www.elsevier.com/locate/pr

A clustering ensemble: Two-level-refined co-association matrix with path-based transformation

Caiming Zhong, Xiaodong Yue, Zehua Zhang, Jingsheng Lei

PII: S0031-3203(15)00074-6
DOI: http://dx.doi.org/10.1016/j.patcog.2015.02.014
Reference: PR5353

To appear in: *Pattern Recognition*

Received date: 29 August 2014
Revised date: 6 December 2014
Accepted date: 18 February 2015

Cite this article as: Caiming Zhong, Xiaodong Yue, Zehua Zhang, Jingsheng Lei, A clustering ensemble: Two-level-refined co-association matrix with path-based transformation, *Pattern Recognition,* http://dx.doi.org/10.1016/j.patcog.2015.02.014

A clustering ensemble: Two-level-refined co-association matrix with path-based transformation

Caiming Zhong[a,*], Xiaodong Yue[b], Zehua Zhang[c], Jingsheng Lei[d]

[a] College of Science and Technology, Ningbo University, 315211 Ningbo, China
[b] Department of Computer Science and Technology, Shanghai University, 200444 Shanghai, China
[c] College of Computer Science and Technology, Taiyuan University of Technology, 030024 Taiyuan, China
[d] School of Computer Science and Technology, Shanghai University of Electric Power, 200090 Shanghai, China

Abstract

The aim of clustering ensemble is to combine multiple base partitions into a robust, stable and accurate partition. One of the key problems of clustering ensemble is how to exploit the cluster structure information in each base partition. Evidence accumulation is an effective framework which can convert the base partitions into a co-association matrix. This matrix describes the frequency of a pair of points partitioned into the same cluster, but ignores some hidden information in the base partitions. In this paper, we reveal some of those information by refining the co-association matrix from data point and base cluster level. From the data point level, as pairs of points in the same base cluster may have varied similarities, their contributions to the co-association matrix can be different. From the cluster level, since the base clusters may have diversified qualities, the contribution of a base cluster as a whole can also be different from those of others. After being refined, the co-association matrix is transformed into a path-based similarity matrix so that more global information of the cluster structure is incorporated into the matrix. Finally, spectral clustering is applied to the matrix to generate the final clustering result. Experimental results on 8 synthetic and 8 real data sets demonstrate that the clustering ensemble based

*Corresponding author. Tel.:+86-21-69589867
Email address: zhongcaiming@nbu.edu.cn (Caiming Zhong)

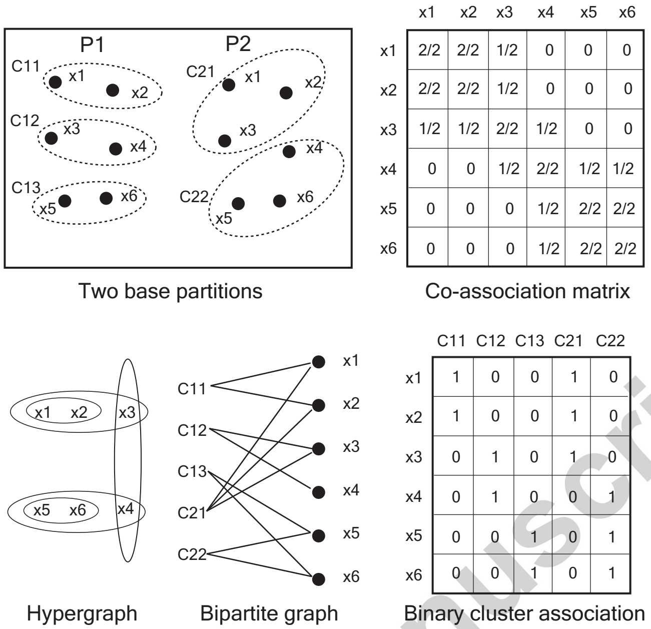
on the refined co-association matrix outperforms some state-of-the-art clustering ensemble schemes.

Keywords: Clustering, cluster ensembles, co-association matrix, path-based measure

1. Introduction

Clustering ensemble detects the cluster structure of a data set from a bag of clusterings, which are called base partitions. There exist a number of clustering ensemble methods proposed in the literature [1, 4, 5, 12, 14, 19, 22, 26, 36, 42, 44, 46, 47]. In general, a clustering ensemble algorithm consists of three components, of which the functions are to produce base partitions, represent the base partitions and generate the final partition. The base partitions are usually produced by different clustering algorithms or one with different parameters. Many ensemble approaches use K-means to generate the base partitions, as it is the most popular and efficient clustering algorithm. Since its initial cluster centers are randomly selected, the clustering results are not unique. The number of clusters in a base partition can be fixed or selected randomly from an interval [14, 22].

A suitable representation of the base partitions can discover more hidden information of the cluster structure, and lead to improvement of the quality of the final clustering. In Fig. 1, four widely used representations of two base partitions are illustrated. The first two are graph-based, i.e. hypergraph [41] and bipartite graph [10], and the other two are matrix-based, i.e. co-association matrix [14] and binary cluster association matrix [22]. In the hypergraph, each data point is a vertex, and each closed curve is a hyperedge, which represents a base cluster. In the bipartite graph, the base clusters and the data points are two disjoint sets of vertices, and an edge between a data point and a cluster vertex indicates the data point belongs to the cluster. In the binary cluster association matrix, each row shows whether a data point resides in a cluster of each column. These three representations convey the similar information



	x1	x2	x3	x4	x5	x6
x1	2/2	2/2	1/2	0	0	0
x2	2/2	2/2	1/2	0	0	0
x3	1/2	1/2	2/2	1/2	0	0
x4	0	0	1/2	2/2	1/2	1/2
x5	0	0	0	1/2	2/2	2/2
x6	0	0	0	1/2	2/2	2/2

	C11	C12	C13	C21	C22
x1	1	0	0	1	0
x2	1	0	0	1	0
x3	0	1	0	1	0
x4	0	1	0	0	1
x5	0	0	1	0	1
x6	0	0	1	0	1

Figure 1: Four representations of two base partitions.

of the base partitions: What data points a base cluster is composed of. The co-association matrix depicts the average frequency that a pair of points is partitioned into the same cluster. Compared with the first three representations, a co-association matrix carries less information of the base partitions, as one can not figure out from it what data points a base cluster contains. However, it can be viewed as a similarity matrix and act as a kernel transformation [15].

Different consensus methods have been applied to the different representations in the literature. For a graph-based representation, a graph cut is used. For example, MERIT [27] is used to cut the hypergraph in [41]; a multi-way spectral graph partitioning [37] is applied to the bipartite graph in [10]. Since the binary association matrix can be regarded as a transformed new data set, a regular clustering algorithm such as K-means, PAM [29] and spectral graph partitioning [37] can be used to produce the final clustering. For the co-association matrix, as it may function as a similarity matrix, any clustering algorithm that processes a similarity matrix can be used for the final results.

Some studies about clustering ensemble have been focused on improving the representation of the base partitions. Iam-On et al. [22] refined the binary cluster association matrix by considering the hidden information of the base clusters. In the original matrix, a data point only belongs to one cluster of a partition, but Iam-On et al. showed that it can also belong to the other clusters of the same partition to some degree.

Fred and Jain [14] treated the contribution of a pair of points of a base partition to the co-association matrix as 1 or 0, that is, when a pair is in the same cluster of the partition, the contribution is 1, otherwise is 0. This treatment ignores the different contributions of two pairs in the same cluster. To improve this situation, Wang et al. [46] took the number of points in a cluster into account, and Iam-On et al. [23] incorporated link-based information into the matrix.

Lourenço et al. extensively studied co-association matrix based clustering ensemble [31, 32, 33, 34]. In [32], the co-association matrix is repeatedly and probabilistically used to assign labels by minimizing a Bregman divergence between the matrix and a probability distribution matrix. In [34], Lourenço et al. determined a median partition by the evidence accumulation and the different importance of the base partitions. To scale co-association matrix based ensemble to large data sets, a partial set of the co-occurrences is used to reduce the computational time and space [33]. In [31], a generative dyadic aspect model is employed to construct the co-association matrix.

To produce the base partitions, K-means is mainly used in the literature. But it should be careful to select the number of clusters for K-means. When the number is set to a big value, the clusters will have good homogeneity, but some global structure information may be missed. If it is set to a small value, the number of heterogenous pairs will increase.

In this paper, we refine the co-association matrix from the data point and base cluster level to exploit some hidden information, and use path-based measure [11] to consider some global information.

2. Proposed method

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a data set, where $\mathbf{x}_i = (x_{i1}, ..., x_{id})^T \in \mathcal{R}^d$, d is the dimensionality of X, and N is the number of data points. A clustering ensemble repeatedly partitions X into M base partitions, $P_1, ..., P_M$, and then employs a consensus function to produce a final solution, $P = \{C_1, ..., C_K\}$, where $P_i = \{C_{i1}, ..., C_{iK_i}\}$, C_{ij} denotes the jth cluster of P_i, K_i is the number of clusters in P_i, K is the number of clusters in P, C_i is a cluster of P.

Evidence accumulation framework generates a co-association matrix (CM) from $P_1, ..., P_M$ as follows [14]:

$$CM(i,j) = \frac{1}{M}\sum_{m=1}^{M}\sum_{l=1}^{K_m}\mathcal{I}(i,j,C_{ml}) \tag{1}$$

where $CM(i,j)$ denotes an entry of CM, C_{ml} is the lth base cluster in P_m, and $\mathcal{I}(i,j,C_{ml})$ is an indicator:

$$\mathcal{I}(i,j,C_{ml}) = \begin{cases} 1, \text{if } \mathbf{x}_i \in C_{ml} \wedge \mathbf{x}_j \in C_{ml} \\ 0, \text{otherwise} \end{cases} \tag{2}$$

Fred and Jain [14] viewed CM as a similarity matrix, and applied single-linkage to it to generate the final partition. Although the co-association matrix is effective, for some data sets with arbitrary shapes the matrix may fail to recognize the cluster structure.

2.1. Two-level-refined co-association matrix

2.1.1. Data point level refinement

A co-association matrix is composed of the average co-occurrence frequencies of all pairs of points. According to Eq. (2), all pairs in a cluster have the same co-occurrence contribution, i.e. 1. But in fact, even if two pairs possess the same cluster label in a base partition, they may have different probabilities of being in the cluster simultaneously.

Suppose that base cluster $C_{ml} = \{\mathbf{x}_1, ..., \mathbf{x}_n\} \in \mathcal{R}^d$ is a finite, independent and identically distributed sample set. A probability density estimate $\hat{p}(\mathbf{x})$ can be obtained from C_{ml} by employing Parzen window density estimator [18]:

$$\hat{p}(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n}\varphi(\frac{\mathbf{x}-\mathbf{x}_i}{h}) \tag{3}$$

where h is a window width, and φ is a window function. Let φ be a Gaussian kernel function:

$$\varphi(\mathbf{u}) = \frac{1}{(2\pi)^{d/2}}e^{-\frac{1}{2}\mathbf{u}^T\mathbf{u}} \tag{4}$$

then the probability density estimate is:

$$\hat{p}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}nh^d}\sum_{i=1}^{n}e^{-\frac{1}{2h^2}(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)} \tag{5}$$

Suppose data point $\mathbf{x}_a \in C_{ml}$, the class-conditional probability $p(\mathbf{x}_a|C_{ml})$ is:

$$p(\mathbf{x}_a|C_{ml}) = \frac{1}{(2\pi)^{d/2}nh^d}\sum_{i=1}^{n}e^{-\frac{1}{2h^2}(\mathbf{x}_a-\mathbf{x}_i)^T(\mathbf{x_a}-\mathbf{x}_i)} \tag{6}$$

For simplicity, we use the mean of C_{ml}, $\bar{\mathbf{x}}$, to replace $\mathbf{x}_i$:

$$p(\mathbf{x}_a|C_{ml}) = \frac{1}{(2\pi)^{d/2}h^d}e^{-\frac{1}{2h^2}(\mathbf{x}_a-\bar{\mathbf{x}})^T(\mathbf{x_a}-\bar{\mathbf{x}})} \tag{7}$$

Suppose data point $\mathbf{x}_b \in C_{ml}$, then the co-occurrence probability of $\mathbf{x}_a$ and $\mathbf{x}_b$ in C_{ml} is:

$$\begin{aligned} p(\mathbf{x}_a, \mathbf{x}_b|C_{ml}) &= \frac{1}{(2\pi)^d h^{2d}}e^{-\frac{1}{2h^2}[(\mathbf{x}_a-\bar{\mathbf{x}})^T(\mathbf{x_a}-\bar{\mathbf{x}})+(\mathbf{x}_b-\bar{\mathbf{x}})^T(\mathbf{x_b}-\bar{\mathbf{x}})]} \\ &= \frac{1}{(2\pi)^d h^{2d}}e^{-\frac{1}{2h^2}[(\mathbf{x}_a-\mathbf{x}_b)^T(\mathbf{x_a}-\mathbf{x}_b)+2(\mathbf{x}_a-\bar{\mathbf{x}})^T(\mathbf{x_b}-\bar{\mathbf{x}})]} \end{aligned} \tag{8}$$

As $\mathbf{x}_a$ and $\mathbf{x}_b$ are independent, $E[(\mathbf{x}_a-\bar{\mathbf{x}})^T(\mathbf{x}_b-\bar{\mathbf{x}})] = \mathbf{0}$ holds, then we have

$$\begin{aligned} p(\mathbf{x}_a, \mathbf{x}_b|C_{ml}) &= \frac{1}{(2\pi)^d h^{2d}}e^{-\frac{1}{2h^2}(\mathbf{x}_a-\mathbf{x}_b)^T(\mathbf{x_a}-\mathbf{x}_b)} \\ &= \frac{1}{(2\pi)^d h^{2d}}e^{-\frac{1}{2h^2}(\|\mathbf{x}_a-\mathbf{x}_b\|)^2} \end{aligned} \tag{9}$$

where $\|\mathbf{x}_a - \mathbf{x}_b\|$ denotes the Euclidean distance between $\mathbf{x}_a$ and $\mathbf{x}_b$.

From Eq. (9), we can see that the co-occurrence probability $p(\mathbf{x}_a, \mathbf{x}_b|C_{ml})$ is determined by $\|\mathbf{x}_a - \mathbf{x}_b\|$ and window width h. When h is fixed, there exists

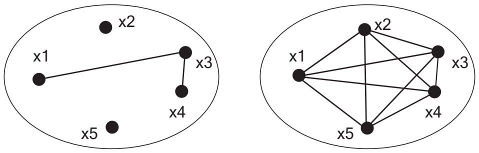


Figure 2: In the left, $\mathbf{x}_1$ and $\mathbf{x}_3$ have a less co-occurrence possibility than $\mathbf{x}_3$ and $\mathbf{x}_4$. In the right, each pair has a similarity measured by the co-association matrix and the cluster stability can be measured by the average of all pairs' similarities.

a negative correlation between $p(\mathbf{x}_a, \mathbf{x}_b|C_{ml})$ and $\|\mathbf{x}_a - \mathbf{x}_b\|$. An example in the left of Fig. 2 shows the correlation. Although the five data points are partitioned into one cluster, $\mathbf{x}_1$ and $\mathbf{x}_3$ have a small co-occurrence possibility while $\mathbf{x}_3$ and $\mathbf{x}_4$ have a relatively large one. As the window width h may affect the probability and it is difficult to select a suitable value for h, the probability can be directly and simply defined by $\|\mathbf{x}_a - \mathbf{x}_b\|$:

$$p(\mathbf{x}_a, \mathbf{x}_b|C_{ml}) = 1 - \frac{\|\mathbf{x}_a - \mathbf{x}_b\|}{L} \tag{10}$$

where L is the maximum distance of two points in C_{ml}. This definition does not need to set window width h, but keeps the negative correlation and normalizes the value to [0,1].

According to the above discussion, we refine the the co-association matrix as:

$$CM'(i,j) = \frac{1}{M}\sum_{m=1}^{M}\sum_{l=1}^{K_m}\mathcal{I}'(i,j,C_{ml}) \tag{11}$$

where CM' is the refined co-association matrix, C_{ml} is the lth base cluster in P_m, and $\mathcal{I}'(i,j,C_{ml})$ is an indicator:

$$\mathcal{I}'(i,j,C_{ml}) = \begin{cases} 1 - \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\max_{\mathbf{x}_p,\mathbf{x}_q \in C_{ml}}(\|\mathbf{x}_p - \mathbf{x}_q\|)}, \text{if } \mathbf{x}_i, \mathbf{x}_j \in C_{ml} \\ 0, \text{otherwise} \end{cases} \tag{12}$$

In the above definition of $\mathcal{I}'$, we use $p(\mathbf{x}_a, \mathbf{x}_b|C_{ml})$ to replace 1 when $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same cluster.

2.1.2. Cluster level refinement

The refinement in the data point level differentiates the co-occurrence probabilities of pairs of points in the same cluster. In the cluster level, the base clusters have different qualities, that is, some of them are of high homogeneity but some of the others are of low. Therefore, the base clusters should have different contributions to the co-association matrix. However, one can not directly compute the homogeneity of a cluster.

In the literature, stability of a cluster is employed to evaluate its quality [2, 20, 45]. Hennig [20] presented a cluster-wise assessment of cluster stability and Volkovich et al. [45] proposed a kernel-based cluster validity index. But here we are only interested in the stability of a single cluster. Although Alizadeh et al. gave a such definition by averaging the Normalized Mutual Information [2], the computational cost is expensive. To make use of the information provided by the co-association matrix, we employ an intra-cluster similarity [35]. The right of Fig. 2 illustrates the concept.

The stability of a cluster is defined as:

$$S(C_{ml}) = \frac{1}{|C_{ml}| \times (|C_{ml}| - 1)/2} \sum_{\mathbf{x}_i, \mathbf{x}_j \in C_{ml} \wedge i \neq j} CM'(i, j) \tag{13}$$

where $S(C_{ml})$ is the stability of base cluster C_{ml}, $|C_{ml}|$ denotes the number of points in C_{ml}, CM' is the point level refined co-association matrix and defined in Eq. 11.

It can be normalized into interval [0, 1] as:

$$S'(C_{ml}) = \frac{S(C_{ml}) - \min_{C_i \in Q}(S(C_i))}{\max_{C_i \in Q}(S(C_i)) - \min_{C_i \in Q}(S(C_i))} \tag{14}$$

where $S'(C_{ml})$ is the normalized stability of C_{ml}, Q is the set of all base clusters.

Combining the point level refinement and the cluster stability, the two-level-refined co-association matrix is defined as:

$$CM''(i, j) = \frac{1}{M} \sum_{m=1}^{M} \sum_{l=1}^{K_m} \mathcal{I}''(i, j, C_{ml}) \tag{15}$$

where CM'' is the two-level-refined co-association matrix, and $\mathcal{I}''(i, j, C_{ml})$ is an indicator:

$$\mathcal{I}''(i,j,C_{ml}) = \begin{cases} (1 - \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\max_{\mathbf{x}_p,\mathbf{x}_q \in C_{ml}}(\|\mathbf{x}_p - \mathbf{x}_q\|)}) \times S'(C_{ml}), \text{if } \mathbf{x}_i, \mathbf{x}_j \in C_{ml} \\ 0, \text{otherwise} \end{cases} \tag{16}$$

2.2. Base partitions and consensus method

2.2.1. Base partitions

K-means is used to generate the base partitions in this study. We care about two parameters of K-means: the maximal number of iterations and the number of clusters in each base partition.

Many clustering ensemble methods that employ K-means to produce the base partitions do not discuss the number of iterations. In Matlab, the default maximal number is set to 100. However, a small number of iterations makes K-means be prone to failure in convergence so that the base partitions have more diversity. In this paper, we set the number to 4, more detailed discussion on this parameter will be given in Section 4.

The number of clusters of a base partition is also an important parameter for the quality of the final partition. In general, a fixed K or a randomly selected K are the options [14, 22]. For the fixed K, it can be set to the true K, or $\sqrt{N}$, while for the randomly selected, it can be selected from $[2, \sqrt{N}]$.

We suppose that in the phase of producing the base partitions the true K is not available. Therefore, we prefer to the fixed $K = \sqrt{N}$ and randomly selected K.

Since the number of iterations in K-means is set to a small value, the number of the base partitions M should be relatively large so that the final partition is more stable. We set $M = 100$ for all experiments and discuss it in detail in Section 4.

2.2.2. Consensus method

A consensus method produces the final partition from the refined co-association matrix. Before a consensus method is applied, we perform a path-based transformation on the matrix.

The *path-based measure* was proposed and applied to clustering by Fischer and Buhmann [11]. A path-based distance is a *minimax* distance, which is defined as follows.

Definition. Let $G = (X, E)$ be an undirected graph and $\mathscr{P}_{ij}$ denote the set of all possible paths from vertex $\mathbf{x}_i$ to $\mathbf{x}_j$ in G. The *minimax* distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is:

$$D(\mathbf{x}_i, \mathbf{x}_j) = \min_{\mathcal{P} \in \mathscr{P}_{ij}} \{ \max_{1 \leq m < |\mathcal{P}|} w(\mathcal{P}[m], \mathcal{P}[m+1]) \} \tag{17}$$

where $D(\mathbf{x}_i, \mathbf{x}_j)$ is the minimax distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathcal{P}$ is a path from vertex $\mathbf{x}_i$ to $\mathbf{x}_j$, $\mathcal{P}[m]$ is the mth vertex along the path, and $w(\mathbf{x}_p, \mathbf{x}_q)$ is the Euclidean distance between $\mathbf{x}_p$ and $\mathbf{x}_q$.

The main property of the path-based measure is that the similarity is transitive: "*My neighbor's neighbor can be also my neighbor.*" Note that a path-based similarity is a *maximin* distance as:

$$SIM(\mathbf{x}_i, \mathbf{x}_j) = \max_{\mathcal{P} \in \mathscr{P}_{ij}} \{ \min_{1 \leq m < |\mathcal{P}|} w(\mathcal{P}[m], \mathcal{P}[m+1]) \} \tag{18}$$

The reason that path-based measure is applied. In semi-supervised learning, the prior assumption of consistency is made [49]: (1) Neighboring data points are likely in the same cluster; (2) Data points on the same structure or manifold are likely in the same cluster. This argument is similar to the following assumption of clustering, called *cluster assumption*: Data points are likely in the same cluster if there is a path connecting them passing through regions of high density only [8, 39]. One can see that neighboring data points imply the local cluster structure, while data points on the same manifold imply the global cluster structure. In other words, some compact clusters consisting of nearby points such as C_2 in Fig 3(a) can be recognized by the local assumption, and some non-compact cluster such as C_1 can be detected by the global assumption.

If the base partitions are produced by K-means, the derived co-association matrix will mainly focus on the information of the local cluster structure, especially when the number of clusters is big. As the goal of K-means is to minimize the within-cluster sum of squares, data points in the same cluster are adjacent.

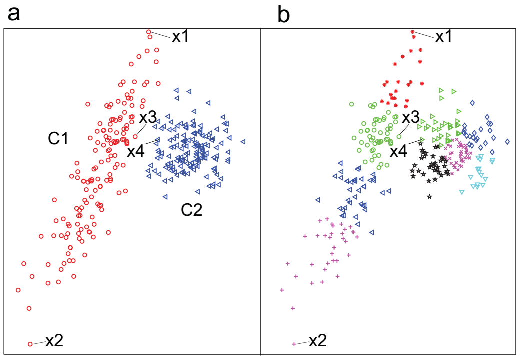


Figure 3: In (a), a data set consisting of 300 points, which form two clusters C_1 and C_2; $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ are three data points in C_1, $\mathbf{x}_4$ is a data point in C_2. In (b), the clustering is generated by K-means with the number of clusters 9.

In Fig. 3(b), the partition is produced by K-means and has 9 clusters. Being composed of neighboring points, all clusters are (relatively) compact. Although point $\mathbf{x}_1$ and $\mathbf{x}_2$ are from C_1 in Fig. 3(a), they are not partitioned into the same clusters in (b), and will never be if the number of clusters is 9.

As the path-based measure is transitive, when it is applied to the refined co-association matrix CM'' of the data set in Fig. 3(a), $\mathbf{x}_1$ and $\mathbf{x}_2$ can be in the same cluster. Note that the similarities depicted in CM'' are not based on Euclidean distance but the refined co-occurrence. Although $\mathbf{x}_1$ and $\mathbf{x}_2$ are far away in Euclidean space, they can be connected by a path with high frequencies of co-occurrence that can be recognized by path-based measure, and this agrees with the cluster assumption in [8, 39]. At the same time, the global assumption in [49] is also met, as $\mathbf{x}_1$ and $\mathbf{x}_2$ are in the same manifold. $\mathbf{x}_3$ and $\mathbf{x}_4$ are relatively close in Euclidean space, but they can only be connected by a path with low frequencies of co-occurrence according to CM''.

Therefore, we can say that the path-based measure can disclose the global structure information of the data set.

After the refined co-association matrix is transformed by the path-based measure, any clustering method that takes a similarity matrix as an input can be selected to produce the final clustering. Since normalized cut [40] is effective and robust, we select it in this study.

The basic idea of normalized cut is to define a cut criterion that takes into account both the total dissimilarity between the different clusters and the total similarity within the clusters. Its main steps are as follows: Construct the similarity graph and compute its weighted adjacent matrix, derive the Graph Laplacian matrix and solve the generalized eigenproblem about it, and use the eigenvector with the second smallest eigenvalue to bipartition the graph.

Combining the normalized cut, the proposed algorithm is described as follows.

```
Algorithm 1 TOME: Two-level-refined cO-association Matrix Ensemble
Input: Data set: $X$, number of clusters: $K$, number of base partitions: $M$,
    mode of producing base partitions: $mode$
Output: Final clustering: $Clus$
 1: $P \leftarrow$ BASE-PARTITION($X, M, mode$)
 2: $CM' \leftarrow$ GEN-POINTLEVEL-CM($P, X$)
 3: $S \leftarrow$ CLUSTER-STABILITY($P, CM'$)
 4: $CM'' \leftarrow$ GEN-TWOLEVEL-CM($P, X, S$)
 5: $W \leftarrow CM''$/ MAX($CM''$)
 6: $W \leftarrow$PATHBASED-SIMILARITY($W$)
 7: $Clus \leftarrow$ NCUT($W, K$)
```

In line 1, function BASE-PARTITION($X, M, mode$) generates the base partitions, where the number of iterations of K-means is set to 4, $mode$ indicates that the number of clusters of a base partition is $\sqrt{N}$ or a value randomly selected from $[2 : \sqrt{N}]$. P is a $N \times M$ matrix and one column is a partition. Function GEN-POINTLEVEL-CM(P, X) in line 2 computes the point-level-refined matrix according to Eq. 11 and 12. Function CLUSTER-STABILITY(P, CM') figures out the normalized stabilities of all base clusters in terms of Eq. 13 and 14. Line 4 computes the tow-level-refined matrix with Eq. 15 and 16. In line 5, CM'' is normalized to [0,1], and MAX(CM'') return the maximum entry of CM''. In line 6, the similarity matrix W is transformed by function

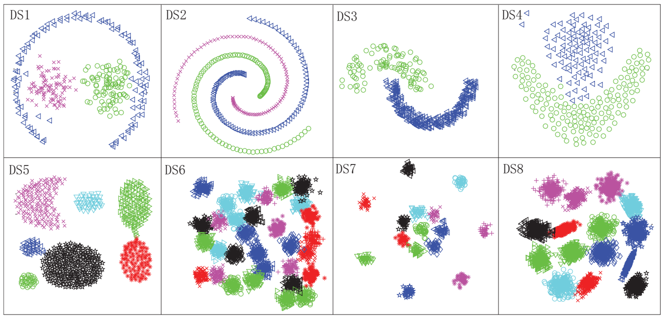


Figure 4: Eight synthetic data sets.

PATHBASED-SIMILARITY(W). Line 7 uses function NCUT(W, K) to produce the final clustering.

2.2.3. *Computational complexity*

The running time of K-means is $O(KIdN)$, where I is the number of iterations. When $K = \sqrt{N}$, it takes $O(IdMN^{1.5})$ to produce M base partitions. For simplicity, we assume that X is divided equally into $\sqrt{N}$ clusters, then the cost of computation of CM' is $O(dMN^{1.5})$. Similarly, the running time of computing cluster stability S and CM'' is also $O(dMN^{1.5})$. Function PATHBASED-SIMILARITY(W) can figure out the path-based similarity from a minimum spanning tree of W [30, 48], and the cost is $O(N^2)$. Therefore, the computational complexity to refine CM and transform it into path-based similarity matrix is $O(MN^{1.5} + N^2)$.

3. Numerical Experiments

3.1. *Experimental data sets*

The proposed method is tested on sixteen data sets, in which eight are synthetic data and the other are real data. The eight synthetic data sets are two dimensional and illustrated in Fig. 4. The detailed descriptions of the data sets are shown in Table 1.

Table 1: The description of the data sets.

Data sets	Classes (K)	Objects (N)	Dimensions (d)	Sources
DS1	3	300	2	[7]
DS2	3	312	2	[7]
DS3	2	373	2	[25]
DS4	2	240	2	[16]
DS5	7	788	2	[17]
DS6	31	3100	2	[43]
DS7	15	600	2	[43]
DS8	15	5000	2	[13]
DS9 (Iris)	3	150	4	UCI [3]
DS10 (Ionosphere)	2	351	34	UCI [3]
DS11 (Wine)	3	178	13	UCI [3]
DS12 (Diabetes)	2	768	8	UCI [3]
DS13 (Segmentation)	7	2130	19	UCI [3]
DS14 (Glass)	6	214	9	UCI [3]
DS15 (WDBC)	2	569	30	UCI [3]
DS16 (WPBC)	2	194	33	UCI [3]

3.2. Algorithms compared

As in the proposed method spectral clustering is used to produce the final clustering, two kinds of clustering algorithms are selected for comparison: spectral clustering and clustering ensemble.

To compare with spectral clustering, we apply normalized cut [40] to the original CM, since the main argument of this study is that the refined and transformed (by path-based measure) CM can disclose more cluster structure information than the original.

To compare with clustering ensemble counterparts, we select two well known approaches: Link-based Clutering Ensemble (LCE) [22] and Strehl's algorithms [41].

LCE discovers some hidden information of the base partitions, and uses it to refine the binary cluster association (BA) matrix. Three refinement schemes are proposed: Weighted Connected-Triple (WCT), Weighted Triple-Quality (WTQ) and Combined Similarity Measure (CSM). Each scheme provides a way to discover the hidden connections between clusters in the same partition, and accordingly BA matrix is refined. After that, K-means, PAM [29] and bi-partite spectral graph clustering [10] are employed to generate the final clustering. Since in the proposed method spectral clustering is employed, for fair comparison, we select WCT, WTQ and CSM combined with spectral clustering in [10].

In [41], three ensemble approaches are presented: Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HGPA) and Meta-CLustering Algorithm (MCLA). CSPA defines a graph $G = (V, E)$, where $V = X$, $e_{ij} \in E$ is an edge with weight $w_{ij} = CM(i, j)$, and employs METIS [27] to cut the graph. HGPA constructs a hypergraph, in which vertices are data points and hyperedges represent clusters, and uses HMETIS [28] to cut the graph. MCLA creates a meta-graph, where a vertex is an indicator vector that indicates if the data points belong to a base cluster or not, and edge weights are proportional to the similarities of vertices measured by binary Jaccard [24].

3.3. Performance measures

We employ two indices to measure the clustering performances: Classification Accuracy (CA) [38] and Adjusted Rand Index (ARI) [21]. Suppose that $\mathcal{S} = \{\mathcal{S}_1, ..., \mathcal{S}_{K_s}\}$ is the ground truth, $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_{K_t}\}$ is a partition to be measured, where K_s and K_t are the number of clusters of $\mathcal{S}$ and $\mathcal{T}$, respectively. CA is defined as:

$$CA(\mathcal{S}, \mathcal{T}) = \frac{1}{N} \sum_{i=1}^{K_t} |\mathcal{T}_i \cap \mathrm{mode}(\mathcal{T}_i, \mathcal{S})| \tag{19}$$

where $\mathrm{mode}(\mathcal{T}_i, \mathcal{S}) = \arg\max_{\mathcal{S}_j \in \mathcal{S}} |\mathcal{T}_i \cap \mathcal{S}_j|$

ARI is defined as:

$$r_0 = \frac{1}{2} \sum_{i=1}^{K_s} \sum_{j=1}^{K_t} |\mathcal{S}_i \cap \mathcal{T}_j|(|\mathcal{S}_i \cap \mathcal{T}_j| - 1)$$

$$r_1 = \frac{1}{2} \sum_{i=1}^{K_s} |\mathcal{S}_i|(|\mathcal{S}_i| - 1)$$

$$r_2 = \frac{1}{2} \sum_{i=1}^{K_t} |\mathcal{T}_i|(|\mathcal{T}_i| - 1)$$

$$r_3 = \frac{2 r_1 r_2}{N(N-1)}$$

$$ARI(\mathcal{S}, \mathcal{T}) = \frac{(r_0 - r_3)}{0.5(r_1 + r_2) - r_3} \tag{20}$$

3.4. Experimental results

The experimental results are shown in Table 2–5. In these tables, TOME stands for the proposed method; CMspec is normalized cut based on the original co-association matrix; WCTspec, WTQspec and CSMspec are the three bipartite graph spectral clustering based ensemble in [22]; CSPA, HGPA and MCLA are the methods proposed in [41].

In all experiments, the parameters are set as follows:

1. Each algorithm runs repeatedly for 50 times, and the corresponding averages are presented.

Table 2: The clustering qualities measured by CA on 16 data sets. The bigger the value, the higher the quality. The number of clusters of the base partitions is $\sqrt{N}$. The highest quality of clustering results in each column is highlighted as the bold item(s). The numbers in the brackets are the corresponding standard deviations. The rightmost column is the number of times each method has the best results.

Method	Data sets																#
	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS9	DS10	DS11	DS12	DS13	DS14	DS15	DS16	
TOME	**0.99** (0.02)	**1.00** (0.00)	**1.00** (0.00)	**0.98** (0.01)	0.88 (0.03)	0.94 (0.02)	**1.00** (0.00)	**0.99** (0.00)	0.95 (0.00)	**0.73** (0.02)	**0.72** (0.01)	**0.65** (0.00)	0.60 (0.03)	0.54 (0.03)	**0.82** (0.04)	**0.76** (0.00)	11
CMspec	0.77 (0.09)	0.36 (0.10)	0.80 (0.00)	0.97 (0.02)	**0.99** (0.05)	**0.98** (0.02)	**1.00** (0.00)	**0.99** (0.00)	0.92 (0.08)	0.70 (0.03)	**0.72** (0.00)	**0.65** (0.00)	**0.70** (0.02)	0.60 (0.02)	**0.82** (0.03)	**0.76** (0.00)	9
WCTspec	0.92 (0.07)	0.40 (0.05)	0.79 (0.10)	0.96 (0.02)	0.91 (0.01)	0.97 (0.00)	**1.00** (0.00)	**0.99** (0.00)	0.94 (0.04)	0.68 (0.04)	**0.72** (0.01)	**0.65** (0.00)	0.69 (0.04)	0.60 (0.01)	0.73 (0.03)	**0.76** (0.00)	5
WTQspec	0.90 (0.07)	0.41 (0.05)	0.89 (0.11)	0.96 (0.01)	0.87 (0.03)	0.97 (0.01)	**1.00** (0.00)	**0.99** (0.00)	0.95 (0.04)	0.71 (0.01)	**0.72** (0.02)	**0.65** (0.00)	0.63 (0.02)	0.60 (0.03)	0.78 (0.02)	**0.76** (0.00)	5
CSMspec	0.88 (0.06)	0.38 (0.03)	0.78 (0.08)	0.95 (0.05)	0.91 (0.00)	0.97 (0.01)	0.99 (0.02)	**0.99** (0.00)	0.91 (0.03)	0.66 (0.03)	**0.72** (0.01)	**0.65** (0.00)	0.68 (0.04)	**0.61** (0.02)	0.71 (0.05)	**0.76** (0.00)	5
CSPA	0.93 (0.05)	0.38 (0.03)	0.76 (0.00)	0.83 (0.01)	0.82 (0.02)	0.97 (0.00)	**1.00** (0.00)	0.97 (0.01)	0.96 (0.05)	0.66 (0.01)	0.71 (0.00)	**0.65** (0.00)	0.69 (0.05)	**0.61** (0.01)	0.76 (0.08)	**0.76** (0.00)	4
HGPA	0.92 (0.01)	0.60 (0.07)	0.74 (0.00)	0.86 (0.05)	0.87 (0.01)	0.93 (0.02)	**1.00** (0.00)	0.89 (0.07)	**0.97** (0.00)	0.68 (0.02)	**0.72** (0.00)	**0.65** (0.00)	0.67 (0.05)	**0.61** (0.02)	**0.82** (0.00)	**0.76** (0.00)	7
MCLA	0.82 (0.11)	0.44 (0.09)	0.77 (0.01)	0.84 (0.02)	0.84 (0.02)	**0.98** (0.00)	**1.00** (0.00)	0.97 (0.01)	0.96 (0.05)	0.71 (0.04)	**0.72** (0.00)	**0.65** (0.00)	0.68 (0.07)	0.58 (0.04)	0.81 (0.04)	**0.76** (0.00)	5

2. For TOME and CMspect, the maximum number of iterations in K-means is 4, and the number of the base partitions is 100.
3. For WCTspec, WTQspec and CSMspec, the maximum number of iterations is 100, and the number of base partitions is 10. Meanwhile, the decay factor parameter DC of these methods is set to 0.9.
4. For CSPA, HGPA and MCLA, the maximum number of iterations is 100, and the number of base partitions is also 100.

In Table 2, the number of clusters in each partitions is fixed to $\sqrt{N}$ and the clustering results are measured by CA. The proposed TOME has the best clustering results on 11 out of total 16 data sets, and the ratio ranks the first. Compared with the other methods, TOME has good performance on DS1–DS4. That indicates TOME can detect the global structure, since each of the four data sets contains some non-compact clusters, which can be described by the global cluster assumption. CMspect, WCTspec, WTQspec and CSMspec also

Table 3: The clustering qualities measured by CA on 16 data sets. The bigger the value, the higher the quality. The number of clusters of the base partitions is randomly selected from $2 : \sqrt{N}$. The highest quality of clustering results in each column is highlighted as the bold item(s). The numbers in the brackets are the corresponding standard deviations. The rightmost column is the number of times each method has the best results.

Method	Data sets																#
	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS9	DS10	DS11	DS12	DS13	DS14	DS15	DS16	
TOME	**0.96** (0.04)	**1.00** (0.00)	**1.00** (0.00)	**0.98** (0.01)	**0.95** (0.05)	0.97 (0.01)	**1.00** (0.00)	**0.99** (0.00)	0.93 (0.05)	**0.73** (0.01)	**0.72** (0.01)	**0.65** (0.00)	0.59 (0.02)	0.53 (0.02)	0.82 (0.04)	**0.76** (0.00)	11
CMspec	0.75 (0.11)	0.36 (0.07)	0.77 (0.00)	0.89 (0.01)	0.93 (0.02)	**0.98** (0.01)	**1.00** (0.00)	**0.99** (0.00)	0.91 (0.06)	0.71 (0.04)	0.71 (0.01)	**0.65** (0.00)	0.58 (0.03)	**0.61** (0.03)	**0.86** (0.04)	**0.76** (0.00)	7
WCTspec	0.80 (0.04)	0.37 (0.01)	0.76 (0.02)	0.93 (0.05)	0.93 (0.02)	0.97 (0.02)	0.99 (0.03)	**0.99** (0.00)	0.90 (0.03)	0.71 (0.02)	**0.72** (0.01)	**0.65** (0.00)	0.66 (0.05)	0.57 (0.03)	0.82 (0.04)	**0.76** (0.00)	4
WTQspec	0.79 (0.02)	0.37 (0.01)	0.76 (0.02)	0.89 (0.05)	0.93 (0.03)	0.97 (0.01)	0.99 (0.03)	**0.99** (0.00)	0.90 (0.05)	0.71 (0.01)	**0.72** (0.01)	**0.65** (0.00)	0.58 (0.04)	0.59 (0.03)	0.84 (0.03)	**0.76** (0.00)	5
CSMspec	0.79 (0.03)	0.37 (0.01)	0.78 (0.03)	0.89 (0.06)	0.92 (0.03)	0.95 (0.03)	0.99 (0.03)	**0.99** (0.02)	0.90 (0.05)	0.70 (0.03)	**0.72** (0.01)	**0.65** (0.00)	0.63 (0.05)	0.56 (0.04)	0.85 (0.06)	**0.76** (0.00)	4
CSPA	0.94 (0.02)	0.42 (0.03)	0.76 (0.00)	0.85 (0.06)	0.83 (0.00)	0.90 (0.01)	0.99 (0.01)	0.96 (0.00)	0.96 (0.05)	0.68 (0.00)	0.71 (0.01)	**0.65** (0.00)	**0.69** (0.05)	**0.61** (0.02)	0.83 (0.03)	**0.76** (0.00)	4
HGPA	0.91 (0.01)	0.48 (0.02)	0.74 (0.00)	0.83 (0.01)	0.82 (0.06)	0.48 (0.02)	0.61 (0.05)	0.76 (0.07)	**0.97** (0.00)	0.64 (0.01)	**0.72** (0.00)	**0.65** (0.00)	0.55 (0.06)	0.54 (0.04)	0.63 (0.00)	**0.76** (0.00)	4
MCLA	0.74 (0.10)	0.42 (0.04)	0.77 (0.01)	0.87 (0.01)	0.88 (0.01)	0.55 (0.03)	0.82 (0.05)	**0.99** (0.00)	0.94 (0.06)	0.71 (0.05)	**0.72** (0.01)	**0.65** (0.00)	0.63 (0.06)	0.57 (0.03)	0.82 (0.10)	**0.76** (0.00)	4

disclose the structure of DS4, but fail on DS1–DS3. Therefore, TOME has more capability of recognizing the global structure than the four methods. For DS5 and DS6, TOME has low quality results, which shows its performance may decrease on compact clusters when the number of the base clusters is fixed. For DS7, DS8, Wine, Diabetes and WPBC, all methods have almost the same performance. Iris is a widely used data set in the literature for testing clustering algorithms. HGPA works well on this data and its CA value is 0.97, while TOME's is 0.95, slightly small.

In Table 3, the number of clusters in base partitions is randomly selected from $[2 : \sqrt{N}]$ and the clustering results are measured by CA. In this setting, TOME has also 11 best clustering results and ranks the first. For data set DS1–DS4, TOME still has better performance than the other compared methods. Compared with Table 2, the performance of TOME on DS5 and DS6, which consist of some compact clusters, is improved. That implies when the number of clusters of a base partition is selected randomly, TOME may have good performance on compact data sets. As the clustering qualities of CMspect are only better than those of TOME on data sets DS6, Glass and WPBC, one can say TOME has better performance than CSpect. HGPA always has the best clustering result on Iris, but its average performance on all 16 data sets is not competitive.

In Table 4, the number of clusters in each partitions is fixed to $\sqrt{N}$ and the clustering results are measured by ARI. In general, for the same clustering result, values produced by ARI are less than those by CA, but they have the same upper bound 1.00. From this table, we can see that TOME wins on 10 data sets and also ranks the first.

In Table 5, the number of clusters in base partitions is randomly selected from $[2 : \sqrt{N}]$ and the clustering results are measured by ARI. TOME wins 9 times, and keeps the first place. Compared with Table 3, the number of times of win for each method decreases, this indicates ARI and CA may have different scales.

In sum, TOME has competitive performance.

Table 4: The clustering qualities measured by *ARI* on 16 data sets. The bigger the value, the higher the quality. The number of clusters of the base partitions is $\sqrt{N}$. The highest quality of clustering results in each column is highlighted as the bold item(s). The numbers in the brackets are the corresponding standard deviations. The rightmost column is the number of times each method has the best results.

Method	Data sets																#
	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS9	DS10	DS11	DS12	DS13	DS14	DS15	DS16	
TOME	**0.96** (0.04)	**1.00** (0.00)	**1.00** (0.00)	**0.93** (0.03)	0.86 (0.04)	0.91 (0.01)	**1.00** (0.00)	**0.99** (0.00)	0.86 (0.01)	**0.21** (0.03)	0.38 (0.02)	**0.01** (0.00)	0.24 (0.04)	**0.26** (0.01)	**0.40** (0.09)	0.01 (0.00)	10
CMspec	0.49 (0.18)	0.00 (0.22)	0.38 (0.00)	0.89 (0.07)	**0.97** (0.05)	**0.95** (0.02)	**1.00** (0.00)	**0.99** (0.00)	0.78 (0.11)	0.16 (0.06)	0.38 (0.00)	0.00 (0.00)	0.53 (0.04)	0.17 (0.02)	**0.40** (0.06)	0.00 (0.02)	5
WCTspec	0.80 (0.07)	0.02 (0.00)	0.23 (0.14)	0.87 (0.15)	0.67 (0.06)	**0.95** (0.02)	**1.00** (0.04)	**0.99** (0.00)	0.83 (0.06)	0.12 (0.03)	**0.40** (0.02)	**0.01** (0.00)	**0.54** (0.06)	0.18 (0.02)	0.21 (0.10)	**0.02** (0.02)	7
WTQspec	0.75 (0.04)	0.02 (0.00)	0.57 (0.12)	0.86 (0.16)	0.59 (0.05)	0.94 (0.02)	**1.00** (0.03)	**0.99** (0.00)	0.86 (0.10)	0.17 (0.02)	0.39 (0.02)	**0.01** (0.01)	0.47 (0.05)	0.19 (0.03)	0.32 (0.07)	**0.02** (0.02)	4
CSMspec	0.70 (0.06)	0.01 (0.00)	0.15 (0.10)	0.82 (0.19)	0.67 (0.05)	0.94 (0.04)	0.99 (0.03)	**0.99** (0.02)	0.78 (0.07)	0.07 (0.07)	0.39 (0.02)	**0.01** (0.03)	0.53 (0.06)	0.18 (0.03)	0.18 (0.15)	0.01 (0.02)	2
CSPA	0.80 (0.10)	0.01 (0.02)	0.27 (0.03)	0.42 (0.03)	0.54 (0.02)	0.94 (0.00)	**1.00** (0.00)	0.93 (0.01)	0.90 (0.10)	0.11 (0.02)	0.39 (0.01)	0.00 (0.00)	0.53 (0.07)	0.18 (0.01)	0.27 (0.16)	0.01 (0.01)	1
HGPA	0.79 (0.01)	0.23 (0.10)	0.17 (0.00)	0.52 (0.14)	0.64 (0.01)	0.88 (0.03)	**1.00** (0.00)	0.83 (0.09)	**0.92** (0.01)	0.13 (0.04)	**0.40** (0.00)	0.00 (0.00)	0.51 (0.07)	0.19 (0.02)	0.42 (0.00)	0.01 (0.01)	3
MCLA	0.59 (0.18)	0.06 (0.08)	0.28 (0.04)	0.47 (0.06)	0.56 (0.03)	**0.95** (0.00)	**1.00** (0.00)	0.94 (0.01)	0.89 (0.11)	0.17 (0.07)	**0.40** (0.00)	0.00 (0.00)	0.52 (0.08)	0.17 (0.03)	0.39 (0.08)	**0.02** (0.00)	4

4. Discussion

4.1. Parameters

We discuss two parameters in the proposed method: the number of iterations of K-means, the number of the base partitions, the number of the base clusters and the width of Parzen window.

4.1.1. Number of iterations of K-means

Many clustering ensemble methods that employ K-means to produce the base partitions concern the number of clusters K for each base partition, but pay no attention to the number of iterations I in K-means. This number controls the degree of convergence of K-means and the diversity of the base partitions. A large I will make a clustering result close to convergence and produce more base clusters with high homogeneity (or quality), but decrease the diversity of the base partitions, or vice versa. To generate a base partition, as the objective

Table 5: The clustering qualities measured by ARI on 16 data sets. The bigger the value, the higher the quality. The number of clusters of the base partitions is randomly selected from $2 : \sqrt{N}$. The highest quality of clustering results in each column is highlighted as the bold item(s). The numbers in the brackets are the corresponding standard deviations. The rightmost column is the number of times each method has the best results.

Method	Data sets																#
	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS9	DS10	DS11	DS12	DS13	DS14	DS15	DS16	
TOME	**0.87** (0.05)	**1.00** (0.00)	**1.00** (0.00)	**0.89** (003.)	**0.93** (0.10)	0.94 (0.02)	**1.00** (0.00)	**0.99** (0.00)	0.82 (0.11)	**0.20** (0.02)	0.37 (0.02)	0.00 (0.00)	0.36 (0.03)	**0.26** (0.00)	0.41 (0.11)	0.01 (0.02)	9
CMspec	0.46 (0.19)	0.00 (0.11)	0.29 (0.00)	0.61 (0.04)	0.75 (0.05)	**0.95** (0.01)	**1.00** (0.00)	**0.99** (0.00)	0.76 (0.12)	0.18 (0.10)	0.37 (0.02)	**0.06** (0.00)	0.35 (0.04)	0.19 (0.03)	**0.52** (0.10)	**0.04** (0.04)	6
WCTspec	0.53 (0.07)	0.00 (0.00)	0.19 (0.14)	0.74 (0.15)	0.74 (0.06)	0.94 (0.02)	0.99 (0.04)	**0.99** (0.00)	0.75 (0.06)	0.18 (0.03)	0.39 (0.02)	0.01 (0.00)	0.49 (0.06)	0.21 (0.02)	0.41 (0.10)	0.03 (0.02)	1
WTQspec	0.50 (0.04)	0.00 (0.00)	0.23 (0.12)	0.62 (0.16)	0.74 (0.05)	0.94 (0.02)	0.99 (0.03)	**0.99** (0.00)	0.78 (0.10)	0.16 (0.02)	0.39 (0.02)	0.01 (0.01)	0.40 (0.05)	0.22 (0.03)	0.46 (0.07)	0.03 (0.02)	1
CSMspec	0.52 (0.06)	0.00 (0.00)	0.30 (0.10)	0.61 (0.19)	0.72 (0.05)	0.91 (0.04)	0.99 (0.03)	**0.99** (0.02)	0.72 (0.07)	0.17 (0.07)	0.38 (0.02)	0.03 (0.03)	0.46 (0.06)	0.21 (0.03)	0.49 (0.12)	0.02 (0.02)	1
CSPA	0.83 (0.05)	0.02 (0.03)	0.26 (0.04)	0.49 (0.01)	0.55 (0.00)	0.82 (0.02)	0.99 (0.01)	0.93 (0.01)	0.90 (0.09)	0.12 (0.00)	0.38 (0.01)	0.00 (0.00)	**0.53** (0.05)	0.18 (0.05)	0.43 (0.08)	0.01 (0.01)	1
HGPA	0.76 (0.02)	0.07 (0.03)	0.17 (0.00)	0.44 (0.03)	0.55 (0.07)	0.40 (0.02)	0.50 (0.06)	0.66 (0.09)	**0.92** (0.01)	0.03 (0.03)	**0.40** (0.00)	0.00 (0.00)	0.31 (0.07)	0.13 (0.03)	0.00 (0.00)	0.01 (0.00)	2
MCLA	0.45 (0.16)	0.02 (0.03)	0.28 (0.01)	0.53 (0.02)	0.65 (0.02)	0.47 (0.02)	0.74 (0.02)	**0.99** (0.08)	0.86 (0.13)	0.18 (0.10)	**0.40** (0.01)	0.01 (0.00)	0.45 (0.07)	0.21 (0.03)	0.44 (0.22)	0.01 (0.00)	2

function of K-means, sum of squared error (SSE), monotonically decreases with each iteration, clusters get more and more compact. As a result, the quality of the base partition increases. At the same time, with their own iteration processes going on, the base partitions approach the convergence closely, and the diversity of the partitions gets small.

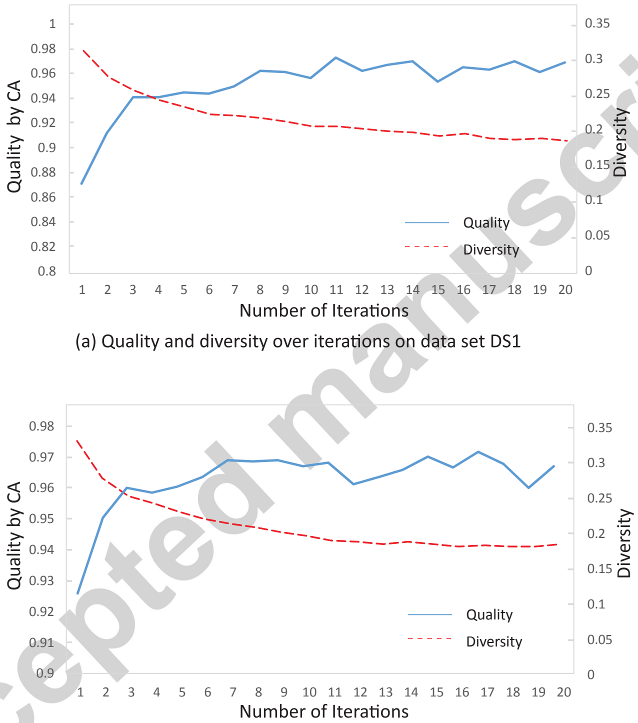
We experimentally demonstrate the above discussion of the impact of I. To measure the quality of a base partition, we still use CA. As for the diversity, it is measured by the average of normalized mutual information [22], and defined as:

$$Div(P_1\cup, ..., \cup P_M) = \frac{2}{M*(M-1)} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} (1 - NMI(P_i, P_j)) \qquad (21)$$

where $Div(P_1\cup, ..., \cup P_M)$ denotes the diversity of M base partitions, $P_1, ..., P_M$, $NMI(P_i, P_j)$ is the normalized mutual information of P_i and P_j [41].

In Fig. 5, the effects of I on the base partition quality and diversity about two data sets are measured. In the experiments, the number of clusters in each base partition is fixed to $\sqrt{N}$; totally 50 base partitions are use to measure the quality and diversity, where the quality is the average of 50 partitions. The data in Fig. 5(a) and (b) are collected from DS1 and Iris, respectively. From the Fig. 5(a), the quality is, on the whole, positively correlated to I and the knee appears roughly between 3 and 4. While the diversity is negatively correlated to I. Fig. 5(b) delivers the similar information to (a). As an ensemble method prefers to high quality and diversity of the base partitions simultaneously, we set I to 4 in the proposed method.

Furthermore, we test the direct impact of I on the final clustering results of all 16 data sets. For each data set, I is changed from 2 to 50 with a step 2. The relationship between Is and the qualities of the results is illustrated in Fig. 6. From this figure, one can see that the number of iterations has a strong impact on some data sets, for example, DS1, DS2, DS5, DS6 and Iris. When the number is from 4 to 6, the clustering results of all 16 data sets will have good



(a) Quality and diversity over iterations on data set DS1

(b) Quality and diversity over iterations on data set Iris

Figure 5: Quality vs. Diversity of base partitions.

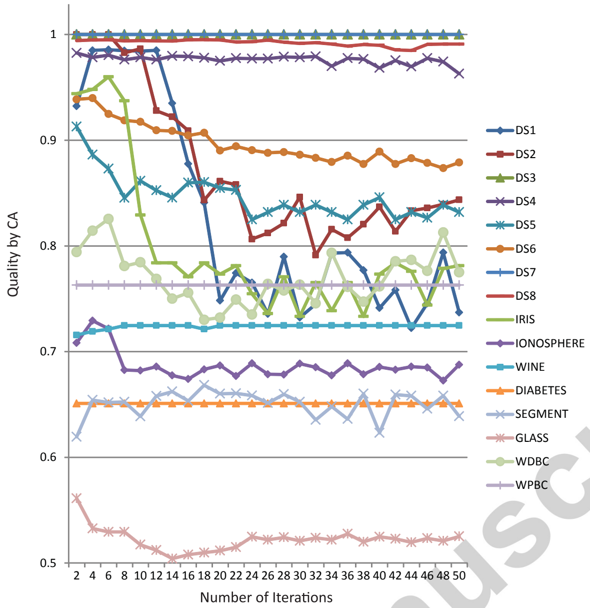


Figure 6: The relationship between the quality of the proposed method and the number of iterations of K-means.

quality. This is approximately accordant to the above discussion of impact of I on the base partition quality and diversity.

4.1.2. Number of base partitions

In general, the larger the number of the base partitions M is, the better the quality of clustering results gets. From the point of view of accuracy, we prefer to a large number of base partitions, but this needs more computational time. Moreover, the ensemble clustering quality is not improved evidently when M increases to a certain extent. To find a relatively suitable number M, we test different Ms from 10 to 200 with a step of 10 on all 16 data sets. The results are illustrated in Fig. 7. When M is set to 30, the clustering results on the majority of data sets are stable. When M is greater than 50, the proposed method will be of high performance for all data sets. For a new data set, we can conservatively set this parameter to 100.

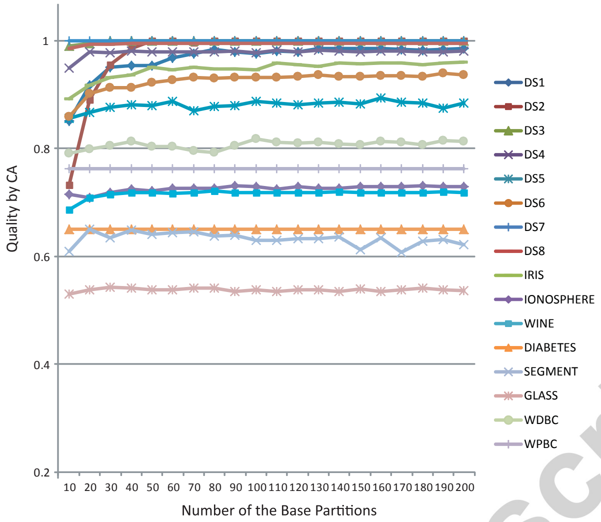


Figure 7: The relationship between the quality of the proposed method and the number of the base partitions.

4.1.3. Number of base clusters

For a base partition, the number of clusters is set to $\sqrt{N}$ or randomly select from $[2 : \sqrt{N}]$. The reason why it is fixed to $\sqrt{N}$ is that a large number of clusters makes base clusters be of more homogeneity. However, a large number has also a negative effect: Some cluster structure information may be lost. Extremely, for example, when the number of clusters in a base partition is N, each data point is a cluster and each cluster is homogenous, but no cluster structure information is conveyed by this partition.

For a clustering, the number of cluster is, as a rule of thumb, not more than $\sqrt{N}$ [6]. That means $\sqrt{N}$ clusters may have more homogeneity and imply the cluster structure well. To verify this rule from the proposed method, we apply the method to the 16 data sets, and for each data set the number of clusters is from k to $10k$ with step k, where $k = \sqrt{N}/10$. The results are shown in Fig. 8. The qualities of some data sets such as DS1, DS2, IRIS and WDBC fluctuate wildly when the number is from k to $7k$, and become stable when the number approaches $10k$. At this moment ($10k$), all the data sets achieve

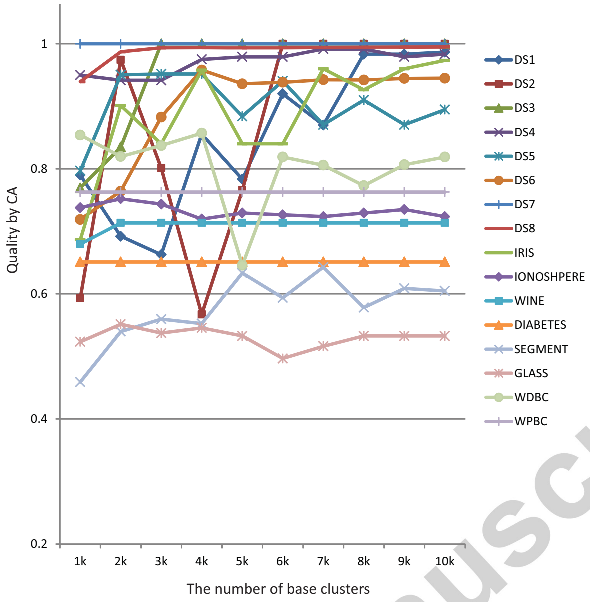


Figure 8: The relationship between the quality of the proposed method and the number of the clusters in a base partition.

clustering results with relatively good quality.

4.1.4. Width of Parzen window

In Section 2.1.1, the co-occurrence probability of a pair of data points is estimated by Parzen window density estimator and can be effected by the window width h. If h is too large, the estimate will suffer from too little resolution; if h is too small, the estimate will suffer from too much statistical variability [9]. In Eq. 10, L has the similar functionality on $p(\mathbf{x}_a, \mathbf{x}_b|C_{ml})$ and is fixed to the maximum distance of two points in a base cluster. To analyse the effect of L on the clustering results, we test L from 0.6 to 2 times of the maximum distance.

From Fig. 9, one can see that L has more effects on DS1, DS2, DS5 and IRIS, where the clustering quality of DS2 decreases after L being greater than 1.2 times. When L is set to the very maximum distance, the 16 data sets will have relative good clustering results. Therefore, L in Eq. 10 is fixed to the maximum distance.

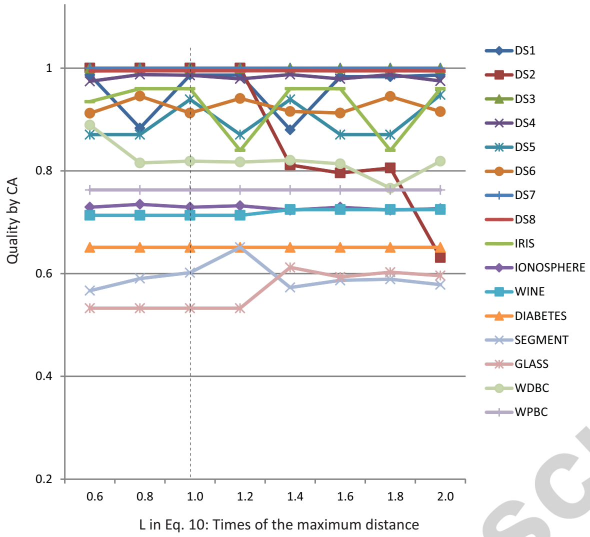


Figure 9: The relationship between the quality and L in Eq. 10.

4.2. Cluster-level refinement

In the cluster-level refinement, cluster stability is employed to measure the contribution of a base cluster to the co-association matrix. Although the intuition of this process is discussed in Section 2.1.2, we demonstrate the contribution experimentally in Fig. 10.

In Fig. 10, the proposed method with the cluster-level refinement produces better clustering results on 7 data sets than that without the refinement, and slightly worse on 2 data sets. Therefore, cluster stability based refinement can

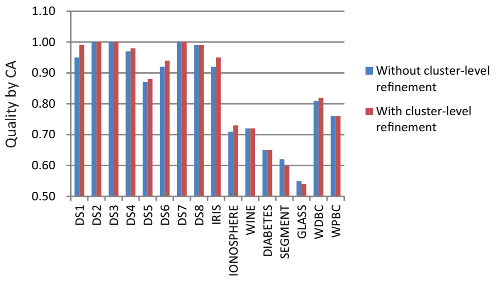


Figure 10: The comparison of qualities between with and without cluster-level refinement.

improve the clustering results.

4.3. Alternatives of refinement

In the proposed method, the co-association matrix is refined in the data point level and the base cluster level. In the data point level, the information of Euclidean distances between two points is incorporated into the corresponding entry of the matrix. That is, two points with a large Euclidean distance will have less evidence of being similar, or vice versa. Since the elements of the original co-association matrix also describe the similarity of pairs of points, an alternative idea may arise: Can we replace the Euclidean distances with the elements in the refinement? If the idea is workable, the co-association matrix can be improved by itself. We experimentally tested it, but results indicated that the improvement is negligible. The possible reason is that no more extra information other than the co-association matrix is presented, while the Euclidean distances can describe some of geometric information of the cluster structure.

In the cluster level of refinement, the stability of a base cluster is defined and used to weight the contribution of the cluster to the final co-association matrix. The stability is computed from the co-association matrix refined in the point level. Another option comes into our mind: Can the stability be defined by the Euclidean distance-based compactness [20]? We also tested this alternative on the 16 data sets, and the results showed that it is almost ineffective, especially towards data sets based on connectedness [20], such as DS2. This is because clusters in this kind of data sets are not compact, and the stability defined with distance-based compactness can not depict the cluster structure.

We also think about if it is helpful to refine the co-association matrix in level of base partition, namely whether weighting a base partition is helpful. There exist some selective ensemble clustering [2, 10] focusing on this point. Actually, selective ensemble methods usually define the contribution of a base partition according to the quality of its clusters. While this information is considered in our second level refinement. Therefore, we ignore the base partition level refinement in this study.

5. Conclusion

The co-association matrix is one of the representations of base partitions in clustering ensemble. In this paper, we refine this matrix from two different levels: the data point level and base cluster level. Traditional ensemble methods usually take each pair of points in a base cluster as one vote of co-occurrence when the matrix is constructed. In the data point level, we weight the co-occurrence by taking into account the Euclidean distance of this pair. It is statistically showed that the distance of two points is negatively correlated to the probability of the two points being in the same cluster. In the level of base cluster, we measure the stability of each base cluster and weight the significance of a base cluster by the stability. After the two levels of refinements, path-base transformation is applied because the matrix is lack of global structure information. Finally, spectral clustering is performed on the matrix to produce the clustering result.

The two refinement processes disclose more local information of cluster structure, while the path-based transformation incorporates some global information into the matrix. Therefore, the final clustering results are improved.

For the future work, we are interested in comparing the intrinsic mechanism of refined co-association matrix with that of the refined binary cluster association matrix in [22]. Meanwhile, since a refined co-association matrix seems like a pairwise similarity transformed by some kernel functions, its applications towards other machine learning problems such as classification, manifold learning would also be interesting.

Acknowledgements

The work was supported by Natural Science Foundation of China (No. 61175054, 61103067), and sponsored by K.C. Wong Magna Fund in Ningbo University.

[1] A. Adán and M. Adán. Consensus strategy for clustering using rc-images. *Pattern Recognition*, 47:402–417, 2014.

[2] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin. To improve the quality of cluster ensembles by selecting a subset of base clusters. *Journal of Experimental & Theoretical Artificial Intelligence*, 26(1):127–150, 2013.

[3] A. Asuncion and D.J. Newman. Uci machine learning repository. *http://www.ics.uci.edu/ mlearn/MLRepository.html*, 2007.

[4] H.G. Ayad and M.S. Kamel. On voting-based consensus of cluster ensembles. *Pattern Recognition*, 43:1943–1953, 2010.

[5] V. Berikov. Weighted ensemble of algorithms for complex data clustering. *Pattern recognition letters*, 38(1):99–106, 2014.

[6] J.C. Bezdek and N.R. Pal. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28 301–315, 1998.

[7] H. Chang and D.Y. Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 41:191–203, 2008.

[8] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. *NIPS*, 2002.

[9] R.O. Duda, P.E. Hart, and D.G. Stork. Pattern Classification. *second ed., New York: John Wiley & Sons*, 2000.

[10] X.Z. Fern and C.E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning*, pages 36–43. ACM, 2004.

[11] B. Fischer and J.M. Buhmann. Bagging for path-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1411–1415, 2003.

[12] L. Franek and X. Jiang. Ensemble clustering by means of clustering embedding in vector spaces. *Pattern Recognition*, 47:833–842, 2014.

[13] P. Fränti and O. Virmajoki. Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5):761–775, 2006.

[14] A.L.N. Fred and A.K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:835–850, 2005.

[15] A.L.N. Fred, A. Lourenço, H. Aidos, S.R. Bulò, N. Rebagliati, M.A.T. Figueiredo, and M. Pelillo. *Learning Similarities from Examples Under the Evidence Accumulation Clustering Paradigm.* Springer, 2013.

[16] L. Fu and E. Medico. Flame, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC bioinformatics*, 8:3, 2007.

[17] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data*, 1(1):1–30, 2007.

[18] M. Girolami and C. He. Probability density estimation from optimally condensed data samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1253–1264, 2003.

[19] B. Hanczar and M. Nadif. Ensemble methods for biclustering tasks. *Pattern Recognition*, 45:3938–3949, 2012.

[20] C. Hennig. Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis*, 52(1):258–271, 2007.

[21] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193C218, 1985.

[22] N. Iam-On, T. Boongoen, S. Garrett, and C. Price. A link-based approach to the cluster ensemble problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:2396–2409, 2011.

[23] N. Iam-On, T. Boongoen, and S. Garrett. Refining Pairwise Similarity Matrix for Cluster Ensemble Problem with Cluster Relations. *Discovery Science, LNAI 5255*, 222–233, 2008.

[24] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[25] A.K. Jain and M.H. Law. Data clustering: A users dilemma. In *Pattern Recognition and Machine Intelligence*, pages 1–10. Springer, 2005.

[26] J. Jia, X. Xiao, B. Liu, and L. Jiao. Bagging-based spectral clustering ensemble selection. *Pattern Recognition Letters*, 32:1456–1467, 2011.

[27] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel Distributed Computing*, 48(1):96–129, 1998.

[28] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Applications in vlsi domain. *IEEE Transactions on Very Large Scale Integration Systems*, 7(1):69–79, 1999.

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

[30] K.H. Kim and S. Choi. Neighbor search with global geometry: A minimax message passing algorithm. In *Proceedings of the 24th International Conference on Machine Learning*, pages 401–408, 2007.

[31] A. Lourenço, A.L.N. Fred, and M. Figueiredo. A generative dyadic aspect model for evidence accumulation clustering. In *Similarity-Based Pattern Recognition*, pages 104–116. Springer, 2011.

[32] A. Lourenço, S.R. Buló, N. Rebagliati, A.L.N. Fred, M.A.T. Figueiredo, and M. Pelillo. Probabilistic consensus clustering using evidence accumulation. *Machine Learning*, 2013.

[33] A. Lourenço, S.R. Buló, N. Rebagliati, A.L.N. Fred, M.A.T. Figueiredo, and M. Pelillo. Consensus clustering using partial evidence accumulation. In *Pattern Recognition and Image Analysis*, pages 69–78. Springer, 2013.

[34] A. Lourenço, S.R. Buló, N. Rebagliati, A.L.N. Fred, and M. Pelillo. Consensus clustering with robust evidence accumulation. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 307–320. Springer, 2013.

[35] S. Mimaroglu and E. Aksehirli. Diclens: Divisive clustering ensemble with automatic cluster number. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(2):408–420, 2012.

[36] S. Mimaroglu and E. Erdil. Combining multiple clusterings using similarity graph. *Pattern Recognition*, 44:694–703, 2011.

[37] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

[38] N. Nguyen and R. Caruana. Consensus clusterings. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 607–612. IEEE, 2007.

[39] M. Seeger. Learning with labeled and unlabeled data. In *Technical report*, The University of Edinburgh, 2007.

[40] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[41] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[42] A. Topchy, A.K. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1866–1881, 2005.

[43] C.J. Veenman, M.J.T. Reinders, and E. Backer. A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1273–1280, 2002.

[44] S. Vega-Pons, J. Correa-Morris, and J Ruiz-Shulcloper. Weighted partition consensus via kernels. *Pattern Recognition*, 43:2712–2724, 2010.

[45] Z. Volkovich, Z. Barzily, and L. Morozensky. A statistical model of cluster stability. *Pattern Recognition*, 41(7):2174–2188, 2008.

[46] X. Wang, C. Yang, and J. Zhou. Combining multiple clusterings using similarity graph. *Pattern Recognition*, 42:668–675, 2009.

[47] Z. Yu, H.S. Wong, J. You, G. Yu, and G. Han. Hybrid cluster ensemble framework based on the random combination of data transformation operators. *Pattern Recognition*, 45:1826–1837, 2012.

[48] C. Zhong, M. Malinen, D. Miao, P. Fränti. fast minimum spanning tree algorithm based on K-means. *Information Sciences*, 295:1–17, 2015.

[49] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, 16, Cambridge, MA, 2004.