based Syst. 91 (2016) 189–203.
[38] L. Zhang, B. Li, L. Zhang, D. Li, Fuzzy clustering of incomplete data based on missing attribute interval size, in: Proceedings of 9th International Conference on Anti-Counterfeiting, Security, and Identification, ASID 2015, 2015, pp. 101–104.
[39] L.Y. Zhang, W. Lu, X.D. Liu, W. Pedrycz, C.Q. Zhong, Fuzzy c-means clustering of incomplete data based on probabilistic information granules of missing values, Knowl.-Based Syst. 99 (2016) 51–70.
[40] R.U. Zuo, Z.J. Zhang, D.J. Zhang, E.J.M. Carranza, H.H. Wang, Evaluation of uncertainty in mineral prospectivity mapping due to missing evidence: a case study with skarn-type Fe deposits in Southwestern Fujian Province, China, Ore Geol. Rev. 71 (2015) 502–515.