A Link-Based Cluster Ensemble Approach for Categorical Data Clustering

Natthakan Iam-On, Tossapon Boongoen, Simon Garrett, and Chris Price

Abstract—Although attempts have been made to solve the problem of clustering categorical data via cluster ensembles, with the results being competitive to conventional algorithms, it is observed that these techniques unfortunately generate a final data partition based on incomplete information. The underlying ensemble-information matrix presents only cluster-data point relations, with many entries being left unknown. The paper presents an analysis that suggests this problem degrades the quality of the clustering result, and it presents a new link-based approach, which improves the conventional matrix by discovering unknown entries through similarity between clusters in an ensemble. In particular, an efficient link-based algorithm is proposed for the underlying similarity assessment. Afterward, to obtain the final clustering result, a graph partitioning technique is applied to a weighted bipartite graph that is formulated from the refined matrix. Experimental results on multiple real data sets suggest that the proposed link-based method almost always outperforms both conventional clustering algorithms for categorical data and well-known cluster ensemble techniques.

Index Terms—Clustering, categorical data, cluster ensembles, link-based similarity, data mining.

1 Introduction

Data clustering is one of the fundamental tools we have for understanding the structure of a data set. It plays a crucial, foundational role in machine learning, data mining, information retrieval, and pattern recognition. Clustering aims to categorize data into groups or clusters such that the data in the same cluster are more similar to each other than to those in different clusters. Many well-established clustering algorithms, such as k-means [1] and PAM [2], have been designed for numerical data, whose inherent properties can be naturally employed to measure a distance (e.g., euclidean) between feature vectors [3], [4]. However, these cannot be directly applied for clustering of categorical data, where domain values are discrete and have no ordering defined. An example of categorical attribute is $sex = \{male,\ female\}$ *or* $shape = \{circle,\ rectangle, \ldots\}$.

As a result, many categorical data clustering algorithms have been introduced in recent years, with applications to interesting domains such as protein interaction data [5]. The initial method was developed in [6] by making use of Gower's similarity coefficient [7]. Following that, the k-modes algorithm in [8] extended the conventional k-means with a simple matching dissimilarity measure and a frequency-based method to update centroids (i.e., clusters' representative). As a single-pass algorithm, Squeezer [9] makes use of a prespecified similarity threshold to determine which of the existing clusters (or a new cluster) to which a data point under examination is assigned. LIMBO [10] is a hierarchical clustering algorithm that uses the Information Bottleneck (IB) framework to define a distance measure for categorical tuples. The concepts of evolutionary computing and genetic algorithm have also been adopted by a partitioning method for categorical data, i.e., GAClust [11]. Cobweb [12] is a model-based method primarily exploited for categorical data sets. Different graph models have also been investigated by the STIRR [13], ROCK [14], and CLICK [15] techniques. In addition, several density-based algorithms have also been devised for such purpose, for instance, CACTUS [16], COOLCAT [17], and CLOPE [18].

Although, a large number of algorithms have been introduced for clustering categorical data, the No Free Lunch theorem [19] suggests[1] there is no single clustering algorithm that performs best for all data sets [20] and can discover all types of cluster shapes and structures presented in data [21]. Each algorithm has its own strengths and weaknesses. For a particular data set, different algorithms, or even the same algorithm with different parameters, usually provide distinct solutions. Therefore, it is difficult for users to decide which algorithm would be the *proper* alternative for a given set of data. Recently, cluster ensembles have emerged as an effective solution that is able to overcome these limitations, and improve the robustness as well as the quality of clustering results. The main objective of cluster ensembles is to combine different clustering decisions in such a way as to achieve accuracy superior to that of any individual clustering. Examples of well-known ensemble methods are:

- *N. Iam-On is with the School of Information Technology, Mae Fah Luang University, Muang, Chiang Rai, 57100, Thailand. E-mail: nt.iamon@gmail.com.*
- *T. Boongoen is with the Royal Thai Air Force Academy, 171/1 Klongthanhon, Saimai, Bangkok, 10220, Thailand. E-mail: t.boongoen@gmail.com.*
- *S. Garrett is with the Aispire Consulting Ltd., Tanyrallt, Aberystwyth, SY23 3PG, UK. E-mail: s.garrett@aispire.co.uk.*
- *C. Price is with the Department of Computer Science, Aberystwyth University, Llandinam Building, Aberystwyth, Ceredigion, SY23 3DB, UK. E-mail: cjp@aber.ac.uk.*

Manuscript received 4 Mar. 2010; revised 11 Aug. 2010; accepted 14 Aug. 2010; published online 21 Dec. 2010.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2010-03-0125.
Digital Object Identifier no. 10.1109/TKDE.2010.268.

[1] The No Free Lunch theorem seems to apply here because the problem of clustering can be reduced to an optimization problem—we are seeking to find the optimal set of clusters for a given data set via an algorithm.

1041-4347/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

1. the feature-based approach that transforms the problem of cluster ensembles to clustering categorical data (i.e., cluster labels) [11], [22], [23], [24],
2. the direct approach that finds the final partition through relabeling the base clustering results [25], [26],
3. graph-based algorithms that employ a graph partitioning methodology [27], [28], [29], and
4. the pairwise-similarity approach that makes use of co-occurrence relations between data points [30], [31], [32].

Despite notable success, these methods generate the final data partition based on incomplete information of a cluster ensemble. The underlying ensemble-information matrix presents only cluster-data point relationships while completely ignores those among clusters [33]. As a result, the performance of existing cluster ensemble techniques may consequently be degraded as many matrix entries are left *unknown*. This paper introduces a link-based approach to refining the aforementioned matrix, giving substantially less unknown entries. A link-based similarity measure [34], [35], [36] is exploited to estimate unknown values from a link network of clusters. This research uniquely bridges the gap between the task of data clustering and that of link analysis. It also enhances the capability of ensemble methodology for categorical data, which has not received much attention in the literature. In addition to the problem of clustering categorical data that is investigated herein, the proposed framework is generic such that it can also be effectively applied to other data types.

The rest of this paper is organized as follows: Section 2 presents the cluster ensemble framework upon which the current research has been established. The proposed link-based approach, including the underlying intuition of refining an ensemble-information matrix and details of a link-based similarity measure, is introduced in Section 3. Then, Section 4 exhibits the evaluation of this new approach against other cluster ensemble methods and categorical data clustering algorithms, over real data sets. Similarity and differences between the proposed method and many clustering algorithms for categorical data are discussed in Section 5. This underlines the novelty of the link-based ensemble framework and its unique application to categorical data clustering. The paper is concluded in Section 6 with suggestions for further work.

2 Cluster Ensemble Methodology

2.1 Problem Formulation and General Framework

Let $X = \{x_1, \ldots, x_N\}$ be a set of N data points and $\Pi = \{\pi_1, \ldots, \pi_M\}$ be a cluster ensemble with M base clusterings, each of which is referred to as an *ensemble member*. Each base clustering returns a set of clusters $\pi_i = \{C_1^i, C_2^i, \ldots, C_{k_i}^i\}$, such that $\bigcup_{j=1}^{k_i} C_j^i = X$, where k_i is the number of clusters in the ith clustering. For each $x \in X$, $C(x)$ denotes the cluster label to which the data point x belongs. In the ith clustering, $C(x) =$ "j" (or "C_j^i") if $x \in C_j^i$. The problem is to find a new partition π^* of a data set X that summarizes the information from the cluster ensemble Π. Fig. 1 shows the general framework of cluster ensembles. Essentially, solutions achieved from different base clusterings are aggregated to form a final partition. This metalevel methodology involves two major tasks of: 1) generating a cluster ensemble, and 2) producing the final partition, normally referred to as a *consensus function*.

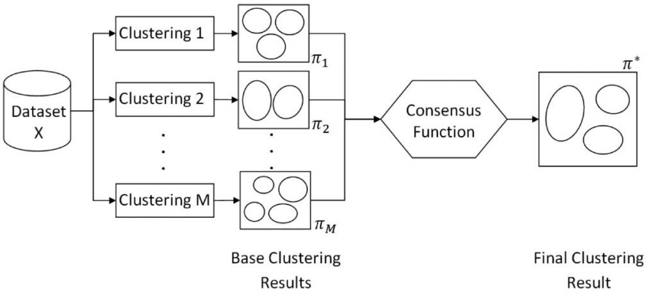
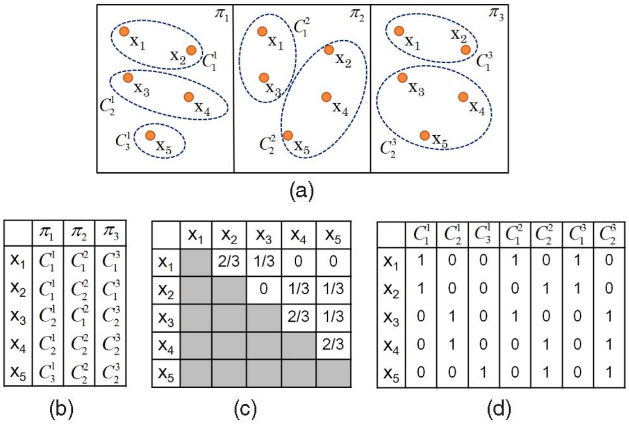


Fig. 1. The basic process of cluster ensembles. It first applies multiple base clusterings to a data set X to obtain diverse clustering decisions ($\pi_1 \ldots \pi_M$). Then, these solutions are combined to establish the final clustering result (π^*) using a consensus function.

2.2 Ensemble Generation Methods

It has been shown that ensembles are most effective when constructed from a set of predictors whose errors are dissimilar [37]. To a great extent, diversity among ensemble members is introduced to enhance the result of an ensemble [38]. Particularly for data clustering, the results obtained with any single algorithm over many iterations are usually very similar. In such a circumstance where all ensemble members agree on how a data set should be partitioned, aggregating the base clustering results will show no improvement over any of the constituent members. As a result, several heuristics have been proposed to introduce artificial instabilities in clustering algorithms, giving diversity within a cluster ensemble. The following ensemble generation methods yield different clusterings of the same data, by exploiting different cluster models and different data partitions.

- *Homogeneous ensembles*. Base clusterings are created using repeated runs of a single clustering algorithm, with several sets of parameter initializations, such as cluster centers of the k-means clustering technique [31], [22], [39].
- *Random-k*. One of the most successful techniques is randomly selecting the number of clusters (k) for each ensemble member [31], [38].
- *Data subspace/sampling*. A cluster ensemble can also be achieved by generating base clusterings from different subsets of initial data. It is intuitively assumed that each clustering algorithm will provide different levels of performance for different partitions of a data set [27]. Practically speaking, data partitions are obtained by projecting data onto different subspaces [40], [24], choosing different subsets of features [29], [41], or data sampling [42], [26], [43].
- *Heterogeneous ensembles*. A number of different clustering algorithms are used together to generate base clusterings [30], [44], [45].
- *Mixed heuristics*. In addition to using one of the aforementioned methods, any combination of them can be applied as well [27], [31], [33], [38], [32], [23], [29].



(a)

	π_1	π_2	π_3
x_1	C_1^1	C_1^2	C_1^3
x_2	C_1^1	C_2^2	C_1^3
x_3	C_2^1	C_1^2	C_2^3
x_4	C_2^1	C_2^2	C_2^3
x_5	C_3^1	C_2^2	C_2^3

(b)

	x_1	x_2	x_3	x_4	x_5
x_1		2/3	1/3	0	0
x_2			0	1/3	1/3
x_3				2/3	1/3
x_4					2/3
x_5					

(c)

	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2	C_1^3	C_2^3
x_1	1	0	0	1	0	1	0
x_2	1	0	0	0	1	1	0
x_3	0	1	0	1	0	0	1
x_4	0	1	0	0	1	0	1
x_5	0	0	1	0	1	0	1

(d)

Fig. 2. Examples of (a) cluster ensemble and the corresponding (b) label-assignment matrix, (c) pairwise-similarity matrix, and (d) binary cluster-association matrix, respectively. Note that $X = \{x_1, \ldots, x_5\}$, $\Pi = \{\pi_1, \pi_2, \pi_3\}$, $\pi_1 = \{C_1^1, C_2^1, C_3^1\}, \pi_2 = \{C_1^2, C_2^2\}$, and $\pi_3 = \{C_1^3, C_2^3\}$.

2.3 Consensus Functions

Having obtained the cluster ensemble, a variety of consensus functions have been developed and made available for deriving the ultimate data partition. Each consensus function utilizes a specific form of information matrix, which summarizes the base clustering results. From the cluster ensemble shown in Fig. 2a, three general types of such ensemble-information matrix can be constructed. First, the label-assignment matrix (Fig. 2b), of size $N \times M$, represents cluster labels that are assigned to each data point by different base clusterings. Second, the pairwise-similarity matrix (Fig. 2c), of size $N \times N$, summarizes co-occurrence statistics among data points. Furthermore, the binary cluster-association matrix (BM) (Fig. 2d) provides a cluster-specific view of the original label-assignment matrix. The association degree that a data point belonging to a specific cluster is either 1 or 0. In light of this background, consensus methods can be categorized as follows:

- *Feature-based approach*. It transforms the problem of cluster ensembles to clustering categorical data. Specifically, each base clustering provides a cluster label as a new feature describing each data point (see Fig. 2b), which is utilized to formulate the ultimate solution [11], [23], [39], [24].
- *Direct approach*. It is based on relabeling π_i and searching for the π^* that has the best match with all $\pi_i, i = 1 \ldots M$ [25], [26], [22]. Conceptually, the underlying relabel process allows the homogeneous labels to be established from heterogeneous clustering decisions, where each base clustering possesses a unique set of decision labels (see Fig. 2b).
- *Pairwise-similarity approach*. It creates a matrix, containing the pairwise similarity among data points (see Fig. 2c for an example), to which any similarity-based clustering algorithm (e.g., hierarchical clustering) can be applied [30], [31], [32].
- *Graph-based approach*. It makes use of the graph representation to solve the cluster ensemble problem [27], [28], [29]. Specifically to the consensus methods in [27] and [29], a graph representing the similarity among data points is created from a pairwise matrix similar to that given in Fig. 2c. To achieve the final clustering result, this graph is partitioned into a definite number of approximately equal-sized partitions, using METIS [46]. In addition, the binary cluster-association matrix shown in Fig. 2d is used for generating a bipartite graph whose vertices represent both data points and clusters. According to [28], the solution to a cluster ensemble problem is to divide this graph using either METIS or Spectral graph partitioning (SPEC) [47].

2.4 Cluster Ensembles of Categorical Data

While a large number of cluster ensemble techniques for numerical data have been put forward in the previous decade, there are only a few studies that apply such a methodology to categorical data clustering. The method introduced in [48] creates an ensemble by applying a conventional clustering algorithm (e.g., k-modes [8] and COOLCAT [17]) to different data partitions, each of which is constituted by a unique subset of data attributes. Once an ensemble has been obtained, the graph-based consensus functions of [28] and [29] are utilized to generate the final clustering result.

Unlike the conventional approach, the technique developed in [49] acquires a cluster ensemble without actually implementing any base clustering on the examined data set. In fact, each attribute is considered as a base clustering that provides a unique data partition. In particular, a cluster in such attribute-specific partition contains data points that share a specific attribute value (i.e., categorical label). Thus, the ensemble size is determined by the number of categorical labels, across all data attributes. The final clustering result is generated using the graph-based consensus techniques presented in [29]. Specific to this so-called "direct" ensemble generation method, a given categorical data set can be represented using a binary cluster-association matrix, whose example is shown earlier in Fig. 2d. Such an information matrix is analogous to the "market-basket" numerical representation of categorical data, which has been the focus of traditional categorical data analysis [50], [51].

3 A Novel Link-Based Approach

Existing cluster ensemble methods to categorical data analysis rely on the typical pairwise-similarity and binary cluster-association matrices [48], [49], which summarize the underlying ensemble information at a rather coarse level. Many matrix entries are left "unknown" and simply recorded as "0." Regardless of a consensus function, the quality of the final clustering result may be degraded. As a result, a link-based method has been established with the ability to discover unknown values and, hence, improve the accuracy of the ultimate data partition [33]. In spite of promising findings, this initial framework is based on the data point-data point pairwise-similarity matrix, which is highly expensive to obtain. The link-based similarity technique, SimRank [52], that is employed to estimate the similarity among data points is inapplicable to a large data set.

To overcome these problems, a new link-based cluster ensemble (LCE) approach is introduced herein. It is more efficient than the former model, where a BM-like matrix is

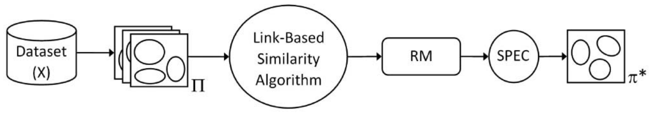


Fig. 3. The link-based cluster ensemble framework: 1) a cluster ensemble $\Pi = \{\pi_1, \ldots, \pi_M\}$ is created from M base clusterings, 2) a refined cluster-association matrix is then generated from the ensemble using a link-based similarity algorithm, and 3) a final clustering result (π^*) is produced by a consensus function of the spectral graph partitioning.

used to represent the ensemble information. The focus has shifted from revealing the similarity among data points to estimating those between clusters. A new link-based algorithm has been specifically proposed to generate such measures in an accurate, inexpensive manner. The LCE methodology is illustrated in Fig. 3. It includes three major steps of: 1) creating base clusterings to form a cluster ensemble (Π), 2) generating a refined cluster-association matrix (RM) using a link-based similarity algorithm, and 3) producing the final data partition (π^*) by exploiting the spectral graph partitioning technique as a consensus function.

3.1 Creating a Cluster Ensemble

Type I (Direct ensemble). Following the study in [49], the first type of cluster ensemble transforms the problem of categorical data clustering to cluster ensembles by considering each categorical attribute value (or label) as a cluster in an ensemble. Let $X = \{x_1, \ldots, x_N\}$ be a set of N data points, $A = \{a_1, \ldots, a_M\}$ be a set of categorical attributes, and $\Pi = \{\pi_1, \ldots, \pi_M\}$ be a set of M partitions. Each partition π_i is generated for a specific categorical attribute $a_i \in A$. Clusters belonging to a partition $\pi_i = \{C_1^i, \ldots, C_{k_i}^i\}$ correspond to different values of the attribute $a_i = \{a_1^i, \ldots, a_{k_i}^i\}$, where $\bigcup_{j=1}^{k_i} C_j^i = a_i$ and k_i is the number of values of attribute a_i. With this formalism, categorical data X can be directly transformed to a cluster ensemble Π, without actually implementing any base clustering. While single-attribute data partitions may not be as accurate as those obtained from the clustering of all data attributes, they can bring about great diversity within an ensemble. Besides its efficiency, this ensemble generation method has the potential to lead to a high-quality clustering result.

Type II (Full-space ensemble). Unlike the previous case, the following two ensemble types are created from base clustering results, each of which is obtained by applying a clustering algorithm to the categorical data set. For this study, the k-modes technique [8] is used to generate base clusterings, each with a random initialization of cluster centers. In particular to a full-space ensemble, base clusterings are created from the original data, i.e., with all data attributes. To introduce an artificial instability to k-modes, the following two schemes are employed to select the number of clusters in each base clusterings: 1) *Fixed-k*, $k = \lceil\sqrt{N}\rceil$ (where N is the number of data points), and 2) *Random-k*, $k \in \{2, \ldots, \lceil\sqrt{N}\rceil\}$.

Type III (Subspace ensemble). Another alternative to generate diversity within an ensemble is to exploit a number of different data subsets. To this extent, the cluster ensemble is established on various data subspaces, from which base clustering results are generated [48]. Similar to the study in [41], for a given $N \times d$ data set of N data points and d attributes, an $N \times q$ data subspace (where $q < d$) is generated by

$$q = q_{min} + \lfloor \alpha(q_{max} - q_{min}) \rfloor, \tag{1}$$

where $\alpha \in [0, 1]$ is a uniform random variable, q_{min} and q_{max} are the lower and upper bounds of the generated subspace, respectively. In particular, q_{min} and q_{max} are set to $0.75d$ and $0.85d$. An attribute is selected one by one from the pool of d attributes, until the collection of q is obtained. The index of each randomly selected attribute is determined as $h = \lfloor 1 + \beta d \rfloor$, given that h denotes the hth attribute in the pool of d attributes and $\beta \in [0, 1)$ is a uniform random variable. Note that k-modes is exploited to create a cluster ensemble from the set of subspace attributes, using both *Fixed-k* and *Random-k* schemes for selecting the number of clusters.

3.2 Generating a Refined Matrix

Several cluster ensemble methods, both for numerical [28], [29] and categorical data [48], [49], are based on the binary cluster-association matrix. Each entry in this matrix $\mathrm{BM}(x_i, cl) \in \{0, 1\}$ represents a *crisp* association degree between data point $x_i \in X$ and cluster $cl \in \Pi$. According to Fig. 2 that shows an example of cluster ensemble and the corresponding BM, a large number of entries in the BM are *unknown*, each presented with "0." Such condition occurs when relations between different clusters of a base clustering are originally assumed to be nil. In fact, each data point can possibly associate (to a certain degree within $[0, 1]$) to several clusters of any particular clustering. These hidden or unknown associations can be estimated from the similarity among clusters, discovered from a network of clusters.

Based on this insight, the refined cluster-association matrix is put forward as the enhanced variation of the original BM. Its aim is to approximate the value of unknown associations ("0") from known ones ("1"), whose association degrees are preserved within the RM, i.e., $BM(x_i, cl) = 1 \rightarrow RM(x_i, cl) = 1$. For each clustering $\pi_t, t = 1 \ldots M$ and their corresponding clusters $C_1^t, \ldots, C_{k_t}^t$ (where k_t is the number of clusters in the clustering π_t), the association degree $RM(x_i, cl) \in [0, 1]$ that data point $x_i \in X$ has with each cluster $cl \in \{C_1^t, \ldots, C_{k_t}^t\}$ is estimated as follows:

$$RM(x_i, cl) = \begin{cases} 1, & if cl = C_*^t(x_i), \\ sim(cl, C_*^t(x_i)), & otherwise, \end{cases} \tag{2}$$

where $C_*^t(x_i)$ is a cluster label (corresponding to a particular cluster of the clustering π_t) to which data point x_i belongs. In addition, $sim(C_x, C_y) \in [0, 1]$ denotes the similarity between any two clusters C_x, C_y, which can be discovered using the following link-based algorithm. Note that, for any clustering $\pi_t \in \Pi$, $1 \leq \sum_{\forall C \in \pi_t} RM(x_i, C) \leq k_t$. Unlike the measure of fuzzy membership, the typical constraint of $\sum_{\forall C \in \pi_t} RM(x_i, C) = 1$ is not appropriate for rescaling associations within the RM. In fact, this local normalization will significantly distort the true semantics of known associations ("1"), such that their magnitudes become dissimilar, different from one clustering to another. According to the empirical investigation, this fuzzy-like enforcement decreases the quality of the RM, and hence, the performance of the resulting cluster ensemble method.

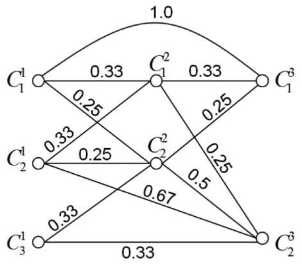


Fig. 4. An example of a cluster network, where each edge is marked with its weight.

3.2.1 Weighted Triple-Quality (WTQ): A New Link-Based Similarity Algorithm

Given a cluster ensemble Π of a set of data points X, a weighted graph $G = (V, W)$ can be constructed, where V is the set of vertices each representing a cluster and W is a set of weighted edges between clusters. Formally, the weight assigned to the edge $w_{xy} \in W$, that connects clusters $C_x, C_y \in V$, is estimated by the proportion of their overlapping members.

$$w_{xy} = \frac{|L_x \cap L_y|}{|L_x \cup L_y|}, \tag{3}$$

where $L_z \subset X$ denotes the set of data points belonging to cluster $C_z \in V$. Fig. 4 shows the network of clusters that is generated from the example given in Fig. 2. Note that circle nodes represent clusters and edges exist only when the corresponding weights are nonzero.

Shared neighbors have been widely recognized as the basic evidence to justify the similarity among vertices in a link network [35], [36]. Formally, a vertex $C_k \in V$ is a common neighbor (sometimes called "triple," which is short for "center of the connected triple") of vertices $C_x, C_y \in V$, provided that $w_{xk}, w_{yk} \in W$. Many advanced methods extend this basis by taking into account common neighbors that may be many edges away from the two under examination: for instance, Connected-Path [34], SimRank [52], and a variation of random walk algorithms [53], [54]. Despite reported effectiveness, these techniques are computationally expensive, or even impractical for a large data set. Henceforth, the Weighted Triple-Quality algorithm is proposed, as part of the current research, for the efficient approximation of the similarity between clusters in a link network. Unlike the technique in [55] that simply counts the number of triples, WTQ aims to differentiate the significance of triples and hence their contributions toward the underlying similarity measure. WTQ is inspired by the initial measure in [56], which evaluates the association between home pages. In particular, features of the compared pages p_a and p_b are used to estimate their similarity $s(p_a, p_b)$ as follows:

$$s(p_a, p_b) = \sum_{\forall z_c \in Z} \frac{1}{\log(frequency(z_c))}, \tag{4}$$

where Z denotes the set of features shared by home pages p_a and p_b, and $frequency(z_d)$ represents the number of times z_d appearing in the studied set of pages. Note that the method gives high weights to rare features and low weights to features that are common to most of the pages. For WTQ, (4) can be modified to discriminate the quality of shared triples between a pair of clusters in question. The quality of each cluster is determined by the rarity of links connecting to other clusters in a network. With a weighted graph G presented in Fig. 4, the WTQ measure of clusters $C_x, C_y \in V$ with respect to each triple $C_k \in V$ is estimated by

$$WTQ_{xy}^k = \frac{1}{W_k}. \tag{5}$$

Here, W_k is defined as $W_k = \sum_{\forall t \in N_k} w_{tk}$, where $N_k \subset V$ denotes the set of clusters that is directly linked to the cluster C_k, such that $\forall C_t \in N_k, w_{tk} \in W$. The accumulative WTQ score from all triples $(1 \ldots q)$ between clusters C_x and C_y can be found as follows:

$$WTQ_{xy} = \sum_{k=1}^{q} WTQ_{xy}^k. \tag{6}$$

The WTQ algorithm is summarized below:

```
ALGORITHM: WTQ$(G, C_x, C_y)$
$G = (V, W)$, a weighted graph, where $C_x, C_y \in V$;
$N_k \subset V$, a set of adjacent neighbors of $C_k \in V$;
$W_k = \sum_{\forall C_t \in N_k} w_{tk}$;
$WTQ_{xy}$, the WTQ measure of $C_x$ {and} $C_y$;
$WTQ_{xy} \leftarrow 0$
For each $c \in N_x$
    If $c \in N_y$
        $WTQ_{xy} \leftarrow WTQ_{xy} + \frac{1}{W_c}$
Return $WTQ_{xy}$
```

Following that, the similarity between clusters C_x and C_y can be estimated by

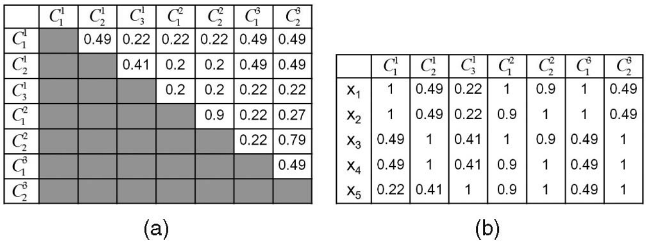
$$sim(C_x, C_y) = \frac{WTQ_{xy}}{WTQ_{max}} \times DC, \tag{7}$$

where WTQ_{max} is the maximum WTQ_{pq} value of any two clusters $C_p, C_q \in V$ and $DC \in [0, 1]$ is a constant decay factor (i.e., confidence level of accepting two nonidentical clusters as being similar). With this link-based similarity metric, $sim(C_x, C_y) \in [0, 1]$ with $sim(C_x, C_x) = 1$, $C_x, C_y \in V$. It is also reflexive such that $sim(C_x, C_y)$ is equivalent to $sim(C_y, C_x)$. Following the example shown in Figs. 2 and 4, the WTQ similarity among different clusters and the resulting RM are presented in Figs. 5a and 5b, respectively.

3.3 Applying a Consensus Function to RM

Having obtained an RM, a graph-based partitioning method is exploited to obtain the final clustering. This consensus function requires the underlying matrix to be initially transformed into a weighted bipartite graph. Given an RM representing associations between N data points and P clusters in an ensemble Π, a weighted graph $G = (V, W)$ can be constructed, where $V = V^X \cup V^C$ is a set of vertices representing both data points V^X and clusters V^C, and W denotes a set of weighted edges that can be defined as follows:

- $w_{ij} \notin W$ when vertices $v_i, v_j \in V^X$.
- $w_{ij} \notin W$ when vertices $v_i, v_j \in V^C$.



	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2	C_1^3	C_2^3
C_1^1	▨	0.49	0.22	0.22	0.22	0.49	0.49
C_2^1	▨	▨	0.41	0.2	0.2	0.49	0.49
C_3^1	▨	▨	▨	0.2	0.2	0.22	0.22
C_1^2	▨	▨	▨	▨	0.9	0.22	0.27
C_2^2	▨	▨	▨	▨	▨	0.22	0.79
C_1^3	▨	▨	▨	▨	▨	▨	0.49
C_2^3	▨	▨	▨	▨	▨	▨	▨

(a)

	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2	C_1^3	C_2^3
x_1	1	0.49	0.22	1	0.9	1	0.49
x_2	1	0.49	0.22	0.9	1	1	0.49
x_3	0.49	1	0.41	1	0.9	0.49	1
x_4	0.49	1	0.41	0.9	1	0.49	1
x_5	0.22	0.41	1	0.9	1	0.49	1

(b)

Fig. 5. The illustrations of (a) WTQ similarity degrees between different clusters and (b) the resulting RM, where $DC = 0.9$.

- Otherwise, $w_{ij} = RM(v_i, v_j)$ when vertices $v_i \in V^X$ and $v_j \in V^C$. Note that the graph G is bidirectional such that w_{ij} is equivalent to w_{ji}.

Given such a graph, a spectral graph partitioning method similar to that in [47] is applied to generate a final data partition. This is a powerful method for decomposing an undirected graph, with good performance being exhibited in many application areas, including protein modeling, information retrieval, and identification of densely connected online hypertextual regions [57]. Principally, given a graph $G = (V, W)$, SPEC first finds the K largest eigenvectors $u_1, \ldots, u_K$ of W, which are used to formed another matrix U (i.e., $U = [u_1, \ldots, u_K]$), whose rows are then normalized to have unit length. By considering the row of U as K-dimensional embedding of the graph vertices, SPEC applies k-means to these embedded points in order to acquire the final clustering result.

4 Performance Evaluation

This section presents the evaluation of the proposed link-based method (LCE), using a variety of validity indices and real data sets. The quality of data partitions generated by this technique is assessed against those created by different categorical data clustering algorithms and cluster ensemble techniques.

4.1 Investigated Data Sets

The experimental evaluation is conducted over nine data sets. The "20Newsgroup" data set is a subset of the well-known text data collection—*20-Newsgroups*,[2] while the others are obtained from the UCI Machine Learning Repository [58]. Their details are summarized in Table 1. Missing values (denoted as "?") in these data sets are simply treated as a new categorical value. The "20Newsgroup" data set contains 1,000 documents from two newsgroups, each of which is described by the occurrences of 6,084 different terms. In particular, the frequency ($f \in \{0, 1, \ldots, \infty\}$) that a key word appears in each document is transformed into a nominal value: "Yes" if $f > 0$, "No" otherwise. Moreover, the "KDDCup99" data set used in this evaluation is a randomly selected subset of the original data. Each data point (or record) corresponds to a network connection and contains 42 attributes: some are nominal and the rest are continuous. Following the study in [17], numerical attributes are transformed to categorical using a simple discretization process. For each attribute, any value less than the median is assigned a label "0," otherwise "1." Note that the selected set of data records covers 20 different connection classes. These two data sets are specifically included to assess the performance of different clustering methods, with respect to the large numbers of dimensionality and data points, respectively.

4.2 Experiment Design

The experiments set out to investigate the performance of LCE compared to a number of clustering algorithms, both specifically developed for categorical data analysis and those state-of-the-art cluster ensemble techniques found in literature. Baseline model is also included in the assessment, which simply applies SPEC, as a consensus function, to the conventional BM (see Section 4.2.2). For comparison, as in [28], [31], [22], each clustering method divides data points into a partition of K (the number of *true classes* for each data set) clusters, which is then evaluated against the corresponding true partition using the following set of label-based evaluation indices: Classification Accuracy (CA) [23], Normalized Mutual Information (NMI) [29] and Adjusted Rand (AR) Index [59]. Further details of these quality measures are provided in Section I of the online supplementary.[3] Note that, true classes are known for all data sets but are explicitly not used by the cluster ensemble process. They are only used to evaluate the quality of the clustering results.

4.2.1 Parameter Settings

In order to evaluate the quality of cluster ensemble methods previously identified, they are empirically compared, using the settings of cluster ensembles exhibited below.

- Five types of cluster ensembles are investigated in this evaluation: Type-I, Type-II (Fixed-k), Type-II (Random-k), Type-III (Fixed-k), and Type-III (Random-k). The k-modes clustering algorithm is specifically used to generate the base clusterings.
- Ensemble size (M) of 10 is experimented.
- The quality of each method with respect to a specific ensemble setting is generalized as the average of 50 runs.
- The constant decay factor (DC) of 0.9 is exploited with WTQ.

4.2.2 Compared Methods

To fully evaluate the potential of the proposed method, it is compared to the baseline model (referred to as "Base" hereafter), which applies SPEC to the BM. This allows the quality of BM and RM to be directly compared. In addition, five clustering techniques for categorical data and five methods developed for cluster ensemble problems are included in this evaluation. Details of these techniques are given below.

Clustering algorithms for categorical data. Based on their notable performance reported in the literature and availability, five different algorithms are selected to demonstrate

[2] http://people.csail.mit.edu/jrennie/20Newsgroups/.

[3] Available at http://itschool.mfu.ac.th/~natthakan/tkde2012/.

TABLE 1
Description of Data Sets: Number of Data Points (N), Attributes (d), Attribute Values ($\mathbb{A}$), and Classes (K)

Dataset	N	d	$\mathbb{A}$	K
Zoo	101	16	36	7
Lymphography	148	18	59	4
Soybean	307	35	132	19
Primary Tumor	339	17	42	22
Congressional Votes	435	16	48	2
Breast Cancer	683	9	89	2
Mushroom	8,124	22	117	2
20Newsgroup	1,000	6,084	12,168	2
KDDCup99	100,000	42	139	20

the efficiency of conventional techniques to clustering categorical data: Squeezer, GAClust, k-modes, CLOPE, and Cobweb. Squeezer [9] is a single-pass algorithm that considers one data point at a time. Each data point is either placed in one of the existing clusters if their distance is less than a given threshold, or used to form a new cluster. GAClust [11] searches for a data partition (referred to as the "median" partition), which has the minimum dissimilarity to those partitions generated by categorical attributes. Note that the similarity (or closeness) between two partitions is estimated by using a generalization of the classical conditional entropy. A genetic algorithm has been employed to make the underlying search process more efficient, with the partitions being represented by chromosomes.

k-modes [8] extends the conventional k-means technique, with a simple matching dissimilarity measure. The distance is estimated by the number of common categorical attributes shared by two data points. It iteratively refines k cluster representatives, each as the attribute vector that has the minimal distance to all the points in a cluster (i.e., the cluster's most frequent attribute values). CLOPE [18] is a fast and scalable clustering technique, initially designed for transactional data analysis. Its underlying concept is to increase the height-to-width ratio of the cluster histogram. This is achieved through a repulsion parameter that controls tightness of transactions in a cluster, and hence the resulting number of clusters. Cobweb [12] is a conceptual clustering method. It creates a classification tree, in which each node corresponds to a concept. Observations are incrementally integrated into the classification tree, along the path of best matching nodes. This is guided by the heuristic evaluation measure, called category utility. A given utility threshold determines the sibling nodes that are used to form the resulting data partition.

In this experiment, a similarity threshold, a repulsion value, and a category utility threshold, which are required (as an input) by Squeezer, CLOPE, and Cobweb, respectively, are set particularly for each data set such that a desired number of clusters is obtained. As for the GAClust algorithm, the population size is set to be 50, the seed parameter $\in \{1, \dots, 10\}$ and other parameters are left to their default values.

Cluster ensemble methods. LCE is also assessed against five ensemble methods of CO+SL, CO+AL, Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper-Graph Partitioning Algorithm (HGPA), and Metaclustering Algorithm (MCLA). The first two algorithms are based principally on the pairwise similarity among data points. Given a cluster ensemble $\Pi = \{\pi_1, \dots, \pi_M\}$ of a data set $X = \{x_1, \dots, x_N\}$, an $N \times N$ similarity matrix (CO) is constructed by $CO(x_i, x_j) = \frac{1}{M} \sum_{m=1}^{M} S_m(x_i, x_j)$, where $CO(x_i, x_j) \in [0, 1]$ represents the similarity measure between data points $x_i, x_j \in X$. In addition, $S_m(x_i, x_j) = 1$ if $C^m(x_i) = C^m(x_j)$, and $S_m(x_i, x_j) = 0$ otherwise. Note that $C^m(x_i)$ denotes the cluster label of the mth clustering to which a data point $x_i \in X$ belongs. Since CO is a similarity matrix, any similarity-based clustering algorithm can be applied to this matrix to yield the final partition π^*. Specifically to [31], the single-linkage (SL) and average-linkage (AL) agglomerative hierarchical clusterings are used for such purpose.

To consolidate the underlying evaluation, three well-known graph-based cluster ensemble algorithms are also examined: CSPA, HGPA, and MCLA [29]. First, the Cluster-based Similarity Partitioning Algorithm creates a similarity graph, where vertices represent data points and edges' weight represent similarity scores obtained from the CO matrix. Afterward, a graph partitioning algorithm called METIS [46] is used to partition the similarity graph into K clusters. The Hypergraph Partitioning Algorithm constructs a hypergraph, where vertices represent data points and the same-weighted hyperedges represent clusters in the ensemble. Then, HMETIS [60] is applied to partition the underlying hypergraph into K parts with roughly the same size. Unlike the previous methods, the Metaclustering Algorithm generated a graph that represents the relationships among clusters in the ensemble. In this metalevel graph, each vertex corresponds to each cluster in the ensemble and each edge's weight between any two cluster vertices is computed using the binary Jaccard measure. METIS is also employed to partition the metalevel graph into K metaclusters. Each data point has a specific association degree to each metacluster. This can be estimated from the number of original clusters to which the data point belongs, in the underlying metacluster. The final clustering is produced by assigning each data point to the metacluster with which it is most frequently associated.

4.3 Experiment Results

Based on the classification accuracy, Table 2 compares the performance of different clustering techniques over examined data sets. Note that the presented measures of cluster ensemble methods that implement the ensemble Type-II and Type-III are the averages across 50 runs. In addition, a measure is marked "N/A" when the clustering result is not obtainable. For each data set, the highest five CA-based values are highlighted in **boldface**.

The results shown in this table indicate that the LCE methods usually perform better than the investigated collection of cluster ensemble techniques and clustering algorithms for categorical data. In particular to Type-II and Type-III ensembles, LCE also enhances the performance of k-modes, which is used as base clusterings. According to the findings with the 20Newsgroup data set, LCE is effective for such high-dimensional data, where Squeezer and Cobweb fail to generate the clustering results. Likewise, LCE is also applicable to a large data set such as

TABLE 2
Classification Accuracy of Different Clustering Methods

Dataset	Ensemble Type	LCE	Base	CO+SL	CO+AL	CSPA	HGPA	MCLA	Squeezer	GAClust	*k*-modes	CLOPE	Cobweb
Zoo	I	0.891	0.802	0.881	0.891	0.802	0.594	0.465	0.871	0.843	0.851	0.832	0.911
	II-Fixed-k	**0.921**	0.890	0.883	0.899	0.826	0.838	0.854					
	II-Random-k	**0.931**	0.877	0.861	0.875	0.824	0.826	0.836					
	III-Fixed-k	**0.941**	0.889	0.891	**0.916**	0.827	0.832	0.858					
	III-Random-k	**0.931**	0.880	0.866	0.876	0.828	0.842	0.837					
Lymphography	I	0.743	0.743	0.574	0.581	0.736	0.662	0.568	0.568	0.737	0.679	0.574	0.662
	II-Fixed-k	**0.797**	**0.744**	0.558	0.726	**0.744**	0.709	0.738					
	II-Random-k	**0.784**	0.715	0.557	0.671	0.693	0.633	0.707					
	III-Fixed-k	**0.784**	0.728	0.564	0.707	0.728	0.696	0.726					
	III-Random-k	**0.777**	0.707	0.557	0.668	0.678	0.637	0.689					
Soybean	I	**0.746**	0.668	0.440	0.619	0.651	0.472	0.130	0.658	0.413	0.618	**0.746**	0.531
	II-Fixed-k	**0.756**	0.645	0.512	0.640	0.589	0.617	0.618					
	II-Random-k	0.681	0.643	0.484	0.576	0.552	0.555	0.488					
	III-Fixed-k	**0.713**	0.642	0.467	0.624	0.571	0.603	0.608					
	III-Random-k	**0.700**	0.613	0.454	0.562	0.539	0.555	0.436					
Primary Tumor	I	0.445	0.416	0.277	0.381	0.445	0.292	0.248	**0.466**	0.373	0.436	0.336	0.369
	II-Fixed-k	**0.478**	0.455	0.308	0.447	0.450	0.436	0.439					
	II-Random-k	**0.496**	0.453	0.298	0.428	0.445	0.424	0.352					
	III-Fixed-k	**0.463**	0.453	0.294	0.440	0.450	0.424	0.432					
	III-Random-k	**0.484**	0.434	0.290	0.414	0.430	0.407	0.378					
Vote	I	0.876	**0.878**	0.616	0.616	0.864	0.614	**0.883**	0.618	0.835	0.861	0.614	**0.878**
	II-Fixed-k	**0.880**	0.868	0.616	0.845	0.855	0.874	0.854					
	II-Random-k	**0.897**	0.871	0.615	0.859	0.854	0.866	0.856					
	III-Fixed-k	**0.878**	0.874	0.615	0.849	0.854	0.873	0.857					
	III-Random-k	**0.883**	0.872	0.615	0.867	0.857	0.859	0.858					
Breast Cancer	I	0.940	0.944	0.652	0.653	0.851	0.676	0.890	0.862	0.828	0.869	0.911	0.950
	II-Fixed-k	**0.971**	0.663	0.651	**0.954**	0.832	0.870	0.889					
	II-Random-k	**0.972**	0.847	0.652	0.941	0.812	0.840	0.910					
	III-Fixed-k	**0.969**	0.666	0.652	0.953	0.837	0.850	0.905					
	III-Random-k	**0.966**	0.791	0.652	0.936	0.812	0.844	0.921					
Mushroom	I	**0.894**	**0.883**	0.522	0.523	N/A	0.518	0.669	0.536	0.603	0.715	0.518	**0.894**
	II-Fixed-k	0.802	0.555	0.535	0.534	N/A	0.716	0.689					
	II-Random-k	**0.889**	0.637	0.518	0.678	N/A	0.540	0.707					
	III-Fixed-k	**0.883**	0.539	0.536	0.524	N/A	0.742	0.732					
	III-Random-k	**0.889**	0.605	0.536	0.694	N/A	0.539	0.698					
20Newsgroup	I	0.600	0.600	0.600	0.600	0.600	**0.941**	0.600	N/A	0.600	0.600	0.600	N/A
	II-Fixed-k	**0.783**	0.600	0.600	0.600	0.600	0.600	0.600					
	II-Random-k	**0.723**	0.600	0.600	0.600	0.600	0.600	0.600					
	III-Fixed-k	**0.728**	0.600	0.600	0.600	0.600	0.600	0.600					
	III-Random-k	**0.740**	0.600	0.600	0.600	0.616	0.600	0.600					
KDDCup99	I	**0.984**	0.971	N/A	N/A	N/A	0.610	0.562	0.976	0.562	0.956	0.817	0.965
	II-Fixed-k	0.977	0.962	N/A	N/A	N/A	0.864	0.957					
	II-Random-k	**0.987**	0.979	N/A	N/A	N/A	0.562	0.562					
	III-Fixed-k	**0.982**	**0.981**	N/A	N/A	N/A	0.819	0.965					
	III-Random-k	**0.986**	0.967	N/A	N/A	N/A	0.804	0.562					

The five highest CA scores of each data set are highlighted in ***boldface****. Note that unobtainable results are marked as "N/A."*

KDDCup99, for which several cluster ensemble techniques (CO+SL, CO+AL, and CSPA) are immaterial.

With the measures of LCE models being mostly higher than those of the corresponding baseline counterparts (Base), the quality of the RM appears to be significantly better than that of the original, binary variation. As compared to the LCE models that use Type-II and Type-III ensembles (both "Fixed-k" and "Random-k"), the LCE with Type-I (or direct) ensemble is less effective. This is greatly due to the quality of base clusterings, which are single attribute and multiattribute for Type-I and the others, respectively. Despite its inefficiency, CSPA has the best performance among assessed ensemble methods. In addition, Cobweb is the most effective among five categorical data clustering algorithms included in this evaluation. Similar experimental results are also observed using NMI and AR evaluation indices. The corresponding details are given in Section II-A of the online supplementary.

In order to further evaluate the quality of identified techniques, the number of times that one method is significantly *better* and *worse* (of 95 percent confidence level) than the others are assessed across experimented data sets. Let $\overline{X}_C(i,\beta)$ be the average value of validity index $C \in \{CA, NMI, AR\}$ across n runs ($n = 50$ in this evaluation) for a clustering method $i \in CM$ (CM is a set of 40 experimented clustering methods), on a specific data set $\beta \in DT$ (DT is a set of six data sets). To obtain a fair comparison, this pairwise assessment is conducted on the results with six data sets, where the clustering results can be obtained for all the clustering methods. Also note that CM consists of five clustering algorithms for categorical data and 35 different cluster ensemble models, each of which is a unique combination of ensemble type (i.e., Type-I, Type-II(Fixed-k), Type-II(Random-k), Type-III(Fixed-k), and Type-III(Random-k)) and ensemble method (i.e., LCE, Base, CO+SL, CO+AL, CSPA, HGPA, and MCLA).

The 95 percent confidence interval, $[L_{\overline{X}_C(i,\beta)}, U_{\overline{X}_C(i,\beta)}]$, for the mean $\overline{X}_C(i,\beta)$ of each validity criterion C is calculated by

$$L_{\overline{X}_C(i,\beta)} = \overline{X}_C(i,\beta) - 1.96\frac{S_C(i,\beta)}{\sqrt{n}}$$
$$\text{and } U_{\overline{X}_C(i,\beta)} = \overline{X}_C(i,\beta) + 1.96\frac{S_C(i,\beta)}{\sqrt{n}}$$

Note that $S_C(i,\beta)$ denotes the standard deviation of the validity index C across n runs for a clustering method i and a data set β. The number of times that one method $i \in CM$ is significantly *better* than its competitors, $B_C(i)$ (in accordance with the validity criterion C), can be defined as

$$B_C(i) = \sum_{\forall \beta \in DT} \sum_{\forall i^* \in CM, i^* \neq i} better_C^{\beta}(i,i^*), \tag{8}$$

$$better_C^{\beta}(i,i^*) = \begin{cases} 1, & if L_{\overline{X}_C(i,\beta)} > U_{\overline{X}_C(i^*,\beta)}, \\ 0, & otherwise. \end{cases} \tag{9}$$

Similarly, the number of times that one method $i \in CM$ is significantly *worse* than its competitors, $W_C(i)$, in accordance with the validity criterion C, can be computed as

$$W_C(i) = \sum_{\forall \beta \in DT} \sum_{\forall i^* \in CM, i^* \neq i} worse_C^{\beta}(i,i^*), \tag{10}$$

$$worse_C^{\beta}(i,i^*) = \begin{cases} 1, & if U_{\overline{X}_C(i,\beta)} < L_{\overline{X}_C(i^*,\beta)}, \\ 0, & otherwise. \end{cases} \tag{11}$$

Using the aforementioned assessment formalism, Table 3 illustrates for each method the frequencies of significant better (B) and significant worse (W) performance, which are categorized in accordance with the evaluation indices (CA, NMI, and AR). The results shown in this table indicate the superior effectiveness of the proposed link-based methods, as compared to other clustering techniques included in this experiment. To better perceive this comparison, Fig. 6 summarizes the total performance ($B - W$) of each clustering method, sorted in the descending order, across all evaluation indices and six data sets. Note that the total performance ($B - W$) of any particular algorithm is specified as the difference between corresponding values of B and W. It can be seen that all link-based methods perform better than their competitors. In fact, these LCE models have the highest five statistics of $B - W$, while CO+AL with a Type-II(Fixed-k) ensemble is the most effective among compared techniques. In addition, Cobweb and Squeezer perform better than the other three categorical data clustering algorithms. Another

TABLE 3
The Pairwise Performance Comparison among Examined Clustering Methods

Ensemble Type	Method	CA		NMI		AR	
		B	W	B	W	B	W
I	LCE	171	35	141	68	151	61
	Base	137	78	141	80	145	73
	CO+SL	34	180	87	138	52	167
	CO+AL	72	143	128	94	92	121
	CSPA	116	93	135	82	112	98
	HGPA	23	196	18	211	27	203
	MCLA	60	166	45	162	62	161
II-Fixed-k	LCE	217	4	212	9	204	14
	Base	133	64	124	74	116	83
	CO+SL	29	171	41	152	32	165
	CO+AL	137	48	141	44	147	41
	CSPA	92	92	84	101	86	105
	HGPA	101	82	73	94	96	94
	MCLA	104	76	96	80	108	77
II-Random-k	LCE	219	4	209	12	208	12
	Base	122	55	109	62	113	66
	CO+SL	20	179	40	160	31	172
	CO+AL	94	82	99	69	123	59
	CSPA	70	113	51	135	55	130
	HGPA	68	122	43	142	56	135
	MCLA	66	116	77	100	72	110
III-Fixed-k	LCE	200	7	193	14	185	16
	Base	114	55	85	72	97	75
	CO+SL	20	167	41	143	37	165
	CO+AL	120	45	127	46	150	31
	CSPA	77	90	59	96	69	106
	HGPA	79	84	56	103	72	102
	MCLA	91	68	81	74	90	78
III-Random-k	LCE	206	6	195	15	189	11
	Base	97	61	67	76	88	75
	CO+SL	17	178	38	150	36	168
	CO+AL	83	67	88	62	106	45
	CSPA	58	112	41	125	50	126
	HGPA	57	117	43	136	41	133
	MCLA	56	100	69	102	67	81
Conventional methods	Squeezer	112	105	119	89	123	93
	GAClust	54	131	56	152	60	139
	k-modes	82	91	69	91	92	82
	CLOPE	80	137	128	99	106	111
	Cobweb	123	91	142	77	125	87

For each evaluation index, "B" and "W" denote the number of times that a particular method performs significantly "better" and "worse" than the others.

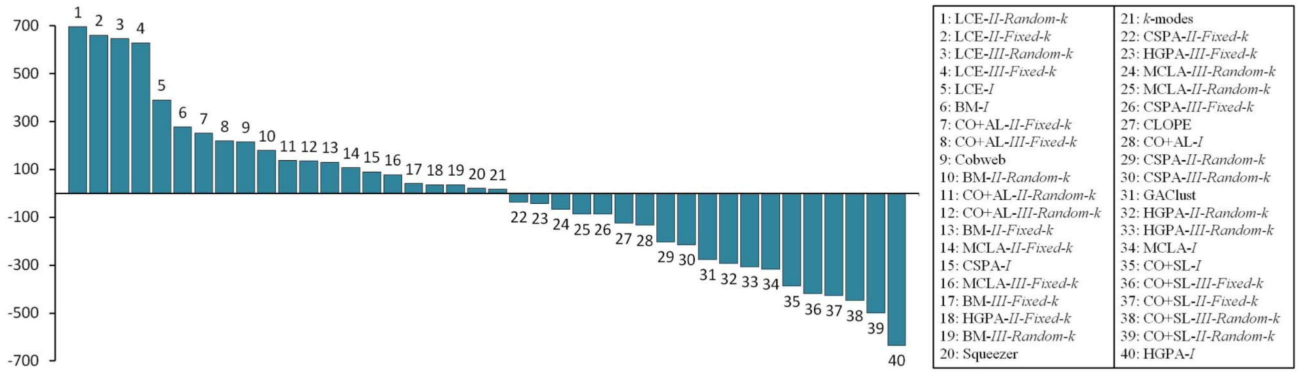


Fig. 6. The statistics of total performance ($B - W$) at 95 percent confidence level, summarized across all evaluation indices and six data sets, and sorted in descending order.

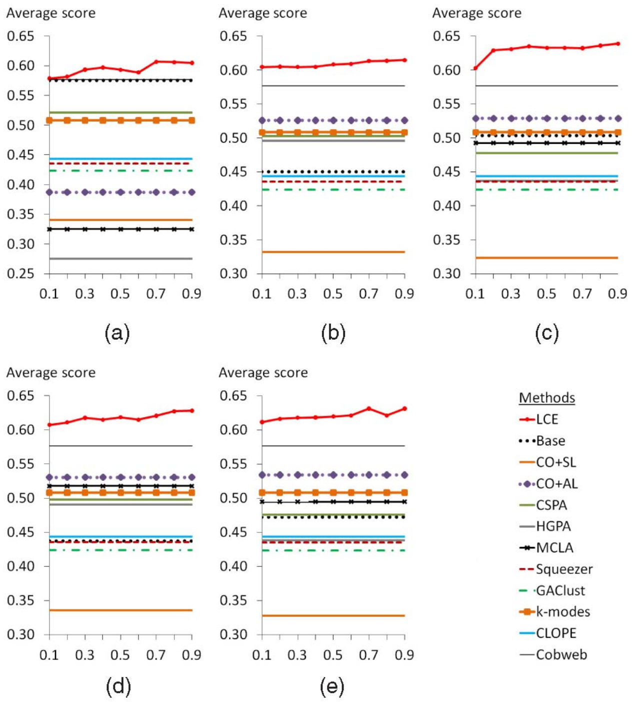


Fig. 7. The relations between $DC \in \{0.1, 0.2, \ldots, 0.9\}$ and the performance of the LCE models (the averages across all validity indices and six data sets), whose values are presented in X-axis and Y-axis, respectively. Note that the performance of other clustering methods is also included for a comparison purpose. (a) Ensemble Type I. (b) Ensemble Type II-Fixed-k. (c) Ensemble Type II-Random-k. (d) Ensemble Type III-Fixed-k. (e) Ensemble Type III-Random-k.

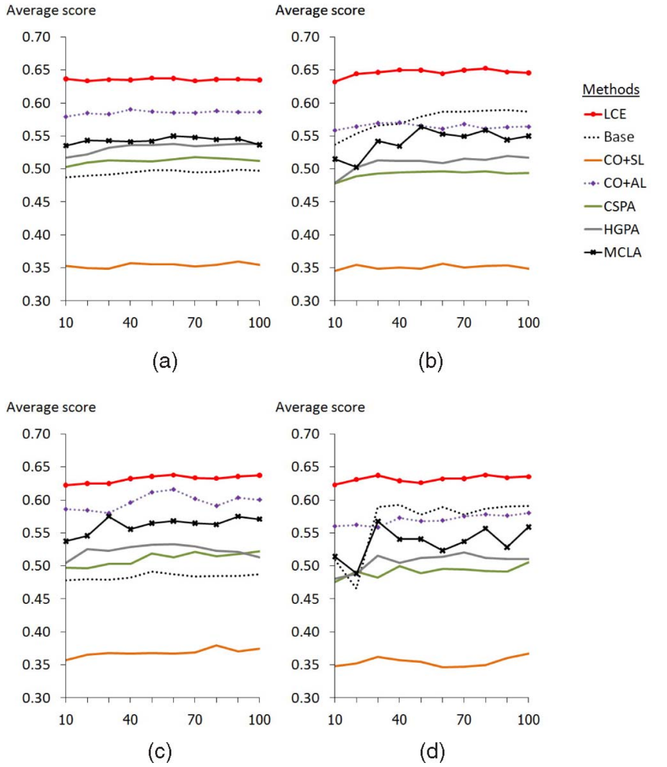


Fig. 8. Performance of different cluster ensemble methods in accordance with ensemble size ($M \in \{10, 20, \ldots, 100\}$), as the averages of validity measures (CA, NMI, and AR) across six data sets. (a) Ensemble Type II-Fixed-k. (b) Ensemble Type II-Random-k. (c) Ensemble Type III-Fixed-k. (d) Ensemble Type III-Random-k.

important investigation is on the subject of relations between performance of experimented cluster ensemble methods and different types of ensemble being explored in the present evaluation. To this point, it has been demonstrated that the LCE approach is more accurate than other cluster ensemble methods, across all examined ensemble types. Further details of the results and discussion regarding the effect of ensemble types on the performance of LCE are provided in Section II-B of the online supplementary.

4.4 Parameter and Complexity Analysis

The parameter analysis aims to provide a practical means by which users can make the best use of the link-based framework. Essentially, the performance of the resulting technique is dependant on the decay factor (i.e., $DC \in [0, 1]$), which is used in estimating the similarity among clusters and association degrees previously unknown in the original BM.

We varied the value of this parameter from 0.1 through 0.9, in steps of 0.1, and obtained the results in Fig. 7. Note that the presented results are obtained with the ensemble size (M) of 10. The figure clearly shows that the results of LCE are robust across different ensemble types, and do not depend strongly on any particular value of DC. This makes it easy for users to obtain high-quality, reliable results, with the best outcomes being obtained with values of DC between 0.7 and 0.9. Although there is variation in response across the DC values, the performance of LCE is always better than any of the other clustering methods included in this assessment. Another important observation is that the effectiveness of the link-based measure decreases as DC becomes smaller. Intuitively, the significance of disclosed associations becomes trivial when DC is low. Hence, they may be overlooked by a consensus function and the quality of the resulting data partition is not improved.

Another parameter that may determine the quality of data partition generated by a cluster ensemble technique is the ensemble size (M). Intuitively, the larger an ensemble is, the better the performance becomes. According to Fig. 8 which is obtained with a DC of 0.9, this heuristic is generally applicable to LCE with Type-II and Type-III ensembles, where its average quality measures (across all validity indices and six data sets) gradually incline to the increasing value of $M \in \{10, 20, \ldots, 100\}$. Furthermore, LCE performs consistently better than its competitors with all different ensemble sizes, while CO+SL is apparently the least effective. Note that a bigger ensemble leads to an improved accuracy, but with the trade-off of runtime—but, again, even the worst results of LCE are better than the best results of the other methods.

Besides previous quality assessments, computational requirements of the link-based method are discussed here. Primarily, the time complexity of creating the RM is $O(P^2 + NP)$, where N is the number of data points. While P denotes the number of clusters in a Type-II or Type-III ensemble, it represents the cardinality of all categorical values in a direct ensemble (i.e., Type-I). Please consult Section III in the online supplementary for the details of the scalability evaluation.

5 Discussion

The difficulty of categorical data analysis is characterized by the fact that there is no inherent distance (or similarity) between attribute values. The RM matrix that is generated within the LCE approach allows such measure between values of the same attribute to be systematically quantified. The concept of link analysis [34], [35], [36] is uniquely applied to discover the similarity among attribute values, which are modeled as vertices in an undirected graph. In particular, two vertices are similar if the neighboring contexts in which they appear are similar. In other words, their similarity is justified upon values of other attributes with which they co-occur. While the LCE methodology is novel for the problem of cluster ensemble, the concept of defining similarity among attribute values (especially with the case of "direct" ensemble, Type-I) has been analogously adopted by several categorical data clustering algorithms.

Initially, the problem of defining a context-based similarity measure has been investigated in [61] and [62]. In particular, an iterative algorithm, called "Iterated Contextual Distances (ICD)," is introduced to compute the proximity between two values. Similar to LCE, the underlying distance metric is based on the occurrence statistics of attribute values. However, the fundamental information model that is used by ICD and LCE to capture the associations between data points and attribute values are notably different: a sequential probabilistic chain and a link network for ICD and LCE, respectively. Note that LCE makes use of WTQ that is a single-pass similarity algorithm, while ICD requires the chain model to be randomly initialized and iteratively updated to a fixed point.

Despite pursuing an objective analogous to that of the LCE approach, several categorical data clustering methods have been developed using different mechanisms to specify a distance between attribute values: STIRR, ROCK, and CACTUS, for instance. STIRR [13] is an iterative algorithm based on nonlinear dynamical systems. A database is encoded into a graph structure, where each weighted node stands for a specific attribute value. STIRR iteratively updates the weight configuration until a stable point (called "basin") is reached. This is achieved using a user-defined "combiner function" to estimate a node weight from those of others that associate to the same data records. Unlike LCE, the similarity between any node pair cannot be explicitly measured here. In fact, STIRR only divides nodes of each attribute into two groups (one with large positive weights and the other with small negative weights) that correspond to projections of clusters on the attribute. Yet, the postprocessing required to generate the actual clusters is nontrivial and not addressed in the original work. While LCE is generally robust to parameter settings, it is hard to analyze the stability of the STIRR system for any useful combiner function [63]. Rigorous experimentation and fine tuning of parameters are needed for the generation of a meaningful clustering [64].

ROCK [14] makes use of a link graph, in which nodes and links represent data points (or tuples) and their similarity, respectively. Two tuples are similar if they shared a large number of attribute values. Note that the link connecting two nodes is included only when the corresponding similarity exceeds a user-defined threshold. With tuples being initially regarded as singleton clusters, ROCK merges clusters in an agglomerative hierarchical fashion, while optimizing a cluster quality that is defined in terms of the number of links across clusters. Note that the graph models used by ROCK and LCE are dissimilar—the graph of data points and that of attribute values (or clusters), respectively. Since the number of data points is normally greater than that of attribute values, ROCK is less efficient than LCE. As a result, it is unsuitable for large data sets [15]. Also, the selection of a "smooth function" that is used to estimate a cluster quality is a delicate and difficult task for average users [17].

CACTUS [16] also relies on the co-occurrence among attribute values. In essence, two attribute values are strongly connected if their support (i.e., the proportion of tuples in which the values co-occur) exceeds a prespecified value. By extending this concept to all attributes, CACTUS searches for the "distinguishing sets," which are attribute value sets that uniquely occur within only one cluster. These sets correspond to cluster projections that can be combined to formulate the final clusters. Unlike LCE, the underlying problem is not designed using a graph based concept. It is also noteworthy that CACTUS and its recent extension [65] assume each cluster to be identified by a set of attribute values that occur in no other cluster. While such conjecture may hold true for some data sets, it is unnatural and unnecessary for the clustering process [15]. This rigid constraint is not implemented by the LCE method.

Besides these approaches, traditional categorical data analysis also utilizes the "market-basket" numerical representation of the nominal data matrix [50], [51]. This transformed matrix is similar to the BM, which has been refined to the RM counterpart by LCE. A similar attempt in [66] identifies the connection between "category utility" of the conceptual clustering (Cobweb) [12] and the classical objective function of k-means. As a result, the so-called market-basket matrix used by the former is transformed to a variation that can be efficiently utilized by the latter. The intuitions of creating this rescaled matrix and the RM are fairly similar. However, the methods used to generate them are totally different. LCE discovers unknown entries (i.e., "0") in the original BM from known entries ("1"), which are preserved and left unchanged. On the other hand, the method in [66] maps the attribute-value-specific "1" and "0" entries to the unique, standardized values. Unlike the RM, this matrix does not conserve the known fact ("1" entries), whose values are now different from one to another attribute.

Despite the fact that many clustering algorithms and LCE are developed with the capability of comparing attribute values in mind, they achieve the desired metric differently, using specific information models. LCE uniquely and explicitly models the underlying problem as the evaluation of link-based similarity among graph vertices, which stand for specific attribute values (for Type-I ensemble) or generated clusters (for Type-II and Type-III). The resulting system is more efficient and robust, as compared to other clustering techniques emphasized thus far. In addition to SPEC, many other classical clustering techniques, k-means and PAM among others, can be directly used to generate the final data partition from the proposed RM. The LCE framework is generic such that it can be adopted for analyzing other types of data.

6 Conclusion

This paper presents a novel, highly effective link-based cluster ensemble approach to categorical data clustering. It transforms the original categorical data matrix to an information-preserving numerical variation (RM), to which an effective graph partitioning technique can be directly applied. The problem of constructing the RM is efficiently resolved by the similarity among categorical labels (or clusters), using the Weighted Triple-Quality similarity algorithm. The empirical study, with different ensemble types, validity measures, and data sets, suggests that the proposed link-based method usually achieves superior clustering results compared to those of the traditional categorical data algorithms and benchmark cluster ensemble techniques. The prominent future work includes an extensive study regarding the behavior of other link-based similarity measures within this problem context. Also, the new method will be applied to specific domains, including tourism and medical data sets.

Acknowledgments

The authors are grateful to the anonymous referees for their constructive comments which have helped considerably in revising this paper.

References

[1] D.S. Hochbaum and D.B. Shmoys, "A Best Possible Heuristic for the K-Center Problem," *Math. of Operational Research,* vol. 10, no. 2, pp. 180-184, 1985.

[2] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley Publishers, 1990.

[3] A.K. Jain and R.C. Dubes, *Algorithms for Clustering.* Prentice-Hall, 1998.

[4] P. Zhang, X. Wang, and P.X. Song, "Clustering Categorical Data Based on Distance Vectors," *The J. Am. Statistical Assoc.,* vol. 101, no. 473, pp. 355-367, 2006.

[5] J. Grambeier and A. Rudolph, "Techniques of Cluster Algorithms in Data Mining," *Data Mining and Knowledge Discovery,* vol. 6, pp. 303-360, 2002.

[6] K.C. Gowda and E. Diday, "Symbolic Clustering Using a New Dissimilarity Measure," *Pattern Recognition,* vol. 24, no. 6, pp. 567-578, 1991.

[7] J.C. Gower, "A General Coefficient of Similarity and Some of Its Properties," *Biometrics,* vol. 27, pp. 857-871, 1971.

[8] Z. Huang, "Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values," *Data Mining and Knowledge Discovery,* vol. 2, pp. 283-304, 1998.

[9] Z. He, X. Xu, and S. Deng, "Squeezer: An Efficient Algorithm for Clustering Categorical Data," *J. Computer Science and Technology,* vol. 17, no. 5, pp. 611-624, 2002.

[10] P. Andritsos and V. Tzerpos, "Information-Theoretic Software Clustering," *IEEE Trans. Software Eng.,* vol. 31, no. 2, pp. 150-165, Feb. 2005.

[11] D. Cristofor and D. Simovici, "Finding Median Partitions Using Information-Theoretical-Based Genetic Algorithms," *J. Universal Computer Science,* vol. 8, no. 2, pp. 153-172, 2002.

[12] D.H. Fisher, "Knowledge Acquisition via Incremental Conceptual Clustering," *Machine Learning,* vol. 2, pp. 139-172, 1987.

[13] D. Gibson, J. Kleinberg, and P. Raghavan, "Clustering Categorical Data: An Approach Based on Dynamical Systems," *VLDB J.,* vol. 8, nos. 3-4, pp. 222-236, 2000.

[14] S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," *Information Systems,* vol. 25, no. 5, pp. 345-366, 2000.

[15] M.J. Zaki and M. Peters, "Clicks: Mining Subspace Clusters in Categorical Data via Kpartite Maximal Cliques," *Proc. Int'l Conf. Data Eng. (ICDE),* pp. 355-356, 2005.

[16] V. Ganti, J. Gehrke, and R. Ramakrishnan, "CACTUS: Clustering Categorical Data Using Summaries," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD),* pp. 73-83, 1999.

[17] D. Barbara, Y. Li, and J. Couto, "COOLCAT: An Entropy-Based Algorithm for Categorical Clustering," *Proc. Int'l Conf. Information and Knowledge Management (CIKM),* pp. 582-589, 2002.

[18] Y. Yang, S. Guan, and J. You, "CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD),* pp. 682-687, 2002.

[19] D.H. Wolpert and W.G. Macready, "No Free Lunch Theorems for Search," Technical Report SFI-TR-95-02-010, Santa Fe Inst., 1995.

[20] L.I. Kuncheva and S.T. Hadjitodorov, "Using Diversity in Cluster Ensembles," *Proc. IEEE Int'l Conf. Systems, Man and Cybernetics,* pp. 1214-1219, 2004.

[21] H. Xue, S. Chen, and Q. Yang, "Discriminatively Regularized Least-Squares Classification," *Pattern Recognition,* vol. 42, no. 1, pp. 93-104, 2009.

[22] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering Aggregation," *Proc. Int'l Conf. Data Eng. (ICDE),* pp. 341-352, 2005.

[23] N. Nguyen and R. Caruana, "Consensus Clusterings," *Proc. IEEE Int'l Conf. Data Mining (ICDM),* pp. 607-612, 2007.

[24] A.P. Topchy, A.K. Jain, and W.F. Punch, "Clustering Ensembles: Models of Consensus and Weak Partitions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 12, pp. 1866-1881, Dec. 2005.

[25] C. Boulis and M. Ostendorf, "Combining Multiple Clustering Systems," *Proc. European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD),* pp. 63-74, 2004.

[26] B. Fischer and J.M. Buhmann, "Bagging for Path-Based Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 11, pp. 1411-1415, Nov. 2003.

[27] C. Domeniconi and M. Al-Razgan, "Weighted Cluster Ensembles: Methods and Analysis," *ACM Trans. Knowledge Discovery from Data,* vol. 2, no. 4, pp. 1-40, 2009.

[28] X.Z. Fern and C.E. Brodley, "Solving Cluster Ensemble Problems by Bipartite Graph Partitioning," *Proc. Int'l Conf. Machine Learning (ICML),* pp. 36-43, 2004.

[29] A. Strehl and J. Ghosh, "Cluster Ensembles: A Knowledge Reuse Framework for Combining Multiple Partitions," *J. Machine Learning Research,* vol. 3, pp. 583-617, 2002.

[30] H. Ayad and M. Kamel, "Finding Natural Clusters Using Multiclusterer Combiner Based on Shared Nearest Neighbors," *Proc. Int'l Workshop Multiple Classifier Systems,* pp. 166-175, 2003.

[31] A.L.N. Fred and A.K. Jain, "Combining Multiple Clusterings Using Evidence Accumulation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 6, pp. 835-850, June 2005.

[32] S. Monti, P. Tamayo, J.P. Mesirov, and T.R. Golub, "Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data," *Machine Learning,* vol. 52, nos. 1/2, pp. 91-118, 2003.

[33] N. Iam-On, T. Boongoen, and S. Garrett, "Refining Pairwise Similarity Matrix for Cluster Ensemble Problem with Cluster Relations," *Proc. Int'l Conf. Discovery Science,* pp. 222-233, 2008.

[34] T. Boongoen, Q. Shen, and C. Price, "Disclosing False Identity through Hybrid Link Analysis," *Artificial Intelligence and Law,* vol. 18, no. 1, pp. 77-102, 2010.

[35] L. Getoor and C.P. Diehl, "Link Mining: A Survey," *ACM SIGKDD Explorations Newsletter,* vol. 7, no. 2, pp. 3-12, 2005.

[36] D. Liben-Nowell and J. Kleinberg, "The Link-Prediction Problem for Social Networks," *J. Am. Soc. for Information Science and Technology,* vol. 58, no. 7, pp. 1019-1031, 2007.

[37] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On Combining Classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 226-239, Mar. 1998.

[38] L.I. Kuncheva and D. Vetrov, "Evaluation of Stability of K-Means Cluster Ensembles with Respect to Random Initialization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 11, pp. 1798-1808, Nov. 2006.

[39] A.P. Topchy, A.K. Jain, and W.F. Punch, "A Mixture Model for Clustering Ensembles," *Proc. SIAM Int'l Conf. Data Mining,* pp. 379-390, 2004.

[40] X.Z. Fern and C.E. Brodley, "Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach," *Proc. Int'l Conf. Machine Learning (ICML),* pp. 186-193, 2003.

[41] Z. Yu, H.-S. Wong, and H. Wang, "Graph-Based Consensus Clustering for Class Discovery from Gene Expression Data," *Bioinformatics,* vol. 23, no. 21, pp. 2888-2896, 2007.

[42] S. Dudoit and J. Fridyand, "Bagging to Improve the Accuracy of a Clustering Procedure," *Bioinformatics,* vol. 19, no. 9, pp. 1090-1099, 2003.

[43] B. Minaei-Bidgoli, A. Topchy, and W. Punch, "A Comparison of Resampling Methods for Clustering Ensembles," *Proc. Int'l Conf. Artificial Intelligence,* pp. 939-945, 2004.

[44] X. Hu and I. Yoo, "Cluster Ensemble and Its Applications in Gene Expression Analysis," *Proc. Asia-Pacific Bioinformatics Conf.,* pp. 297-302, 2004.

[45] M. Law, A. Topchy, and A.K. Jain, "Multiobjective Data Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 424-430, 2004.

[46] G. Karypis and V. Kumar, "Multilevel K-Way Partitioning Scheme for Irregular Graphs," *J. Parallel Distributed Computing,* vol. 48, no. 1, pp. 96-129, 1998.

[47] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems,* vol. 14, pp. 849-856, 2001.

[48] M. Al-Razgan, C. Domeniconi, and D. Barbara, "Random Subspace Ensembles for Clustering Categorical Data," *Supervised and Unsupervised Ensemble Methods and Their Applications,* pp. 31-48, Springer, 2008.

[49] Z. He, X. Xu, and S. Deng, "A Cluster Ensemble Method for Clustering Categorical Data," *Information Fusion,* vol. 6, no. 2, pp. 143-151, 2005.

[50] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* pp. 207-216, 1993.

[51] P.N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining.* Addison Wesley, 2005.

[52] G. Jeh and J. Widom, "Simrank: A Measure of Structural-Context Similarity," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD),* pp. 538-543, 2002.

[53] F. Fouss, A. Pirotte, J.M. Renders, and M. Saerens, "Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation," *IEEE Trans. Knowledge and Data Eng.,* vol. 19, no. 3, pp. 355-369, Mar. 2007.

[54] E. Minkov, W.W. Cohen, and A.Y. Ng, "Contextual Search and Name Disambiguation in Email Using Graphs," *Proc. Int'l Conf. Research and Development in IR,* pp. 27-34, 2006.

[55] P. Reuther and B. Walter, "Survey on Test Collections and Techniques for Personal Name Matching," *Int'l J. Metadata, Semantics and Ontologies,* vol. 1, no. 2, pp. 89-99, 2006.

[56] L.A. Adamic and E. Adar, "Friends and Neighbors on the Web," *Social Networks,* vol. 25, no. 3, pp. 211-230, 2003.

[57] U. Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing,* vol. 17, no. 4, pp. 395-416, 2007.

[58] A. Asuncion and D.J. Newman, "UCI Machine Learning Repository," School of Information and Computer Science, Univ. of California, http://www.ics.uci.edu/~mlearn/MLRepository.html, 2007.

[59] L. Hubert and P. Arabie, "Comparing Partitions," *J. Classification,* vol. 2, no. 1, pp. 193-218, 1985.

[60] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel Hypergraph Partitioning: Applications in VLSI Domain," *IEEE Trans. Very Large Scale Integration Systems,* vol. 7, no. 1, pp. 69-79, Mar. 1999.

[61] G. Das and H. Mannila, "Context-Based Similarity Methods for Categorical Attributes," *Proc. Principles of Data Mining and Knowledge. Discovery (PKDD),* pp. 201-211, 2000.

[62] G. Das, H. Mannila, and P. Ronkainen, "Similarity of Attributes by External Probes," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD),* pp. 16-22, 1998.

[63] Y. Zhang, A. Fu, C. Cai, and P. Heng, "Clustering Categorical Data," *Proc. Int'l Conf. Data Eng. (ICDE),* p. 305, 2000.

[64] M. Dutta, A.K. Mahanta, and A.K. Pujari, "QROCK: A Quick Version of the ROCK Algorithm for Clustering of Categorical Data," *Pattern Recognition Letters,* vol. 26, pp. 2364-2373, 2005.

[65] E. Abdu and D. Salane, "A Spectral-Based Clustering Algorithm for Categorical Data Using Data Summaries," *Proc. Workshop Data Mining using Matrices and Tensors,* pp. 1-8, 2009.

[66] B. Mirkin, "Reinterpreting the Category Utility Function," *Machine Learning,* vol. 45, pp. 219-228, 2001.

Natthakan Iam-On received the PhD degree in computer science from Aberystwyth University. She is a lecturer in the School of Information Technology, Mae Fah Luang University, Thailand. Her research focuses on data clustering, cluster ensembles and applications to biomedical data analysis, advance database technology and knowledge discovery.

Tossapon Boongoen received the PhD degree in artificial intelligence from Cranfield University and worked as a postdoctoral research associate at Aberystwyth University, United Kingdom. He is a lecturer in the Department of Mathematics and Computer Science, Royal Thai Air Force Academy, Thailand. His research interests include data mining, link analysis, data clustering, fuzzy aggregation, and classification system.

Simon Garrett founded and is CEO of Aispire Consulting Ltd., having worked at Aberystwyth University in the Department of Computer Science as both a lecturer and researcher. His research has been in machine learning and clustering, which have been his interests for more than 10 years. He has been recognized for his contribution to artificial immune systems, and has done work on their ability to cluster data and find cluster centers quickly and efficiently.

Chris Price received the BSc degree in computer science from Aberystwyth University, United Kingdom, in 1979 and, after eight years building artificial intelligence systems in industry, returned to academia in 1986. He received the PhD degree in computer science from Aberystwyth University in 1994, where he was made a full professor in 1999. Much of his research has concentrated on reasoning from models to build design and diagnosis tools for use by engineers in automotive and aerospace companies.