Available online at www.sciencedirect.com

ScienceDirect

Data & Knowledge Engineering 63 (2007) 879–893

DATA & KNOWLEDGE ENGINEERING

www.elsevier.com/locate/datak

MMR: An algorithm for clustering categorical data using Rough Set Theory

Darshit Parmar, Teresa Wu *, Jennifer Blackhurst

Department of Industrial Engineering, PO Box 875906, Arizona State University, Tempe, AZ 85287-5906, USA

Received 5 August 2006; received in revised form 18 April 2007; accepted 29 May 2007
Available online 13 June 2007

Abstract

A variety of cluster analysis techniques exist to group objects having similar characteristics. However, the implementation of many of these techniques is challenging due to the fact that much of the data contained in today's databases is categorical in nature. While there have been recent advances in algorithms for clustering categorical data, some are unable to handle uncertainty in the clustering process while others have stability issues. This research proposes a new algorithm for clustering categorical data, termed Min–Min-Roughness (MMR), based on Rough Set Theory (RST), which has the ability to handle the uncertainty in the clustering process.
Published by Elsevier B.V.

Keywords: Cluster analysis; Categorical data; Rough Set Theory; Data mining

1. Introduction

Cluster analysis is a data analysis tool used to group data with similar characteristics. It has been used in data mining tasks such as unsupervised classification and data summation, as well as segmentation of large heterogeneous data sets into smaller homogeneous subsets that can be easily managed, separately modeled and analyzed [12]. The basic objective in cluster analysis is to discover natural groupings of objects [14]. Cluster analysis techniques have been used in many areas such as manufacturing, medicine, nuclear science, radar scanning and research and development planning. For example, Jiang et al. [13] analyze a variety of cluster techniques for complex gene expression data. Wu et al. [40] develop a clustering algorithm specifically designed to handle the complexities of gene data that can estimate the correct number of clusters and find them. Wong et al. [39] present an approach used to segment tissues in a nuclear medical imaging method known as positron emission tomography (PET). Mathieu and Gibson [26] use cluster analysis as a part of a decision support tool for large-scale research and development planning to identify programs to participate in and to determine resource allocation. Finally, Haimov et al. [8] use cluster analysis to segment radar signals in scanning land and marine objects.

* Corresponding author.
E-mail address: teresa.wu@asu.edu (T. Wu).

0169-023X/$ - see front matter Published by Elsevier B.V.
doi:10.1016/j.datak.2007.05.005

A problem with many of the clustering methods and applications mentioned above is that they are applicable for clustering data having numerical values for attributes. Most of the work in clustering is focused on attributes with numerical value due to the fact that it is relatively easy to define similarities from the geometric position of the numerical data. Unlike numerical data, categorical data have multi-valued attributes. Thus, similarity can be defined as common objects, common values for the attributes, and the association between the two. In such cases, the horizontal co-occurrences (common attributes for the objects) as well as the vertical co-occurrences (common values for the attributes) can be examined [40].

A number of algorithms for clustering categorical data have been proposed including work by Huang [12], Gibson et al. [5], Guha et al. [6], Ganti et al. [4], and Dempster et al. [2]. While these methods make important contributions to the issue of clustering categorical data, they are not designed to handle uncertainty in the clustering process. This is an important issue in many real world applications where there is often no sharp boundary between clusters. Recently, there has been work in the area of applying fuzzy sets in clustering categorical data including work by Huang [12] and Kim et al. [16]. However, these algorithms require multiple runs to establish the stability needed to obtain a satisfactory value for one parameter used to control the membership fuzziness.

Therefore, there is a need for a robust clustering algorithm that can handle uncertainty in the process of clustering categorical data. This research proposes a clustering algorithm based on Rough Set Theory (RST). The proposed algorithm, named Min–Min-Roughness (MMR), is designed to deal with uncertainty in the process of clustering categorical data. In addition, the algorithm is implemented and tested with three real world data sets. To compare the algorithm performance in handling uncertainty, Soybean and Zoo data sets are used and the results are compared with fuzzy set theory based algorithms (including *K*-modes, fuzzy *K*-modes and fuzzy centroids). To test the applicability to large scale date sets, the Mushroom data set is used and the results are compared with Squeezer, *K*-modes and LCBCDC, as well as ROCK and a traditional hierarchical algorithm. The contributions of our proposed approach include:

(1) Unlike previous methods, MMR gives the user the ability to handle uncertainty in the clustering process.
(2) Using MMR, the user is able to obtain stable results given only one input: the number of clusters.
(3) MMR has the capability of handling large data sets.

This paper is structured as follows: Section 2 presents an overview of standard clustering methods existing in the literature. In Section 3, the basics of the rough set theory are introduced followed by the proposed MMR algorithm. A synthetic data set is used to illustrate the MMR algorithm. Section 4 discusses the implementation of the algorithm and the results from the application of the algorithm on Soybean, Zoo and Mushroom data sets (from the UCI Machine Learning Repository[1]). In addition, the comparison results are analyzed. Section 5 presents conclusions and identifies future research directions.

2. Literature review

In this section, an overview of methods available in the literature to cluster categorical data is presented. Ralambondrainy [33] proposes a method to convert multiple category attributes into binary attributes using 0 and 1 to represent either a category absence or presence, and to treat the binary attributes as numeric in the *K*-means algorithm. Dempster et al. [2] presents a partitional clustering method, called the Expectation-Maximization (EM) algorithm. EM first randomly assigns different probabilities to each class or category, for each cluster. These probabilities are then successively adjusted to maximize the likelihood of the data given the specified number of clusters. Since the EM algorithm computes the classification probabilities, each observation belongs to each cluster with a certain probability. The actual assignment of observations to a cluster is determined based on the largest classification probability. After a large number of iterations, EM terminates at a locally optimal solution. Han et al. [9] propose a clustering algorithm to cluster related items in a market database based on an association rule hypergraph. A hypergraph is used as a model for relatedness. The

[1] Available at http://www.ics.uci.edu/~mlearn/MLRepository.html.

approach targets binary transactional data. It assumes item sets that define clusters are disjoint and there is no overlap amongst them. However, this assumption may not hold in practice as transactions in different clusters may have a few common items. *K*-modes [12] extends *K*-means and introduces a new dissimilarity measure for categorical data. The dissimilarity measure between two objects is calculated as the number of attributes whose values do not match. The *K*-modes algorithm then replaces the means of clusters with modes, using a frequency based method to update the modes in the clustering process to minimize the clustering cost function. One advantage of *K*-modes is it is useful in interpreting the results [12]. However, *K*-modes generates local optimal solutions based on the initial modes and the order of objects in the data set. *K*-modes must be run multiple times with different starting values of modes to test the stability of the clustering solution. Huang [12] also proposes the *K*-prototypes algorithm, which allows clustering of objects described by a combination of numeric and categorical data. CACTUS (Clustering Categorical Data Using Summaries) [4] is a summarization based algorithm. In CACTUS, the authors cluster for categorical data by generalizing the definition of a cluster for numerical attributes. Summary information constructed from the data set is assumed to be sufficient for discovering well-defined clusters. CACTUS finds clusters in subsets of all attributes and thus performs a subspace clustering of the data. Guha et al. [6] propose a hierarchical clustering method termed ROCK (Robust Clustering using Links), which can measure the similarity or proximity between a pair of objects. Using ROCK, the number of "links" are computed as the number of common neighbors between two objects. An agglomerative hierarchical clustering algorithm is then applied: first, the algorithm assigns each object to a separate cluster, clusters are then merged repeatedly according to the closeness between clusters, where the closeness is defined as the sum of the number of "links" between all pairs of objects. Gibson et al. [5] propose an algorithm called STIRR (Sieving Through Iterated Relational Reinforcement), a generalized spectral graph partitioning method for categorical data. STIRR is an iterative approach, which maps categorical data to non-linear dynamic systems. If the dynamic system converges, the categorical data can be clustered. Clustering naturally lends itself to combinatorial formulation. However, STIRR requires a non-trivial post-processing step to identify sets of closely related attribute values [4]. Additionally, certain classes of clusters are not discovered by STIRR [4]. Moreover, Zhang et al. [41] argue that STIRR cannot guarantee convergence and therefore propose a revised dynamic system algorithm that assures convergence. He et al. [11] propose an algorithm called Squeezer, which is a one-pass algorithm. Squeezer puts the first-tuple in a cluster and then the subsequent-tuples are either put into an existing cluster or rejected to form a new cluster based on a given similarity function. He et al. [10] explore categorical data clustering (CDC) and link clustering (LC) problems and propose a LCBCDC (Link Clustering Based Categorical Data Clustering), and compare the results with Squeezer and *K*-mode. In reviewing these algorithms, some of the methods such as STIRR and EM algorithms cannot guarantee the convergence while others have scalability issues. In addition, all of the algorithms have one common assumption: each object can be classified into only one cluster and all objects have the same degree of confidence when grouped into a cluster [7]. However, in real world applications, it is difficult to draw clear boundaries between the clusters. Therefore, the uncertainty of the objects belonging to the cluster needs to be considered.

One of the first attempts to handle uncertainty is fuzzy *K*-means [34]. In this algorithm, each pattern or object is allowed to have membership functions to all clusters rather than having a distinct membership to exactly one cluster. Krishnapuram and Keller [18] propose a probabilistic approach to clustering in which the membership of a feature vector in a class has nothing to do with its membership in other classes and modified clustering methods are used to generate membership distributions. Krishnapuram et al. [17] present several fuzzy and probabilistic algorithms to detect linear and quadratic shell clusters. Note the initial work in handling uncertainty was based on numerical data. Huang [12] proposes a fuzzy *K*-modes algorithm with a new procedure to generate the fuzzy partition matrix from categorical data within the framework of the fuzzy *K*-means algorithm. The method finds fuzzy cluster modes when a simple matching dissimilarity measure is used for categorical objects. By assigning confidence to objects in different clusters, the core and boundary objects of the clusters can be decided. This helps in providing more useful information for dealing with boundary objects. More recently, Kim et al. [16] have extended the fuzzy *K*-modes algorithm by using fuzzy centroids to represent the clusters of categorical data instead of the hard-type centroids used in the fuzzy *K*-modes algorithm. The use of fuzzy centroids makes it possible to fully exploit the power of fuzzy sets in representing the uncertainty in the classification of categorical data. However, fuzzy *K*-modes and fuzzy centroids algorithms suffer from the same problem as K-modes, that is they require multiple runs with different starting values of modes to test the stability of the clustering solution. In addition, these algorithms have to adjust one control parameter for membership fuzziness to obtain better solutions. This necessitates the effort for multiple runs of these algorithms to determine an acceptable value of this parameter. Therefore, there is a need for a categorical data clustering method, having the ability to handle uncertainty in the clustering process while providing stable results. One methodology with potential for handling uncertainty is Rough Set Theory (RST) which has received considerable attention in the computational intelligence literature since its development by Pawlak in the 1980s. Unlike fuzzy set based approaches, rough sets have no requirement on domain expertise to assign the fuzzy membership. Still, it may provide satisfactory results for rough clustering. The objective of this research is to develop a rough set based approach for categorical data clustering. The approach, termed Min–Min-Roughness (MMR), is presented and its performance is evaluated on large scale data sets.

3. Min–Min-Roughness (MMR) algorithm

3.1. Nomenclature

U	universe or the set of all objects $(x_1, x_2, \ldots)$
X	subset of the set of all objects, $(X \subset U)$
x_i	object belonging to the subset of the set of all objects, $x_i \in X$
A	the set of all attributes (features or variables)
a_i	attribute belonging to the set of all attributes, $a_i \in A$
$V(a_i)$	set of values of attribute a_i (or called domain of a_i)
B	non-empty subset of $A(B \subseteq A)$
$\underline{X_B}$	lower approximation of X with respect to B
$\overline{X_B}$	upper approximation of X with respect to B
$R_{a_i}(X)$	roughness with respect to $\{a_i\}$
$\mathrm{Rough}_{a_j}(a_i)$	mean roughness on attribute a_i with respect to $\{a_j\}$
MR (a_i)	minimum roughness of attribute a_i
MMR	minimum of MR of all attributes
Ind(B)	indiscernibility relation
$[x_i]_{\mathrm{Ind}(B)}$	equivalence class of x_i in relation Ind(B), also known as elementary set in B.

3.2. Rough Set Theory (RST)

RST is an approach to aid decision making in the presence of uncertainty [30,31]. It classifies imprecise, uncertain or incomplete information expressed in terms of data acquired from experience. In RST, a set of all similar objects is called an elementary set, which makes a fundamental atom of knowledge [29]. Any union of elementary sets is called a crisp set and other sets are referred to as rough set [29]. As a result of this definition, each rough set has boundary-line elements. For example, some elements cannot be definitively classified as members of the set or its complement. In other words, when the available knowledge is employed, boundary-line cases cannot be properly classified. Therefore, rough sets can be considered as uncertain or imprecise. Upper and lower approximations are used to identify and utilize the context of each specific object and reveal relationships between objects. The upper approximation includes all objects that possibly belong to the concept while the lower approximation contains all objects that surely belong to the concept.

Let U be the set of all objects, A be the set of all attributes, B be a non-empty subset of A, V be the set of all attribute values and $U \times A \rightarrow V$ be an information function.

Definition 1 (*Indiscernibility relation* (*Ind*(B))). 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, $(x_i, x_j \in \mathrm{Ind}(B)$ if and only if $\forall a \in B$ where $B \subseteq A$, $a(x_i) = a(x_j)$.

Definition 2 (*Equivalence class* ($[x_i]_{\mathrm{Ind}(B)}$)). Given Ind(B), the set of objects x_i having the same values for the set of attributes in B consists of an equivalence classes, $[x_i]_{\mathrm{Ind}(B)}$. It is also known as elementary set with respect to B.

Definition 3 (*Lower approximation*). Given the set of attributes B in A, set of objects X in U, the lower approximation of X is defined as the union of all the elementary sets which are contained in X. That is,

$$\underline{X_B} = \cup\{x_i | [x_i]_{\mathrm{Ind}(B)} \subseteq X\} \tag{1}$$

Definition 4 (*Upper approximation*). Given the set of attributes B in A, set of objects X in U, the upper approximation of X is defined as the union of the elementary sets which have a non-empty intersection with X. That is,

$$\overline{X_B} = \cup\{x_i | [x_i]_{\mathrm{Ind}(B)} \cap X \neq \emptyset\} \tag{2}$$

Definition 5 (*Roughness*). The ratio of the cardinality of the lower approximation and the cardinality of the upper approximation is defined as the accuracy of estimation, which is a measure of roughness. It is presented as

$$R_B(X) = 1 - \frac{|\underline{X_B}|}{|\overline{X_B}|} \tag{3}$$

If $R_B(X) = 0$, X is *crisp* with respect to B, in other words, X is precise with respect to B. If $R_B(X) < 1$, X is *rough* with respect to B, that is, B is *vague* with respect to X. It is this measure, roughness, which allows an object to belong to a cluster with different degrees of belonging using the MMR algorithm. In other words, MMR has the ability to deal with uncertainty with the calculation of lower bound and upper bound that gives a degree of belonging rather than having the same degree of belonging to all objects.

Recently, RST has found many different applications. Examples include semiconductor manufacturing [20,35], the automobile industry [21], business failure predictions [3], customer retention [19], intelligent image filtering [42], clinical databases [36], classification of highway sections [22], and web mining [15] just to name a few. RST has been used extensively for supervised learning with a focus on classification problems, where prior group membership is known. Results generated usually are rules for group membership [32]. Clustering within the context of RST is attracting increasing interest. Lingras [22] explores how to use a rough set genome to represent a rough set theoretic classification scheme. Later, the modified K-means algorithm [24] and Kohonen Neural Network [23] are proposed to create intervals of clusters based on RST. Furthermore, considering the possibility that one object may belong to more than one cluster, Lingras et al. [25] investigate three methodologies (genetic algorithms, K-means and Kohonen Self-Organizing Maps) for clustering based on the properties of rough sets for developing the lower-upper bound representation of the clusters. Voges et al. [37] propose a technique called rough clustering, which is a simple extension of RST, and apply it to the problem of market segmentation. However, the majority of research in exploring RST for clustering aims to handle numerical data sets where distance can be easily derived from the data set. Instead of defining distance between objects, MMR is developed for categorical data based on the concept of roughness.

3.3. Min–Min-Roughness (MMR)

Mazlack et al. [27] attempt to use RST for choosing partitioning attributes for clustering. They use a measure called total roughness to determine the crispness of the partition. However, for partitioning, the method starts with binary valued attributes and uses the total roughness criterion only for multi-valued attributes. This creates from a handicap due to the fact that the partitioning is done on a binary attribute even though the total roughness for a multi-valued attribute is lower. MMR overcomes this drawback by clustering the objects on all attributes. In addition, MMR proposes a new way to measure data similarities based on the roughness concept. MMR utilizes a measure termed mean roughness comparable to that proposed by Mazlack et al. [27] based on RST. This is reproduced below:

Definition 6 (*Mean roughness* [27]). Given $a_i \in A$, $V(a_i)$ refers to the set of values of attribute a_i, X is a subset of objects having one specific value, α, of attribute a_i, that is, $X(a_i = \alpha)$, $\underline{X}_{a_j}(a_i = \alpha)$ refers to the lower approximation, and $\overline{X}_{a_j}(a_i = \alpha)$ refers to the upper approximation with respect to $\{a_j\}$, then $R_{a_j}(X)$ is defined as the roughness of X with respect to $\{a_j\}$, that is

$$R_{a_j}(X|a_i = \alpha) = 1 - \frac{|\underline{X}_{a_j}(a_i = \alpha)|}{|\overline{X}_{a_j}(a_i = \alpha)|}, \quad \text{where } a_i, a_j \in A \quad \text{and} \quad a_i \neq a_j \tag{4}$$

Let $|V(a_i)|$ be the number of values of attributes a_i, the mean roughness on attribute a_i with respect to $\{a_j\}$ is defined as

$$\text{Rough}_{a_j}(a_i) = \frac{R_{a_j}(X|a_i = \alpha_1) + \cdots R_{a_j}(X|a_i = \alpha_{|V(a_i)|})}{|V(a_i)|}, \tag{5}$$

where a_i, $a_j \in$ A and $a_i \neq a_j$.

Next, Min-Roughness (MR) and Min–Min-Roughness (MMR) developed in this research are introduced.

Definition 7 (*Min-Roughness* (*MR*)). Given n attributes, MR, min-roughness of attribute a_i ($a_i \in$ A) refers to the minimum of the mean roughness, that is,

$$\text{MR}(a_i) = \text{Min } (\text{Rough}_{a_1}(a_i), \ldots \text{Rough}_{a_j}(a_i) \ldots, \quad \text{where } a_i, a_j \in A, \;\; a_i \neq a_j, \;\; 1 \leqslant i, \;\; j \leqslant n \tag{6}$$

Definition 8 (*Min–Min-Roughness* (*MMR*)). Given n attributes, the MMR is defined as the minimum of the Min-Roughness of the n attributes. That is,

$$\text{MMR} = \text{Min}(\text{MR}(a_1), \ldots \text{MR}(a_i), \ldots \quad \text{where } a_i \in A, \;\; i \text{ goes from 1 to cardinality } (A) \tag{7}$$

The lower the mean roughness is, the higher the crispness of the clustering. Min-Roughness (MR) (Definition 7) determines the best crispness each attribute can achieve. MMR (Definition 8) determines the best split on the attributes. The MMR algorithm (as shown in Table 1) iteratively divides the group of objects with the goal of achieving better clustering crispness. The algorithm takes the number of clusters, k, as one input and will terminate when this pre-defined number k, is reached.

Next, we present an illustrative example of the MMR algorithm.

[*Illustrative Example]*: Table 2 introduces a data set (I) used to illustrate the application of the MMR algorithm. There are ten objects ($m = 10$) and six attributes ($n = 6$). The maximum number of values is 4 ($l = 4$). Our interest is to create clusters of similar objects. As seen from the data set in Table 2, variables can be multi-valued. That is, the domain of an attribute can contain more than two distinct values.

First, the mean roughness on each attributes a_i ($i = 1, \ldots, 6$) is calculated. Let us take attribute a_1 as an example. The mean roughness on a_1 with respect to $\{a_2\}$ is calculated as following. There are three elementary sets for a_1: X (a_1 = Small) = $\{3, 5, 7, 8\}$, X (a_1 = Medium) = $\{2, 4, 10\}$ and X (a_1 = Big) = $\{1, 6, 9\}$. There are four elementary sets for a_2: $X(a_2$ = Blue) = $\{1, 4\}$, $X(a_2$ = Red) = $\{2\}$, $X(a_2$ = Yellow) = $\{3, 5, 7, 8\}$ and $X(a_2$ = Green) = $\{6, 9, 10\}$. According to Definition 3 and 4, the lower approximation of X (a_1 = Small) is $\{3, 5, 7, 8\}$ and the upper approximation is the same, the lower approximation of X (a_1 = Medium) is $\{2\}$ and the upper approximation is $\{1, 2, 4, 6, 9, 10\}$, the lower approximation of X (a_1 = Big) is empty thus there is no need of calculating the upper approximation. According to Definition 6, the mean roughness on a_1 with respect to $\{a_2\}$ is 0.6111. Following the same procedure, the mean roughness on a_1 with respect to $\{a_3\}$, $\{a_4\}$, $\{a_5\}$, $\{a_6\}$ is computed. These calculations are summarized in Table 3. Similar calculations are performed for all the attributes.

Second, the partitioning attribute with the MMR is found. Table 4 shows the calculations and illustrates that attribute a_1 and a_3 have the same MMR. In using the algorithm, it is recommended to look at the next lowest MMR inside the attributes that are tied and so on until the tie is broken. In the case where all the numbers are tied, selecting any attribute randomly can break the tie. In the example, the second MMR corresponding to attribute a_1 is lower than that of a_3. Therefore, attribute a_1 is selected as the partitioning attribute and binary splitting is conducted.

Table 1
MMR Algorithm

```
Procedure MMR(U, k)
Begin
Set current number of cluster CNC = 1
Set ParentNode = U
Label 1:
If CNC < $k$ and CNC $\neq$ 1 then
    ParentNode = ProcParentNode (CNC)
End if
    // Clustering the ParentNode
    For each $a_i \in A$ ($i = 1$ to $n$, where $n$ is the number of attributes in $A$)
        Determine $[x_i]_{\mathrm{Ind}(a_i)}$
        For each $a_j \in A$($j = 1$ to $n$, where $n$ is the number of attributes in $A$, $j \neq i$)
            Calculate $\mathrm{Rough}_{a_j}(a_i)$
        Next
        Min-Roughness ($a_i$) = Min ($\mathrm{Rough}_{a_j}(a_i)$)
    Next
    Set Min–Min-Roughness = Min (Min-Roughness ($a_i$)), $i = 1,\ldots,n$
    Determine splitting attribute $a_i$ corresponding to the Min–Min-Roughness
    Do binary split on the splitting attribute $a_i$
    CNC = the number of leaf nodes
    Go to Label 1
End
ProcParentNode (CNC)
Begin
    Set $i = 1$
    Do until $i$ < CNC
        Size ($i$) = Count (Set of Elements in Cluster $i$)
        $i = i + 1$
    Loop
    Determine Max (Size ($i$))
    Return (Set of Elements in cluster $i$) corresponding to Max (Size ($i$))
End
```

Table 2
Example data set (I)

Rows	a_1	a_2	a_3	a_4	a_5	a_6
1	Big	Blue	Hard	Indefinite	Plastic	Negative
2	Medium	Red	Moderate	Smooth	Wood	Neutral
3	Small	Yellow	Soft	Fuzzy	Plush	Positive
4	Medium	Blue	Moderate	Fuzzy	Plastic	Negative
5	Small	Yellow	Soft	Indefinite	Plastic	Neutral
6	Big	Green	Hard	Smooth	Wood	Positive
7	Small	Yellow	Hard	Indefinite	Metal	Positive
8	Small	Yellow	Soft	Indefinite	Plastic	Positive
9	Big	Green	Hard	Smooth	Wood	Neutral
10	Medium	Green	Moderate	Smooth	Plastic	Neutral

Table 3
Mean roughness calculation for attribute a_1

	X_1 (Small)	X_2 (Medium)	X_3 (Big)	Mean roughness
With respect to $\{a_2\}$	0	0.8333	1	0.6111
With respect to $\{a_3\}$	0.5714	0	1	0.5238
With respect to $\{a_4\}$	1	1	1	1
With respect to $\{a_5\}$	0.7143	1	1	0.9048
With respect to $\{a_6\}$	1	1	1	1

Table 4
MMR calculation

Attributes	Mean roughness	Min roughness
a_1	$\text{Rough}_{a_j}(a_1), j = 23456$ (0.6111, 0.5238, 1, 0.9048, 1)	**Min: 0.5238 Second Min: 0.6111**
a_2	$\text{Rough}_{a_j}(a_2), j = 1, 3, 4, 5, 6$ (0.7500, 0.8929, 1, 0.9286, 0.7500)	0.7500
a_3	$\text{Rough}_{a_j}(a_3), j = 1, 2, 4, 5, 6$ (0.5238, 0.9444, 1, 0.9074, 1)	**Min: 0.5238 Second Min: 0.9074**
a_4	$\text{Rough}_{a_j}(a_4), j = 1, 2, 3, 5, 6$ (1, 0.6667, 1, 0.7639, 1)	0.6667
a_5	$\text{Rough}_{a_j}(a_5), j = 1, 2, 3, 4, 6$ (1, 0.8820, 1, 1, 0.9500)	0.8820
a_6	$\text{Rough}_{a_j}(a_6), j = 1, 2, 3, 4, 5$ (1, 0.6250, 1, 1, 0.9333)	0.6250

Third, the splitting point on attributes a_1 is determined. Note the binary optimal partition problem is NP-hard [28]. MMR determines the splitting point using the heuristic based on the roughness calculation that simplifies the computational complexity. The splitting set should include the attribute value which has minimum roughness. Taking a look at Table 3, X ($a_1 =$ Small) has overall minimum roughness with respect to $\{a_i\}$ ($i = 2, 3, 4, 5, 6$) comparing to X ($a_1 =$ Medium) and X ($a_1 =$ Big). Thus, splitting on X ($a_1 =$ Small) vs. X ($a_1 =$ Medium) and X ($a_1 =$ Big) is chosen. The partition at this stage can be represented as a tree and is shown in Fig. 1.

The numbers in the parenthesis at each of the child nodes correspond to the objects in the original data set. Set (1, 2, 4, 6, 9, 10) corresponds to all objects having either *Big* or *Medium*a s the value for attribute a_1 and set (3, 5, 7, 8) corresponds to all objects having *Small* as value for attribute a_1. The algorithm is applied recursively to obtain further partitions. At subsequent iterations, the leaf node having more objects is selected for further splitting. The algorithm terminates when it reaches a pre-defined number of clusters. This is subjective and is pre-decided based either on user requirement or domain knowledge.

Next we present a comparison with the MMR algorithm with the approach by Mazlack et al. [27].

[*Comparison example*]: As an extension of the approach proposed by Mazlack et al. [27], MMR provides better solution by considering all attributes (bi-valued, multi-valued) equally. Consider the data set (II) shown in Table 5.

Following the procedures calculating MMR, the results are summarized in Table 6.

Clearly, the approach proposed by Mazlack will partition on attribute a_1 (“Volume”) since it is binary (i.e., has only two distinct attribute values). However, MMR will partition on attribute a_2 (“Material”). Overall, the MMR algorithm will result in a crisper solution.

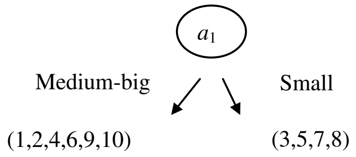

Fig. 1. Partition after first iteration.

Table 5
Example data set (II)

Volume (a_1)	Material (a_2)	Location (a_3)
High	Hard	Pacific
High	Hard	Midwest
High	Medium	East Coast
High	Soft	International
High	Soft	Pacific
Low	Hard	Midwest
Low	Medium	East Coast
Low	Soft	International

Table 6
MMR results on example data set (II)

Attributes	Mean roughness	Min roughness
a_1	$\text{Rough}_{a_j}(a_1) = (1, 0.95), j = 2,3$	0.95
a_2	$\text{Rough}_{a_j}(a_2) = (1, 0.56), j = 1,3$	0.56
a_3	$\text{Rough}_{a_j}(a_3) = (1, 0.92), j = 1,2$	0.92

Thus, given a data set, assume n is the number of attributes, m is the number of objects, k is the chosen number of clusters and l is the maximum number of values in the attribute domains. $k - 1$ iterations are required to achieve k clusters from the data set. In each iteration, the time to find all the elementary sets of each attribute is $n * m$, the time to calculate the mean roughness is approximately n^2l, the time to calculate MR and MMR is 2n. Thus, the computation complexity is polynomial, that is $O(knm + kn^2l)$. For any larger data set with an increasing number of objects (m) and an increasing number of attributes (n), the computation time increases by $m * n$. Given the minimum roughness over all other attributes exists on one particular value (e.g., p) of the attribute (e.g., a_i), the splitting point can be set as ($a_i = p$) and ($a_i \neq p$). Since application of RST in clustering is relatively new, our focus has been on evaluating the performance of MMR. Looking in the future, with the ever-increasing computing capabilities, computation complexity may not be an issue. However, future plans will include efforts to reduce the complexity of the MMR algorithm. For example, we will explore the use of special data structure to reduce the computation effort.

The following section describes the implementation and the results obtained from the application of the MMR algorithm on three real world data sets. It also includes the results of comparison of the MMR algorithm with fuzzy set theory based algorithms and a traditional hierarchical algorithm.

4. Experimental analysis

In order to test MMR, a prototype implementation system is developed using VB.Net and tested on several data sets obtained from the UCI Machine Learning Repository. Validating clustering results is a non-trivial task. The purity of clusters was used as a measure to test the quality of the clusters. The purity of a cluster is defined as

$$\text{Purity}(i) = \frac{\text{the number of data occuring in both the } i\text{th cluster and its corresponding class}}{\text{the number of data in the data set}} \tag{8}$$

The overall purity is defined as

$$\text{Over all Purity} = \frac{\sum_{i=1}^{\#\text{ of clusters}} \text{Purity}(i)}{\#\text{ of clusters}} \tag{9}$$

According to this measure, a higher value of overall purity indicates a better clustering result, with perfect clustering yielding a value of 1. Note that similar measures have been used in Kim et al. [16] and Guha et al. [6].

4.1. Comparison of MMR with algorithms based on fuzzy set theory

Currently, there are only a few algorithms which aim to handle uncertainty in the clustering process. These algorithms are fuzzy set based algorithms and include *K*-modes, fuzzy *K*-modes and fuzzy centroids. *K*-modes uses a dissimilarity measure between two objects which is calculated as the number of attributes whose values do not match. The *K*-modes algorithm then replaces the means of the clusters with modes and uses a frequency based method to update the modes in the clustering process to minimize the clustering cost function. Fuzzy *K*-modes generates a fuzzy partition matrix from categorical data. By assigning a confidence to objects in different clusters, the core and boundary objects of the clusters are determined for clustering purposes. The fuzzy centroids algorithm uses the concept of fuzzy set theory to derive fuzzy centroids to create clusters of objects which have categorical attributes. In this section, MMR is compared with these three algorithms based on Soybean and Zoo data sets in three experiments.

Table 7
Applying MMR on the 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	8	17	0.68
4	0	0	2	0	1
Overall purity					0.83

Experiment 1. The Soybean data set contains 47 objects on diseases in soybeans. Each object can be classified as one of the four diseases namely, Diaporthe Stem Canker, Charcoal Rot, Rhizoctonia Root Rot, and Phytophthora Rot and is described by 35 categorical attributes. The data set is comprised 17 objects for Phytophthora Rot disease and 10 objects for each of the remaining diseases. Since there are four possible class values (four diseases), the algorithms based on fuzzy set theory generate four clusters. For comparison purposes the number of clusters is set to 4 for MMR. The results are summarized in Table 7. Out of 47 objects, 39 belong to the majority class label of the cluster in which they are classified. Thus, the overall purity of the clusters is 83%.

Experiment 2. The Zoo data set is comprised of 101 objects, where each data point represents information of an animal in terms of 18 categorical attributes. Each animal data point is classified into seven classes. Therefore, stopping criterion for MMR is set at seven clusters. Table 8 summarizes the results of running the MMR algorithm on the Zoo data set. Out of 101 objects, 92 belong to the majority class label of the cluster in which they are classified. Thus, the overall purity of the clusters is 91%.

Kim et al. [16] have applied the fuzzy centroid method to the Soybean and Zoo data sets and compared the results with *K*-modes and fuzzy *K*-modes. In this research, MMR is applied to the same data sets and therefore can be used to compare with all three algorithms. Similar to Kim et al. [16], purity is used as the evaluation criteria for comparison study. Table 9 summarizes the results of comparison study.

As shown in Table 9, MMR out performs *K*-modes and fuzzy *K*-modes on both the data sets as well as performing better than the fuzzy centroid method on the Zoo data set. The performance of MMR is comparable to the performance of algorithms based on the fuzzy set theory. As discussed above, fuzzy set based algorithms all face a challenging issue, namely stability. These algorithms require great effort to adjust the parameter, which is used to control the fuzziness of membership of each data point. Therefore, the algorithms

Table 8
Applying MMR on 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
Overall purity								0.91

Table 9
Purity comparison of MMR with algorithms based on Fuzzy Set Theory

Data set	*K*-modes	Fuzzy *K*-modes	Fuzzy centroids	MMR
Soybean	0.69	0.77	0.97	0.83
Zoo	0.60	0.64	0.75	0.91

need to be run at different values of the parameter. At each value of this parameter, the algorithms need to be run multiple times to achieve a stable solution. In these regards, MMR has no such issues and offers the following advantages:

1. MMR needs just one input parameter namely the number of clusters.
2. MMR does not require multiple iterations with different starting values.

4.2. MMR tested on a large data set

Both Soybean and Zoo data sets are relatively small (47, 101 objects, respectively). To test the applicability of MMR on a larger data set, additional tests on the Mushroom data set (8124 objects) are performed comparing MMR to first the Squeezer, *K*-modes and LCBCDC methods and second to a traditional hierarchical algorithm in two experiments. Squeezer is a one-pass algorithm which places the first-tuple in a cluster and then the subsequent-tuples are either put into an existing cluster or rejected to form a new cluster based on a given similarity function. *K*-modes, as previously discussed, uses a dissimilarity measure between two objects which is calculated as the number of attributes whose values do not match. LCBCDC uses Link Clustering (LC) and Categorical Data Clustering (CDC). Finally, the traditional hierarchical algorithm converts the categorical into boolean attributes with 0/1 values. Euclidean distance is used as the distance measure between the centroids of clusters. Pairs of clusters whose centroids or means are the closest are then successively merged until the desired number of clusters remain.

The Mushroom data set contains 8124 objects where each object contains information of a single mushroom. There are 22 categorical attributes; each attribute corresponds to a physical characteristic of a mushroom. An object also contains a poisonous or edible class label for a mushroom. The data set has 4208 edible mushrooms and 3916 poisonous mushrooms.

Experiment 3. According to He et al. [10], Squeezer and *K*-modes algorithms produce better clustering output than other algorithms in categorical data sets with respect to clustering purity. LCBCDC was compared with Squeezer and *K*-modes in He et al. [10] based on the Mushroom data set with two clusters created by each algorithm. For comparison purposes, MMR uses two clusters as the stopping criterion. Results are shown in Table 10. Clearly, MMR outperforms the Squeezer and *K*-modes algorithms, which do not handle uncertainty. It is interesting to note that LCBCDC provides better purity (86%). However, the results shown indicate that only partial data set (5478 records out of 8124) is used for LCBCDC. He et al. [10] explain that LCBCDC reasonably discards the malignant records as outliers and presents better cluster results.

Experiment 4. Guha et al. [6] compare ROCK with a traditional hierarchical algorithm based on the Mushroom data set. MMR is applied on the same data set and the results are used to compare MMR to the traditional hierarchical algorithm. As the traditional hierarchical algorithm generates 20 clusters, for comparison purpose, MMR also uses 20 clusters as the stopping criterion. Table 11 summarizes the results of running the MMR algorithm on the Mushroom data set. Out of 8124 mushrooms, 6785 belong to the majority class label of the cluster in which they are classified. Thus, the overall purity of the clusters is 84%.

Table 10
Comparing with Squeezer, *K*-modes and LCBCDC on Mushroom data set

Cluster number	Squeezer		*K*-modes		LCBCDC		MMR	
1	No. of poisonous	3873	No. of poisonous	1856	No. of poisonous	1768	No. of poisonous	1994
	No. of edible	3723	No. of edible	1470	No. of edible	48	No. of edible	260
2	No. of poisonous	43	No. of poisonous	2060	No. of poisonous	712	No. of poisonous	1922
	No. of edible	485	No. of edible	2738	No. of edible	2950	No. of edible	3948
Purity	0.56		0.56		0.86		0.73	

Table 11
Applying MMR on the mushroom data set with 20 clusters

Cluster number	Number of poisonous mushrooms	Number of edible mushrooms	Purity
1	0	164	1
2	36	425	0.92
3	0	640	1
4	0	539	1
5	0	19	1
6	0	1	1
7	151	329	0.68
8	0	14	1
9	115	144	0.56
10	120	111	0.52
11	782	0	1
12	321	1177	0.79
13	6	0	1
14	1	0	1
15	28	0	1
16	0	40	1
17	323	120	0.73
18	1440	88	0.94
19	196	12	0.94
20	397	385	0.51
Overall purity			0.84

Results were then compared with the application of the Mushroom data set on the traditional hierarchical algorithm. One noticeable discrepancy in the results obtained from the traditional hierarchical algorithm is in the total number of mushrooms. The total number of mushrooms turns out to be 7795 instead of 8124 in the original data set. With the assumption that the remaining 329 mushrooms are classified correctly, 4692 out of 8124 mushrooms belong to the majority class label of the cluster in which they are classified. Thus, the overall purity of the clusters is 58%, whereas the purity from MMR is 84%. Clearly, MMR outperforms the traditional hierarchical algorithm. However, results from MMR indicate that there are significant differences in the size of the clusters. This may be attributed to the fact that the leaf node with the largest number of objects to split with at the beginning of each iteration was selected, which leaves leaf nodes with next largest number of objects untouched. Different splitting strategies will be explored in future work. Another reason for the differences in the size of the clusters might be due to the fact that MMR does not remove any outliers. MMR uses the entire data set and the outliers may be affecting the size of the clusters. Therefore, outliers from the data set will be removed in future work.

Results from Guha et al. [6] indicate that ROCK performs very well on the Mushroom data set after the parameters are fine-tuned. With 21 clusters being created, the purity of ROCK is 97%. However, as discussed by Andritsos et al. [1], ROCK has some limitations. ROCK is very sensitive to the threshold value. In many cases, this subjectively determined threshold value could significantly affect the final clustering results. Secondly, ROCK tends to produce one giant cluster that includes objects from most classes. Thirdly, ROCK cannot guarantee the number of generated clusters is the same as what is specified when ROCK is initially launched. For example, the application of the ROCK algorithm on the mushroom data set resulted in 21 clusters even though the input number of clusters was 20. Therefore, a detailed analysis on comparing MMR with ROCK on the Mushroom data set is not provided.

5. Conclusions

Most algorithms designed to cluster categorical data sets are not designed to handle uncertainty in the data set. The majority of the clustering techniques implicitly assume that an object can be classified into, at most, one cluster and that all objects classified into a cluster belong to it with the same degree of confidence. However, in many applications, there can be objects that might have the potential of being classified into more than one cluster. Thus, there is a need for a clustering algorithm that can handle this uncertainty in the clustering process. The MMR algorithm proposed in this paper has potential for clustering categorical attributes in data mining. It is capable of handling the uncertainty in the clustering process. Unlike other algorithms, MMR requires only one input, the number of clusters, and has been tested to be stable.

Additionally, we propose a number of future research activities related to MMR. First, we propose to explore a new stopping criterion. Instead of picking the subset of the data set with the maximum number of objects, we can first determine the distance between the objects falling under each leaf node. The leaf node with the maximum distance between the objects can be picked for splitting at the subsequent iteration. Secondly, we will extend MMR to handle both numerical and categorical data. We will explore a discretization algorithm (such as 4cDiscretizzed which is an unsupervised discretization algorithm to divide the attribute range into a constant number of intervals containing an equal number of the attribute values). Thirdly, to achieve lower computation complexity, we will study the roughness measure based on relationship between a_i and the set defined as A-$\{a_i\}$ instead of calculating the maximum with respect to all $\{a_j\}$ where $a_j \neq a_i$. Fourthly, we propose to study the approach proposed by Voges and Pope [38] and explore the potential application of evolutionary algorithms with rough sets to help determine the number of clusters in advance. Finally, more experiments need to be done on even larger data sets with more objects and more attributes.

References

[1] P. Andritsos, P. Tsaparas, R.J. Miller, K.C. Sevcik, Clustering categorical data based on information loss minimization, in: Second Hellenic Data Management Symposium, 2003, pp. 334–344.

[2] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39 (1) (1977) 1–38.

[3] A.I. Dimitras, R. Slowinski, R. Susmaga, C. Zopounidis, Business failure prediction using rough sets, European Journal of Operational Research 114 (22) (1999) 263–280.

[4] V., Ganti, J. Gehrke, R. Ramakrishnan, CACTUS – clustering categorical data using summaries, in: Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, pp. 73–83.

[5] D. Gibson, J. Kleinberg, P. Raghavan, Clustering categorical data: an approach based on dynamical systems, The Very Large Data Bases Journal 8 (3–4) (2000) 222–236.

[6] S. Guha, R. Rastogi, K. Shim, ROCK: a robust clustering algorithm for categorical attributes, Information Systems 25 (5) (2000) 345–366.

[7] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, Journal of Intelligent Information Systems 17 (2–3) (2001) 107–145.

[8] S. Haimov, M. Michalev, A. Savchenko, O. Yordanov, Classification of radar signatures by autoregressive model fitting and cluster analysis, IEEE Transactions on Geo Science and Remote Sensing 8 (1) (1989) 606–610.

[9] E. Han, G. Karypis, V. Kumar, B. Mobasher, Clustering based on association rule hypergraphs, in: Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997, pp. 9–13.

[10] Z. He, X. Xu, S. Deng, A link clustering based approach for clustering categorical data, Proceedings of the WAIM Conference, 2004. <http://xxx.sf.nchc.org.tw/ftp/cs/papers/0412/0412019.pdf>.

[11] Z. He, X. Xu, S. Deng, Squeezer: an efficient algorithm for clustering categorical data, Journal of Computer Science & Technology 17 (5) (2002) 611–624.

[12] Z. Huang, Extensions to the *k*-means algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2 (3) (1998) 283–304.

[13] D. Jiang, C. Tang, A. Zhang, Cluster analysis for gene expression data: a survey, IEEE Transactions on Knowledge and Data Engineering 16 (11) (2004) 1370–1386.

[14] R. Johnson, W. Wichern, Applied Multivariate Statistical Analysis, Prentice Hall, New York, 2002.

[15] A. Joshi, R. Krishnapuram, Robust fuzzy clustering methods to support web mining, in: Proceedings of the Workshop on Data Mining and Knowledge Discovery, vol. 15, 1998, pp. 1–8.

[16] D. Kim, K. Lee, D. Lee, Fuzzy clustering of categorical data using fuzzy centroids, Pattern Recognition Letters 25 (11) (2004) 1263–1271.

[17] R. Krishnapuram, H. Frigui, O. Nasraoui, Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation, IEEE Transactions on Fuzzy Systems 3 (1) (1995) 29–60.

[18] R. Krishnapuram, J. Keller, A possibilistic approach to clustering, IEEE Transactions on Fuzzy Systems 1 (2) (1993) 98–110.

[19] W. Kowalczyk, F. Slisser, Analyzing customer retention with rough data models, Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery, Springer-Verlag, Trondheim, Norway, 1997, pp. 4–13.

[20] A. Kusiak, Rough set theory: a data mining tool for semiconductor manufacturing, IEEE Transactions on Electronics Packaging Manufacturing 24 (1) (2001) 44–50.

[21] S. Lee, G. Vachtsevanos, An application of rough set theory to defect detection of automotive glass, Mathematics and Computers in Simulation 60 (3–5) (2002) 225–231.

[22] P. Lingras, Unsupervised rough set classification using GAs, Journal of Intelligent Information Systems 16 (3) (2001) 215–228.

[23] P. Lingras, M. Hogo, M. Snorek, Interval set clustering of web users using modified kohonen self-organizing maps based on the properties of rough sets, Web Intelligence and Agent Systems 2 (3) (2004) 217–225.

[24] P. Lingras, C. West, Interval set clustering of web users with rough *K*-means, Journal of Intelligent Information Systems 23 (1) (2004) 5–16.

[25] P. Lingras, P.R. Yan, M. Hogo, Rough set based clustering: evolutionary, neural, and statistical approaches, Proceedings of the First Indian International Conference on Artificial Intelligence (2003) 1074–1087.

[26] R. Mathieu, J. Gibson, A Methodology for large scale R&D planning based on cluster analysis, IEEE Transactions on Engineering Management 40 (3) (2004) 283–292.

[27] L. Mazlack, A. He, Y. Zhu, S. Coppock, A rough set approach in choosing partitioning attributes, in: Proceedings of the ISCA 13th International Conference (CAINE-2000), 2000, pp. 1–6.

[28] S.H. Nguyen, H.S. Nguyen, Pattern extraction from data, Fundamentra Informaticae 34 (1–2) (1998) 1–16.

[29] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences 11 (5) (1982) 341–356.

[30] Z. Pawlak, Rough Sets – Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Boston, 1991.

[31] Z. Pawlak, Rough set approach to knowledge-based decision support, European Journal of Operational Research 99 (1) (1997) 48–57.

[32] Z. Pawlak, Rough classification, International Journal of Man–Machine Studies 20 (5) (1984) 469–483.

[33] H. Ralambondrainy, A conceptual version of the *K*-means algorithm, Pattern Recognition Letters 16 (11) (1995) 1147–1157.

[34] E. Ruspini, A new approach to clustering, Information Control 15 (1) (1969) 22–32.

[35] T.-L. Tseng, M.C. Jothishankar, T. Wu, Quality control problem in printed circuit manufacturing – an extended rough set approach, Journal of Manufacturing Systems 23 (1) (2004) 56–72.

[36] S. Tsumoto, Extraction of experts' decision process from clinical databases using rough set model, in: Proceedings of the First European Symposium, 1997, pp. 58–67.

[37] K. Voges, N. Pope, M. Brown, Cluster analysis of marketing data examining on-line shopping orientation: a comparison of *K*-means and rough clustering approaches, in: H. A Abbass, R. A Sarker, C.S. Newton (Eds.), Heuristics and Optimization for Knowledge Discovery, Idea Group Publishing, Hershey, PA, 2002, pp. 207–224.

[38] K. Voges, N. Pope, Generating compact rough cluster descriptions using an evolutionary algorithm, in: K. Deb (Ed.), GECCO2004: Genetic and Evolutionary Algorithm Conference – LNCS, Springer-Verlag, Berlin, 2004, pp. 1332–1333.

[39] K. Wong, D. Feng, S. Meikle, M. Fulham, Segmentation of dynamic pet images using cluster analysis, IEEE Transactions on Nuclear Science 49 (1) (2002) 200–207.

[40] S. Wu, A. Liew, H. Yan, M. Yang, Cluster analysis of gene expression data based on self-splitting and merging competitive learning, IEEE Transactions on Information Technology in BioMedicine 8 (1) (2004) 5–15.

[41] Y. Zhang, A. Fu, C. Cai, P. Heng, Clustering categorical data, in: Proceedings of the 16th International Conference on Data Engineering, 2000, pp. 305–324.

[42] W. Ziarko, Rough sets for intelligent image processing, in: Proceedings of the International Workshop on Rough Sets and Knowledge Discovery, 1993, pp. 399–410.

Darshit Parmar is a senior consultant in the Supply Chain Practice of IBM Global Services Group. He holds a MS degree in Industrial Engineering from Arizona State University and is currently working on his Ph.D. His research interests include Sense and Respond Supply Chain, Data Mining, Predictive Modeling and Optimization.

Teresa Wu (teresa.wu@asu.edu) is an associate professor in Industrial Engineering Department of Arizona State University. She has published papers in International Journal of Concurrent Engineering: Research and Application, International Journal of Product Research, ASME Transactions: Journal of Computing and Information Science in Engineering, the Journal of Operations Management. She received her Ph.D. from the University of Iowa. Her main areas of interests are in supply chain management, distributed decision support and information systems.

Jennifer Blackhurst is an Assistant Professor of Logistics and Supply Chain Management at Iowa State University. She received her Ph.D. in Industrial Engineering from the University of Iowa. Her current research interests include: supply chain risk and disruptions; supply chain coordination; and supplier assessment. Professor Blackhurst has articles published (or accepted) in such journals as *Production and Operations Management Journal, Decision Sciences Journal*, *Journal of Operations Management, International Journal of Production Research, Omega,* and *Supply Chain Management Review*. She serves on the Editorial Review Board for Decision Sciences and is a member of DSI and POMS.