An Adaptive Robust Semi-supervised Clustering Framework Using Weighted Consensus of Random k-Means Ensemble

Yongxuan Lai, Songyao He, Zhijie Lin, Fan Yang, Qifeng Zhou, Xiaofang Zhou

Abstract—Semi-supervised cluster ensemble usually introduces a small amount of supervision in the first stage of cluster ensemble, i.e. ensemble generation, by performing many runs of semi-supervised clustering algorithms. However, it is neither efficient in terms of computational complexity, nor flexible in a dynamic learning environment where limited supervision changes over time. In this work we propose a new framework which generates base partitions in an unsupervised manner and attributes different weights to each cluster of the base partitions. The weighting scheme considers both the internal validation measures of clustering and the degrees of satisfaction of pairwise constraints. A weighted co-association matrix based consensus approach is then applied to achieve a final partition. To handle high-dimensional data, we generate base partitions using k-means with both random sampling and random subspace techniques. The new framework retains a high accuracy, and is efficient since it avoids performing semi-supervised clustering in ensemble generation and the complexity of the weighting scheme is independent of the number of instances in a dynamic environment. It is more adaptive than the traditional approach because it does not require rerunning semi-supervised clustering algorithms when the limited supervision changes. Empirical results on twelve datasets demonstrate that it is also more robust to noisy constraints.

Index Terms—Cluster ensemble, semi-supervised clustering, pairwise constraint, k-means.

1 Introduction

Clustering ensemble, also known as consensus clustering, has been emerging as a promising tool for combining results of different runs of homogeneous or heterogeneous clustering algorithms which provide different yet consistent partitions for the same dataset [1], [2], [3]. Using a consensus function, cluster ensemble aims to achieve a consensus among these partitions and outperforms each individual clustering in the ensemble in terms of both accuracy and stability [4].

In this work, we mainly focus on semi-supervised cluster ensemble, which incorporates semi-supervised clustering (also known as constrained clustering) with cluster ensemble together. Previous studies have demonstrated that ensemble approaches can enhance the performance of semi-supervised clustering algorithms [2], [3], [5], [6]. Generally speaking, there are two basic approaches to engage a relatively small amount of supervision, typically pairwise instance-level constraints, in cluster ensemble: (1) a very natural and common way is to bring in supervision in cluster ensemble generation step, that is, generate a large library of clusterings using semi-supervised clustering algorithms with limited supervision [5]. For convenience of discussion, we call it "Constraint-first" approach; (2) perform ensemble generation using clustering algorithms without any supervision, and then introduce limited supervision in the consensus function [6], [7]. We call it "Cluster-first" approach. It is often presumed that ensemble members or base partitions produced by semi-supervised clustering algorithms are more accurate than those generated by clustering algorithms without any supervision, however, there are several disadvantages of the "Constraint-first" approach.

First, it is often inefficient to perform many runs of semi-supervised clusterings on a large dataset. Compared with typical clustering algorithms, semi-supervised clustering algorithms are much more complicated, yielding additional time cost of involving supervision. Further, involving the supervision into the clustering algorithms may raise several serious difficulties [8], [9], [10]. For example, some popular methods such as *COP-Kmeans* are easily over-constrained and may not converge [9].

Second, generating a large library of semi-supervised clusterings is inflexible especially when the supervision changes. The limited supervision is usually provided in the forms of priori knowledge or user feedbacks. With respect to the former, since multiple users probably have multiple views of data, the priori knowledge provided by them can be different. On the other hand, user feedback is likely to increase or even change over time. In these cases, "Constraint-first" approach requires rerunning the semi-supervised clustering algorithms using the updated supervision.

Third, user-provided constraint set may be accompanied by some noise, i.e. noisy constraints. However, semi-

- *Y. Lai and Z. Lin are with the School of Software, and Shenzhen Research Institute, Xiamen University, China. E-mail: laiyx@xmu.edu.cn*
- *S.He, F. Yang and Q. Zhou are with the Department of Automation and Xiamen Key Laboratory of Big Data Intelligent Analysis and Decision-marking, Xiamen University, China. Email: yang@xmu.edu.cn*
- *X. Zhou is with the School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, Australia Email: zxf@itee.uq.edu.au*

Manuscript received January xx, xxxx; revised August xx, xxxx. (Corresponding author: Fan Yang)

supervised clustering algorithms are usually susceptible to noisy constraints. Besides, it has been observed that even the constraints are generated from the ground-truth without any level of noise, they can also have ill effects that decrease the performances of clustering algorithms [8].

Few studies concentrated on the comparison of the "Constraint-first" approach and "Cluster-first" approach, nor did they concern about the potential disadvantages of the "Constraint-first" approach. For the "Cluster-first" approach, the main question is how to incorporate supervision into the consensus function. Our recent study [7] demonstrated that using an ensemble selection method, the "Cluster-first" approach can achieve even higher accuracy than "Constraint-first" approach. The main idea is to select base partitions which agree most with the given constraints from the unsupervised clustering library while retaining their diversity and quality. This preliminary result provides a way to improve the computational efficiency of semi-supervised cluster ensemble. However, the problem formulation of the framework requires defining quality and diversity of base partitions, which needs measure similarities between base partitions. Besides, since the maximization of the objective function for ensemble selection is proven to be NP-hard, we had to adopt a greedy forward selection strategy.

Assuming that a unique ground-truth partition exists, a very recent study suggests that only the base clusterings (or base partitions) close to this ground-truth partition are meaningful to the goal of finding a consensus, no matter how diverse they are [11]. It also indicates that incorporating prior knowledge can help find those base partitions that are close to the ground-truth partition. Another recent study on weighted co-association matrix based consensus clustering proves that careful choices of weighting scheme for the co-association matrix is able to help improve the quality of consensus clustering when using a small library of base partitions [12]. In [13], the authors propose implementing Bayesian Model Averaging (BMA) [14] in model selection for model-based clustering and taking a weighted average of the candidate models. The co-association matrix is viewed as the quantity of interest in BMA, which can measure the probability that each pair of observations belong to the same cluster. Thus, the weighted co-association matrix can represents an average of the posterior distributions under each candidate model weighted by the corresponding posterior model probabilities.

Inspired by the aforementioned studies on cluster ensemble, in the current work we further propose a "Cluster-first" semi-supervised cluster ensemble framework based on a new weighting scheme of base partitions. Suppose that each partition consists of more than one cluster, the new framework adopts *k-means* to generate base partitions and attributes different weights to each cluster of base partitions. The weighting scheme considers both the internal validation measures of clustering and the degrees of satisfaction of pairwise constraints. Within the BMA paradigm, the clusters can be viewed as candidate models, and the internal validation measurement of clusters can represent the marginal likelihood of the cluster, while the consistency with pairwise constraints measure the posterior probability of the cluster. A weighted co-association matrix based consensus approach is then applied to achieve a final partition. We call this method WEighted Consensus of Random *k-means* ensemble (*WECR k-means*). The motivation is to improve the computation efficiency and robustness of semi-supervised cluster ensemble while retaining a high accuracy with limited supervision. The main contributions are as follows:

- We propose a weighted consensus framework for semi-supervised clustering ensemble. The base partitions are generated in an unsupervised learning manner, while the supervision in the form of pairwise constraints is utilized to evaluate and weight each cluster of the base partitions. Compared with the "Constraint-first" approach, the framework avoids performing many runs of time-consuming semi-supervised clustering and is much more computational efficient. Moreover, it is more adaptive to limited supervision as it does not require rerunning a large library of unsupervised or semi-supervised clusterings when the supervision changes.
- To handle the high-dimensional data, we generate base partitions using *k-means* with both random sampling and random subspace techniques. *K-means* is linear in the number of instances, dimensions and clusters, and thus is scalable to large datasets. Using both random sampling and random subspace techniques not only guarantee the diversity of base partitions, but also ease the curse of dimensionality by clustering in a low-dimensional feature space.
- With different level of supervision as well as the presence of noise provided by pairwise constraints, we empirically demonstrate that the weighted consensus of many unsupervised clusterings can outperform the consensus of many semi-supervised clusterings on the qualities of final consensus in three aspects: (1) among all the eight compared algorithms, it achieves the best average qualities over twelve datasets in each case of different sizes of constraint set; (2) it shows less fluctuations with the change of constraints; and (3) it is the most robust with respect to different levels of noisy constraints while keeping high qualities.

The rest of the paper is organized as follows. Section 2 surveys related work of unsupervised and semi-supervised clustering ensemble. Section 3 describes the proposed framework. Section 4 presents the experiments under different settings of constraints, and demonstrates the merits of the new framework. Section 5 summarizes the work.

2 Related Works

In this section, we give a brief review of existing work on cluster ensemble and semi-supervised cluster ensemble.

2.1 Cluster Ensemble

As a state-of-the-art approach, ensemble learning, which generates multiple models and then combines them to a single consensus solution, not only has attracted great attentions in machine learning community, but has been widely used in real applications for its good properties of

robustness and accuracy. Generally, clustering ensemble approaches consist of two main steps, i.e. ensemble generation and finding a consensus. To obtain a good ensemble, diverse partition solutions need to be generated. Previous studies have provided many different ways of ensemble generation, such as performing heterogeneous clustering algorithms [4], changing initialization or other parameters for homogeneous clustering algorithms [15], [16], manipulating the original data by projection onto different subspaces [16], [17] and partition into different subsets [18], [19]. To combine the base partitions to get a final solution, various approaches are proposed, e.g. relabeling and voting [18], [20], co-association matrix [15], [21], graph partitioning [4], cumulative voting [22], mixture models [23], Bayesian cluster ensemble [24], and so on.

Generally, cluster ensemble or consensus clustering look for a consensus solution which is the closest to all the base partitions, that is, an 'average' solution which agrees the most with all the given base partitions. Under the assumption that a base partition λ_i is a noisy version of the ground truth partition λ^* of the data set $X = \{\mathbf{x}_1, \mathbf{x}_2 ... \mathbf{x}_N\}$, A. P. Topchy, M. H. C. Law, A. K. Jain, and A. L. N. Fred [25] provided a rigorous proof and justification of why a clustering ensemble can converge to a high quality consensus solution from two perspectives. First, they proved that voting with re-labeling method can detect the true partition λ^* with a probability that approaches to 1 as the size of ensemble increases. Second, when a consensus function is regarded as finding a mean partition of base partitions, the probability of failing to find the true partition also drops exponentially as the size of ensemble increases.

Note that those justifications of asymptotic convergence properties are based on a sufficiently large ensemble and base partitions are treated equally. However, the ensemble size is always finite and hence some base partitions which are far away from the ground-truth should be suppressed in the consensus function. Attributing different weights to base partitions or performing ensemble selection based on a certain criterion are then regarded as reasonable. Among the criteria, the tradeoff between diversity and quality of base partitions, also known as bias-variance tradeoff, is directly extended from ensemble classification. Diversity of the ensemble members, i.e. the disagreement of the base partitions on the partitioning of the data, is certainly a prerequisite for ensemble clustering. If ensemble members resemble each other, the consensus is merely a replication. Enhancing diversity can reduce the redundancy. For this reason, diversity as well as quality has been regarded as a key factor in ensemble clustering, and is widely used as an important metric to improve the consensus clustering by performing ensemble selection with an objective function consisting of quality and diversity [26], [27], [28], [29], [30]. And some other studies assigned varying weights to base partitions in the ensemble according to their mutual similarities [31].

These studies often focused on reducing the redundancy while retaining quality of the base partitions, from the perspective of tradeoff between diversity and quality. For example, S. Hadjitodorov, L. Kuncheva, and L. Todorova proposed different evaluation metrics of diversity, then ranked all the solutions based on their diversity, and finally chose the base partitions with median diversity [28] . Fern and Lin [26] first proposed an optimization objective consisted of diversity and quality simultaneously and selected only a subset of partitions from a library of clustering solutions to maximize the objective function. However, how to balance the role of quality and diversity in the objective function of consensus clustering still remains unknown. Besides, some empirical studies on the role of diversity contradict each other [29].

Recently, Brijnesh J. Jain's theory provided another perspective for improving the quality of cluster ensemble, which may help us step out of the paradigm of trade-off between diversity and quality [11]. In the paradigm of collective wisdom, Condorcet's Jury Theorem has been investigated as a theoretical justification of ensemble approaches for classification tasks [32]. Brijnesh J. Jain extended Condorcet's Jury Theorem to the mean partition approach in order to explain the underlying mechanism of cluster ensemble, and provided a different opinion on diversity-quality tradeoff [11]. The theory claimed that what really matters is that the mean partition is a consistent estimator of the ground-truth. To get close to the ground-truth, diversity in the ensemble members is neither necessary nor sufficient, while limiting the diversity is necessary but not sufficient. Note that Brijnesh J. Jain' theorem proposed to include not only the assumptions of Condorcet's theorem, but two additional assumptions, i.e. the existence of a unique ground-truth partition, and the existence of a sufficiently small ball containing the ground-truth partition and sample partition. Although these assumptions may not hold in some cases, it still provides a new insight into clustering ensemble, that is, only the base partitions close to ground-truth partitions are meaningful no matter how diverse they are. This study gives us inspiration that *additional prior knowledge can be used to help find the region nearby the ground-truth in the partition space.*

In another recent study [12], the authors aim to theoretically investigate the convergence properties of cluster ensemble approaches based on weighted co-association matrices with increasing ensemble size. The elements in the co-association matrix are assigned with weights dependent on an evaluation function, which measures the quality of base partitions with both local and global levels of information. An important conclusion is that under some natural assumptions the misclassification probability converges to zero when the ensemble size goes to infinity, no matter what evaluation function is applied. However, if the goal is to achieve high accuracy with a small ensemble, careful choices of weighting scheme that are determined by the evaluation function is crucially important. Hence, it also indicates that *prior knowledge can be used in the design of an evaluation function for attributing weights to the co-association matrix.*

The aforementioned studies on cluster ensemble rarely involved utilizing limited supervision to help improve the quality of the final consensus. The design of ensemble selection strategies or weight function only consider the internal quality and mutual similarities of base partitions. In [33], a framework of generalized weighted clustering aggregation with constraints is proposed, while the information of pairwise constraints is introduced as the constraints

of an optimization objective defined based on Bregman divergence between base partitions. And in this work, we also focus on semi-supervised cluster ensemble and design a framework of using pairwise constraints.

2.2 Semi-supervised Cluster Ensemble

Semi-supervised clustering is popular in many fields where limited prior knowledge is available [34], [35], [36], e.g. *COP-Kmeans* [37], *Seeded-Kmeans*, *Constrained-Kmeans* [38], and *Exhaustive and Efficient Constraint Propagation*(E^2CP) [39]. However, few studies concentrated on semi-supervised clustering ensemble. Recent studies demonstrated that ensemble approach can also enhance the performance of semi-supervised clustering algorithms. Yu et. al. [5] first propose an incremental semi-supervised clustering ensemble framework (ISSCE) for high dimensional data clustering, which combines the random subspace technique, the constraint propagation approach, and the normalized cut algorithm. Also the authors proposed an adaptive semi-supervised clustering ensemble framework [2]. Most of these approaches can be classified as "Constraint-first" approach, which adopt semi-supervised clustering algorithms to produce many base partitions and then combine them. As mentioned in Section 1, this framework is computational inefficient and inflexible, and may also raise many other difficulties such as being susceptible to noisy constraints. The difficulties of using constrained clustering were discussed in detail in previous studies [8], [9], [10]. However, few studies are reported on the computational inefficiency and other potential problems of "Constraint-first" approach. In our previous study [7], we propose a "Cluster-first" approach which utilizes the pairwise constraints to help select a subset of base partitions after ensemble generation instead of engaging the constraints in the generation of base partitions. The objective function of ensemble selection is defined as the weighted sum of diversity and quality of base partitions as well as their consistencies with the pairwise constraints. In this way, a subset of high-quality yet diverse base partitions can be found. However, the problem formulation of the framework requires defining quality and diversity of base partitions, which need measure similarities between base partitions. Further, the maximization of the objective function for ensemble selection is proven to be NP-hard.

In this work, we propose a "Cluster-first" semi-supervised cluster ensemble framework. Instead of using constraints for ensemble selection, it evaluates and attributes weights to each cluster of the base partitions directly with both internal validation measure of clusterings and pairwise constraints. Accordingly, the co-association matrix is refined and a consensus solution is achieved using graph partitioning method. Hence it does not require either measuring similarities between base partitions or solving a NP-hard problem.

3 Weighted Consensus of Random K-means Ensemble

Assuming that a data matrix of dimension N-by-p, $X = \{\mathbf{x}_1, \mathbf{x}_2 ... \mathbf{x}_N\}$, is given as a set of N instances, with $\mathbf{x}_i$ denoting a p-dimensional vector. Cluster ensemble looks for a consensus partition λ based on a set of l partitions $\lambda_1, ..., \lambda_l$ of the dataset X. The notations used throughout this paper are summarized in Table 1.

TABLE 1
Table of notations

Notation	Meaning
$\mathbf{x}_i$	The i-th instance, $\mathbf{x}_i \in \mathbb{R}^p$
$\mathbf{X}$	The data matrix, $\mathbf{X} \in \mathbb{R}^{N \times p}$
λ_i	The i-th partition
π_{ij}	The j-th cluster in λ_i
y_m^i	The class label of instance m determined by λ_i
$\mathbf{y^i}$	The label vector of all instances determined by λ_i
$\pi_{i \leftarrow t}$	The cluster that instance t belongs to in λ_i
$\mathcal{N}(\pi_{i \leftarrow k})$	The nearest cluster of $\pi_{i \leftarrow t}$ in λ_i
(m, n)	The pairwise constraint between instances m and n
$\mathcal{M}, \mathcal{C}$	The must/cannot-link constraints set
$\mathcal{M}_{ij}, \mathcal{C}_{ij}$	The associated constraints set of cluster π_{ij}
$\theta_{(m,n,i)}$	The consistency with constraint (m, n) of λ_i
μ_i	The clustering-level weight of λ_i
ν_{ij}	The cluster-level weight of π_{ij}
φ_{ij}	The combined weight of π_{ij}
γ	The scaling factor between the two level consistencies
ρ_{ij}	The size of associated constraint set of π_{ij}
σ_{ij}	The Silhouette Coefficient of π_{ij}
ω_{ij}	The final weight of π_{ij}
l	The size of the ensemble
$\mathbf{H}$	The complete binary membership matrix
$\mathbf{H}^{(i)}$	The binary membership matrix of λ_i
$\mathbf{W}$	The diagonal weight matrix
$\mathbf{S}$	The co-association matrix

3.1 Co-association Matrix Based Consensus Approach

WECR k-means adopts the co-association matrix based consensus approach, which is widely used for combining different partitions [15]. This $N \times N$ matrix defines the similarities between N instances using the frequency of a pair of instances partitioned into the same cluster. Considering a hard clustering method, e.g. *k-means*, a binary membership matrix is constructed first in order to derive the co-association matrix.

For a base partition λ_i in a cluster ensemble of size l, a *binary membership matrix* $\mathbf{H}^{(i)}$ is a $N \times k_i$ matrix, with k_i denoting the number of clusters in λ_i, each row of $\mathbf{H}^{(i)}$ standing for a instance and each column for a cluster in λ_i. For λ_i, the corresponding element in the matrix $\mathbf{H}^{(i)}$ is set to 1 when an instance belongs to a cluster π_{ij}, otherwise it is set to 0, as defined by Eq.1.

$$\mathbf{H}^{(i)}(t, j) = \begin{cases} 1 & \text{if } \mathbf{x}_t \in \pi_{ij} \\ 0 & \text{if } \mathbf{x}_t \notin \pi_{ij} \end{cases} \quad (1)$$

A complete binary membership matrix $\mathbf{H}$ of the ensemble can be built by combining all the binary membership matrices of the base partitions in the ensemble horizontally, i.e. $(\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, ..., \mathbf{H}^{(l)})$. Hence the co-association matrix $\mathbf{S}$ containing pairwise similarities between instances can be simply derived by matrix multiplication, where l is the number of base partitions:

$$\mathbf{S} = \frac{1}{l}\mathbf{H}\mathbf{H}^\top \quad (2)$$

However, using the information carried by the original co-association matrix alone is not enough to produce a

good consensus partition since all base partitions are treated equally during the construction of the matrix. It will be refined with additional information in *WECR k-means*.

3.2 Bayesian Model Averaging for Clustering

BMA is a classical method of model averaging. Assuming that data X come from one of a set of possible models $F_1, F_2, ..., F_k$, the BMA approach estimates some quantity of interest Δ under each model F_i and then averages the estimates according to the posterior probability $P(F_i|X)$ for each model [14], i.e.,

$$P(\Delta|X) = \sum_{i=1}^{k} P(\Delta|F_i, X)P(F_i|X) \tag{3}$$

Within the paradigm of BMA for model based clustering [13], it is assumed that the data come from different subpopulations, and within each subpopulation the data can be modeled using a single component model. Hence a cluster π can be viewed as a candidate model separately. Given a set of base partitions, the co-association matrix $\mathbf{S}$ (Eq.2) which represents the probability that each pair of instances belong to the same cluster, has the same meaning in each model and is invariant to the labeling of clusters, so it can be used as the quantity of interest in the inference. Therefore, for each pair of instances x_i and x_j, BMA estimates the probability,

$$\begin{aligned} &P(x_i, x_j \ in \ the \ same \ cluster|X) \\ &= \sum_{i=1}^{k} P(x_i, x_j \ in \ the \ same \ cluster|\pi_i, X)P(\pi_i|X) \end{aligned} \tag{4}$$

Note that if all the clusters are treated equally, i.e. with the same $P(\pi_i|X)$, then Eq.4 can be viewed as calculating the element in the original co-association matrix. The term $P(\pi_i|X)$ can assign weight to the cluster π_i by its posterior probability. Then we can get an average of the posterior distributions under each cluster weighted by the corresponding posterior model probabilities.

3.3 Overview of The Framework

The proposed framework is based on the assumption that the quality of base partitions in the ensemble is different from each other. We further assume that even the clusters in the same partition have different quality. They should have different contributions to the final consensus. Under these assumptions, we extend BMA for clustering to semi-supervised ensemble clustering using Eq.4. The key is the posterior probability of cluster π_i, which can be given by

$$P(\pi_i|X) = \frac{P(X|\pi_i)P(\pi_i)}{\sum_{i=1}^{k} P(X|\pi_i)P(\pi_i)} \tag{5}$$

where $P(X|\pi_i)$is the marginal likelihood of π_i, and $P(\pi_i)$ is the prior probability. Since both terms are difficult to estimate, an approximation is required. We propose to adopt the internal validation measurement to approximate the likelihood, and use the external evaluation to approximate the prior probability. Then the weight can be given by the multiplication of the internal validation measurement and the external evaluation, since the denominator in Eq.5 can be seen as a scaling factor. Note that, the cluster π_i can be generated by semi-supervised or unsupervised clustering algorithms. However, in our framework, we propose to employ a "Cluster-first" approach to avoid running semi-supervised algorithms many times.

Intuitively, *WECR k-means* directly measures the closeness between base partitions and the ground-truth partition λ^* by using both internal and external evaluations of quality of each cluster in base partitions, and then attributes different weights to each cluster according to the closeness. Consistencies with pairwise constraints are utilized as the external evaluation criterion. A pairwise constraint is given in the form of a pairwise relation (m, n) between instances m and n, and could be regarded as priori knowledge or user feedback. A *must-link* constraint indicates a must-link relation of two instances that they should be associated with the same cluster. A *cannot-link* constraint indicates a cannot-link relation of two instances that they should not be associated with the same cluster.

The framework of *WECR k-means* is described in Fig.1. It mainly consists of three steps: (1) generate the cluster ensemble using unsupervised clustering, namely, *k-means* with random sampling and random subspace techniques; (2) attribute weights to clusters, using both internal validation measure and external constraints, which will be utilized to make refinements on the original co-association matrix; (3) partition the co-association matrix using a graph partition approach, e.g., Cluster-based Similarity Partitioning (CSPA) [4].

Compared with the "Constraint-first" semi-supervised cluster ensemble, *WECR k-means* circumvents the generation of a large library of semi-supervised clusterings. Compared with ensemble selection approaches [26], the main merits of *WECR k-means* are two fold: (1) The method avoids defining similarities between base partitions, which is usually difficult and remains an open question; (2) The circumvention of solving an NP-hard optimization problem makes it more efficient.

3.4 Generation of Random K-means Ensemble

In ensemble generation, *k-means* is adopted as its time complexity is linear in the number of instances, dimensions, and clusters. The *FS-RS-NC* method proposed in our previous work [27] is used to generate diverse base partitions. Specifically, before each run of *k-means* we perform random sampling on the data and random subspace technique on the feature vectors. *K-means* is then conducted on the new dataset. To capture local distribution of data, a relatively large value of parameter k, number of clusters, is chosen to produce many clusters in each partition.

We call it *Random K-means Ensemble*. The intention is two fold. First, diverse base partitions would provide different views on the original data, help create a more robust consensus, and enhance the probability of finding the ground-truth solution. Second, it is able to ease the curse of dimensionality by performing clustering in a low-dimensional feature space.

3.5 Adaptive Weighting of Base Partitions

As discussed in Section 2.2, a diverse library of base partition does not naturally lead to high quality solutions. Gen-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2952596, IEEE Transactions on Knowledge and Data Engineering

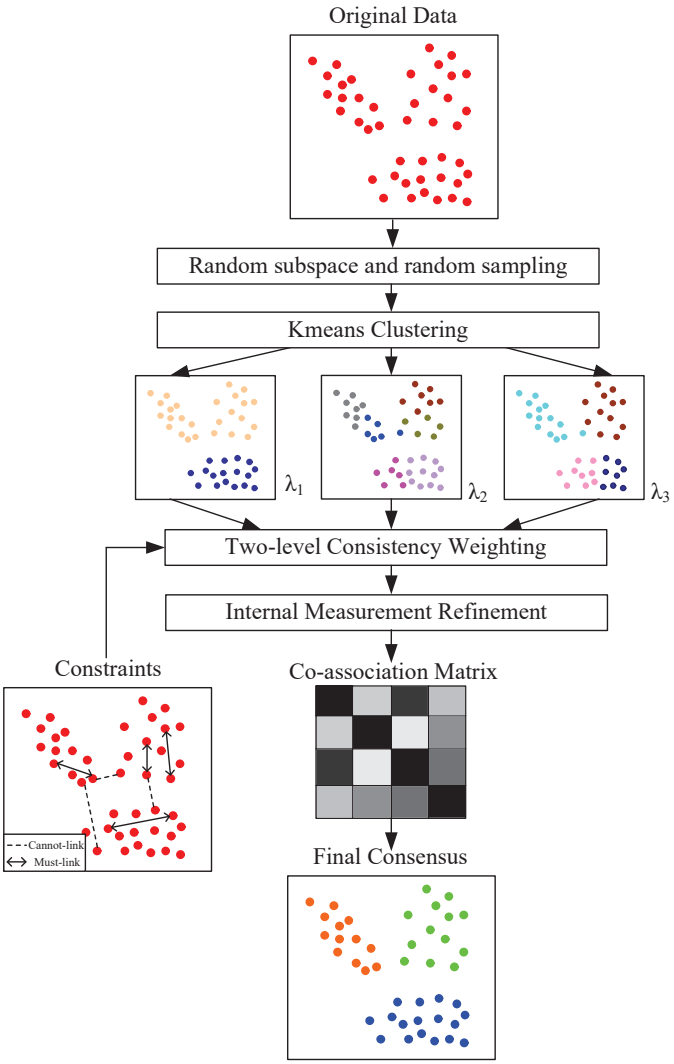


Fig. 1. The Framework of Weighted Consensus of Random K-means Ensemble

erally speaking, some partitions or clusters are more close to the ground truth partition, so different base partitions or clusters have different contributions to the final solution, or more directly, have different influences on learning pairwise similarities between instances.

In this work, we propose an adaptive weighting scheme for each cluster of base partitions which is adaptive to the limited supervision. The weighting scheme consists of three ingredients: (1) *clustering-level consistency* and (2) *cluster-level consistency*, which measure the consistency of a base partition with the limited supervision at the global-level and local-level respectively; (3) internal evaluation of clustering quality which measures the intra-cluster and inter-cluster similarities of a base partition simultaneously.

Specifically, pairwise constraints are used to evaluate the consistency of base partitions. We assume that under the same conditions, the more constraints are satisfied in a base partition, the closer it is to the ground-truth partition. Hence it has a larger prior probability. Fig.2 describes the *clustering-level consistency* with constraints of two partitions. The partition on the left clusters the points into four clusters, yet the must-link constraint (a, b) is violated. The partition on the right consists of three clusters in which the cannot-link constraint (c, d) is violated.

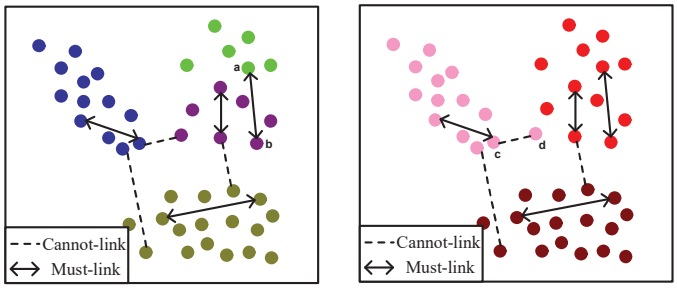


Fig. 2. Illustration of clustering-level consistency

Definition 1. The clustering-level consistency measures the degree of satisfaction of a clustering with respect to constraints globally. Assume that $\mathcal{M}$ and $\mathcal{C}$ are *must-link* and *cannot-link* constraints sets respectively. The *Clustering-level consistency* μ_i of the i-th base partition λ_i can be defined by the average satisfaction of the constraints set, as shown in Eq.6:

$$\mu_i = \frac{\sum_{(m,n)\in\mathcal{M}} \theta^{=}_{(m,n,i)} + \sum_{(m,n)\in\mathcal{C}} \theta^{\neq}_{(m,n,i)}}{|\mathcal{M}| + |\mathcal{C}|} \tag{6}$$

Here, $\theta^{=}_{(m,n,i)}$ or $\theta^{\neq}_{(m,n,i)}$ denotes whether a must-link or cannot-link constraint (m, n) is satisfied in λ_i. As defined in Eq.7, for a must-link constraint, it would be assigned 1 if the instances m and n are within the same cluster (denoted by y) in partition λ_i, otherwise it would be assigned 0. The situation is just the opposite for a cannot-link constraint.

$$\theta^{=}_{(m,n,i)} = \begin{cases} 1 & \text{if } y^i_m = y^i_n \\ 0 & \text{if } y^i_m \neq y^i_n \end{cases} \quad \theta^{\neq}_{(m,n,i)} = \begin{cases} 1 & \text{if } y^i_m \neq y^i_n \\ 0 & \text{if } y^i_m = y^i_n \end{cases} \tag{7}$$

The *clustering-level consistency* evaluates the partition quality at the global level, and assigns the same weight to all the clusters in that partition. However, note that even a good partition of excellent overall performance is likely to contain some bad clusters, as illustrated in Fig.3: π_1 and π_4 are good clusters with all the associated constraints satisfied, while π_2 and π_3 are not so good: the must-link constraint (e, f) is not satisfied in π_2 while the cannot-link constraint (f, g) and the must-link constraint (e, f) are violated in π_3. On the other hand, a low-quality partition might also contain one or more good clusters. In this perspective the consistency with constraints should be extended to the *cluster-level* to get a more local and fine-grained evaluation.

Since every cluster contains only a part of instances, the calculation of the *cluster-level consistency* of a cluster should only take into consideration the constraints associated with that cluster, which are described below.

Definition 2. The associated constraint set of a cluster is a set of constraints with at least one instance of a pair included in that cluster. The *associated constraint sets* $\mathcal{M}_{ij}$ and $\mathcal{C}_{ij}$ for the j-th cluster in i-th clustering π_{ij} are given in Eq.8 and Eq.9 respectively.

$$\mathcal{M}_{ij} = \{(m,n)|(m,n) \in \mathcal{M}, m \in \pi_{ij} \vee n \in \pi_{ij}\} \tag{8}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2952596, IEEE Transactions on Knowledge and Data Engineering

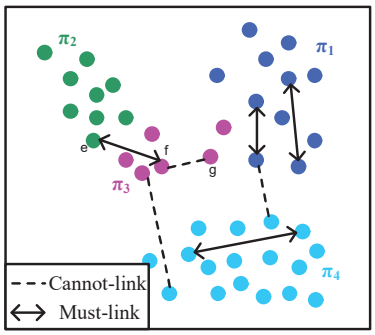


Fig. 3. Illustration of cluster-level consistency

$$\mathcal{C}_{ij} = \{(m,n)|(m,n) \in \mathcal{C}, m \in \pi_{ij} \vee n \in \pi_{ij}\} \tag{9}$$

Based on the definition of *associated constraint set*, the *cluster-level consistency* of a cluster is defined below.

Definition 3. The cluster-level consistency measures the degree of satisfaction of a cluster with respect to its *associated constraint set*. As defined by Eq.10, it is in the same form of the *clustering-level consistency*. For those clusters with no constraints associated, we set the *cluster-level consistency* to 1.

$$\nu_{ij} = \begin{cases} \frac{\sum_{(m,n)\in\mathcal{M}_{ij}} \theta^{=}_{(m,n,i)} + \sum_{(m,n)\in\mathcal{C}_{ij}} \theta^{\neq}_{(m,n,i)}}{|\mathcal{M}_{ij}|+|\mathcal{C}_{ij}|} \\ \text{if } \mathcal{M}_{ij} \cup \mathcal{C}_{ij} \neq \emptyset \\ 1 \qquad \text{otherwise} \end{cases} \tag{10}$$

Weighting based on two-level consistencies

The weighting scheme of base partitions first combines the *clustering-level consistency* (global-level) and *cluster-level consistency* (local-level). Here we use a nonlinear combination function , as the tradeoff between global-level and local-level consistencies depends on the complicated relations between the size of both the constraints set and associated constraint set.

First, note that the size of *associated constraints set* $\mathcal{M}_{ij}$ and $\mathcal{C}_{ij}$ may vary significantly among all clusters in a clustering λ_i. Intuitively, when the size of *associated constraints set* is large, we have more confidence in the *cluster-level consistency weighting* of a cluster π_{ij}. On the contrary, it is less reliable to use *cluster-level consistency* to measure the quality of a cluster alone when the size of *associated constraints set* of that cluster is small. In such case, *clustering-level consistency*, which measures the average consistency of all clusters, should be used as an auxiliary weight to depict the quality of that cluster.

In this regard, we must first consider how to judge whether the size of the associated constraint set of a cluster is sufficient or not. We introduce the concept of *expected size of associated constraints set* $E(\rho_{ij})$ for cluster π_{ij}. Assuming that constraints are distributed uniformly among N instances, the *expected size of associated constraints set* should be proportionate to the size of each cluster, as shown in Eq.11, where $\rho_{ij} = |\mathcal{M}_{ij}| + |\mathcal{C}_{ij}|$, denoting the size of *associated constraints set* of π_{ij}.

$$E(\rho_{ij}) = \frac{|\pi_{ij}|}{N} \times (|\mathcal{M}| + |\mathcal{C}|) \tag{11}$$

If $\rho_{ij} > E(\rho_{ij})$, we have full confidence in using *cluster-level consistency* of cluster π_{ij} to evaluate its quality. Otherwise, *clustering-level* consistency should be used together.

Second, the total number of constraints should be taken into consideration. Note that the *expected size of associated constraints set* depends in part on the size of constraint set, so that it can be very small if the total number of constraints is small. In such case, *cluster-level consistency* is still not reliable in evaluating the quality of a cluster, even if the *expected size of associated constraints set* is achieved in that cluster.

Taking the above considerations into account, a nonlinear combination function $\varphi(\pi_{ij})$, as shown in Eq.12, is proposed. Here the weight function $g_\gamma(\pi_{ij})$ is actually a function of con_{ij} with a hyperparameter γ (Eq.13). Fig.4 describes the graphs of function g_γ with different parameters $\gamma \in [0,1]$.

$$\varphi(\pi_{ij}) = (1 - g_\gamma(\pi_{ij}))\mu_i + g_\gamma(\pi_{ij})\nu_{ij} \tag{12}$$

$$g_\gamma(\pi_{ij}) = \begin{cases} \frac{\rho_{ij}}{E(\rho_{ij})} \times \gamma & 0 < \rho_{ij} \leq E(\rho_{ij}) \\ \frac{\rho_{ij} - E(\rho_{ij})}{|\mathcal{M}|+|\mathcal{C}|-E(\rho_{ij})} \times (1-\gamma) + \gamma & \text{otherwise} \end{cases} \tag{13}$$

First, we have $g_\gamma(\pi_{ij}) \to 0$ when $\rho_{ij} \to 0$, which means we tend to use *clustering-level consistency* μ_i to measure the quality of π_{ij} when there are not enough associated constraints. Similarly, we have $g_\gamma(\pi_{ij}) \to 1$ when $\rho_{ij} \to |\mathcal{M}| + |\mathcal{C}|$ since a larger *associated constraints set* makes *cluster-level consistency* more trustable. Second, it is designed as a piecewise linear function because the *expected size of associated constraints set* plays a role in determining the function value, as shown in Fig.4. It can be observed that, when $\rho_{ij} < E(\rho_{ij})$, the influence of *cluster-level consistency* on the weight of π_{ij} is suppressed. When $\rho_{ij} > E(\rho_{ij})$, the contribution of *cluster-level consistency* to the weight grows more quickly to 1 as the size of associated constraints set increases.

Taking the total number of constraints into account, the scaling factor $\gamma \in [0,1]$ is introduced to further control to what extent *cluster-level consistency* contribute to the final weight of cluster π_{ij}. Generally speaking, it can be adjusted according to the total number of given constraints. It should be set to a large value close to 1 if we have a large number of constraints. As shown in Fig.4:(c), it is the ideal case when there is sufficient number of constraints. And in this case, if $\rho_{ij} > E(\rho_{ij})$, we have full confidence in using *cluster-level consistency* to evaluate the quality of π_{ij}, as mentioned above. On the contrary, when the number of constraints is small, it should be set to a small value, as shown in Fig.4:(b). In such case, if $\rho_{ij} < E(\rho_{ij})$, it even gets rid of the effect of the *cluster-level consistency*.

Refining with Internal Validation Measurement

Internal validation measurements are widely used to quantify the quality of clustering solutions by considering both compactness and separation, when class labels of data is absent. It is clear that well-formed cluster has a larger likelihood, and should make more contribution to the consensus partition. Hence we adopt the internal validation measurements to approximate the marginal likelihood of a cluster, along with the aforementioned weighting scheme

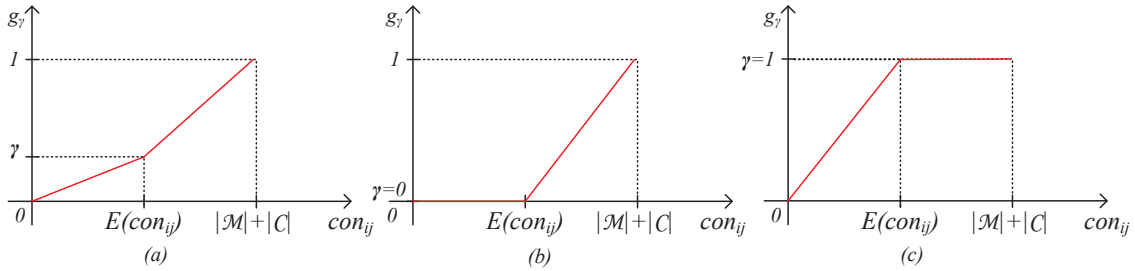


Fig. 4. The graphs of function g_γ with different parameters $\gamma \in [0, 1]$

based on two-level consistencies as a refinement to provide a more precise weight for each cluster. Here we adopt *Silhouette Coefficient* [40] as the evaluation metric, which evaluates the internal quality of clustering by comparing pairwise distances within and between the clusters. It can be scaled to the interval [0,1], and is easy to calculate.

Silhouette Coefficient

Given a base partition λ_i, the *Silhouette Coefficient of point* $\mathbf{x_t}$ in the partition λ_i is defined in Eq.14. Here $\pi_{i\leftarrow t}$ represents the cluster that $\mathbf{x_t}$ belongs to, and $\mathcal{N}(\pi_{i\leftarrow t})$ stands for the nearest cluster of $\pi_{i\leftarrow t}$ in λ_i. $dist(\cdot,\cdot)$ denotes the Euclidean distance between two instances, to keep consistent with the Euclidean distance metric adopted in *k-means*.

$$\sigma_t^i = \frac{\frac{\sum_{x\in\pi_{i\leftarrow t}} dist(x_t,x)}{|\pi_{i\leftarrow t}|} - \frac{\sum_{x\in\mathcal{N}(\pi_{i\leftarrow t})} dist(x_t,x)}{|\mathcal{N}(\pi_{i\leftarrow t})|}}{max(\frac{\sum_{x\in\pi_{i\leftarrow t}} dist(x_t,x)}{|\pi_{i\leftarrow t}|}, \frac{\sum_{x\in\mathcal{N}(\pi_{i\leftarrow t})} dist(x_t,x)}{|\mathcal{N}(\pi_{i\leftarrow t})|})} \tag{14}$$

The *Silhouette Coefficient of a cluster* π_{ij} is the average over all points in π_{ij}, as described in Eq.15.

$$\sigma_{ij} = \frac{\sum_{\mathbf{x_t}\in\pi_{ij}} \sigma_t^i}{|\pi_{ij}|} \tag{15}$$

According to Eq. 5, the final weight of j-th cluster in λ_i can be derived by multiplication of its *Silhouette Coefficient* and consistency with constraints:

$$w_{ij} = \sigma_{ij} \times \varphi(\pi_{ij}) \tag{16}$$

3.6 Partition of the Refined Co-association Matrix

Next, the weights of clusters are utilized to refine the original co-association matrix. Specifically, a weight matrix $\mathbf{W}$ corresponding to $\mathbf{H}$ is defined as:

$$\mathbf{W} = \begin{pmatrix} w_{11} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_{lk_l} \end{pmatrix} \tag{17}$$

$\mathbf{W}$ is a $K\times K$ diagonal matrix, where $K = \sum_{i=1}^{l} k_i$. Each element on the diagonal is the weight of a cluster. Then the refined co-association matrix $\mathbf{S}$ could be defined by Eq.18:

$$\mathbf{S} = \frac{1}{l}\mathbf{HWH}^\top \tag{18}$$

After the construction of co-association matrix, for simplicity, we use two widely adopted methods [4], Cluster-based Similarity Partitioning (CSPA) and Spectral Clustering (SPEC) to partition the final graph in order to produce a final consensus clustering. For more details of CSPA and SPEC, please refer to [4].

The whole procedure of *WECR k-means* is described in Algorithm 1 in Appendix A. First, an ensemble of base partitions is generated and the complete binary membership matrix is derived (line 1∼6). Then for each partition λ_i, the *clustering-level consistency* μ_i, *internal validation measure* σ_{ij} and *cluster-level consistency* ν_{ij} of clusters in λ_i are computed, and the weight vector $\mathbf{w}_i$ of partition λ_i is obtained(line 7∼18). Next, the refined co-association matrix $\mathbf{S}$ is derived by a matrix multiplication using the weight matrix $\mathbf{W}$ built from all weight vectors $\mathbf{w}_1, \mathbf{w}_2 ... \mathbf{w}_l$. Finally, a consensus partition method is applied to $\mathbf{S}$ to get the final solution(line 19∼22).

3.7 Time Complexity Analysis

Compared with "Constraint-first" approach, *WECR k-means* is much more computational efficient as it avoids performing time-consuming semi-supervised clustering algorithms. Hence the main reduction in time cost comes from the ensemble generation step, while the time cost is the same in the consensus partition step which is determined by the adopted consensus approach. In ensemble generation, *WECR k-means* is with a complexity of $O(l * N * k * p)$, where l stands for the size of ensemble, N is the number of instances, p is the number of features, and k is the number of clusters. For "Constraint-first" approach, this step would have a much higher complexity. For example, the E^2CP ensemble has a complexity of $O(l * k' * N^2 * p)$, where k' denotes the number of nearest neighbors when constructing *k-NN graph*. For *COP-Kmeans*, the actual time cost relies on the exact situation of constraints satisfaction in the iterations.

The additional time cost of our method comes from the adaptive weighting procedure, which can mainly be decomposed into three parts: (1) calculation of clustering-level consistency (with complexity $O(l * |M + C|)$); (2) calculation of cluster-level consistency($O(l*max(|M+C|, k))$, k is the average number of clusters in each base partition, and $|M+C|$ denotes the total number of constraints); (3) calculation of *Silhouette Coefficient*($O(l * N^2 * p)$). Note that the overall complexity of first two parts is $O(l*max(|M+C|, k))$, which is independent of the number of instances. In a large dataset, $l, |M + C|$ and k are often far less than the total number of instances N. Hence the overall complexity of building the weighted co-association matrix is $O(l * N^2 * (k + p))$.

TABLE 2
Overview of datasets

Name	#instances	#features	#classes
Segmentation	2310	19	7
Optdigits	3823	64	10
Drift	13910	256	6
ISOLET-5	1559	617	26
Fashion-MNIST	10000	784	10
MNIST4000	4000	784	10
COIL20	1440	1024	20
SRBCT	83	2309	4
Leukemia1	72	5328	3
Prostate	102	10510	2
Leukemia2	72	11226	3
LungCancer	203	12601	4

4 Experiments

4.1 Datasets and Constraint Generation Method

We evaluate our method and other clustering algorithms on twelve real-world datasets, including four UCI datasets, Optdigits, ISOLET-5, Segmentation, Drift [41], five high-dimensional and small-sample Microarray datasets [42], and three image datasets COIL20, MNIST4000 [43] and Fashion-MNIST [44]. Details of the datasets are listed in Table 2. In order to distinguish them from Microarray datasets, we call the other seven datasets normal datasets. All pairwise constraints used in our experiments are generated artificially based on the ground-truth labels, following the main framework presented by K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl [37]. In order to thoroughly investigate the algorithms, we consider two types of pairwise constraints.

(1) Random constraints. Generation of random constraints is summarized in Algorithm 2 in Appendix A. Slightly different from Wagstaff's method [37], the must-link constraints (line 5∼17) and cannot-link constraints (line 18∼25) are generated separately in order to generate must-link constraints proportionate to the size of each class (line 6).

(2) Noisy constraints. In order to explore the robustness of the algorithms to noisy supervision, some *noisy constraints*, i.e. constraints providing incorrect information, are generated using Algorithm 3 in Appendix A.

Note that in order to explore the relationships between constraints, a standard process of calculating transitive closure of a given constraint set [2] should be applied. The detailed descriptions of Algorithm 2 and Algorithm 3 are displayed in Appendix A.

4.2 Evaluation Metric

Since the datasets used in the experiments have ground-truth labels, we use *Normalized Mutual Information (NMI)* to evaluate the consensus partition, which measures shared information between the consensus partition and ground-truth partition. According to [45], $1 - NMI_{max}$, also called *normalized information distance*, has the normalization and metric properties. It can be used as a general measure to compare two partitions as a distance metric. Hence we use the NMI_{max} between a consensus partition λ and the ground-truth partition λ^*, as defined in Eq. 21, where $H(\lambda^*)$ and $H(\lambda)$ denote the information entropy of λ^* and λ respectively (Eq. 19), $MI(\lambda^*, \lambda)$ denotes the mutual information (Eq. 20), and π^* and π denote a cluster in λ^* and λ respectively.

$$H(\lambda) = -\sum_i \frac{|\pi_i|}{|\mathbf{X}|} log(\frac{|\pi_i|}{|\mathbf{X}|}) \tag{19}$$

$$MI(\lambda^*, \lambda) = \sum_i \sum_j \frac{|\pi_i \cap \pi_j^*|}{|\mathbf{X}|} log(\frac{|\mathbf{X}||\pi_i \cap \pi_j^*|}{|\pi_i||\pi_j^*|}) \tag{20}$$

$$NMI_{max}(\lambda, \lambda^*) = \frac{MI(\lambda, \lambda^*)}{max(H(\lambda), H(\lambda^*))} \tag{21}$$

4.3 Experimental Setting

We compare *WECR k-means* with classical and state-of-the-art semi-supervised algorithms, i.e. *COP-Kmeans* and its ensemble [37], E^2CP [39] and its ensemble, *SSKKE* [46] and *DCECP* [47]. Additionally, we also report the results of two unsupervised ensemble clustering algorithms, i.e. *KCC* [48] and *U-SENC* [49].

For cluster ensemble generation, 100 base partitions are generated for each ensemble. CSPA and SPEC [4] are adopted to combine the base partitions of *COP-Kmeans*, E^2CP and *WECR k-means*.

The methods and their own settings in the experiments are summarized in Table 5 in Appendix B. Our methods are implemented in python 3.6.3 using scikit-learn 0.19.1 [50], while in CSPA we partition the similarity graph using METIS [51]. All experiments are conducted in a PC with i7-3770 CPU and 8GB RAM.

In the following experiments, we first compare the computational efficiency of multiple semi-supervised and unsupervised clustering ensemble approaches over different datasets. Then we study the effect of the hyparameters. Finally, we investigate the performance of different algorithms in three cases: (1) Experiment I: we generate a random constraint pool of size $2N$. Then we compare the algorithms in an incremental setting, that is, gradually adding new constraints into the current constraint set from the pool, with the set size increasing from $0.25N$ to $2N$. (2) Experiment II: to show the stability of the algorithms with respect to different random constraints, we report the results with ten constraints sets of the same size, for five size $0.25N \sim 2N$ respectively. (3) Experiment III: we add different levels $(5\%, 10\%, 15\%, 20\%)$ of noise into a random constraints set of size N, and explore the effect of noisy constraints on the performances of the algorithms.

4.4 Comparison on Computational Efficiency

We report the average time cost of ten random experiments of the ensemble algorithms (with 100 base partitions) on every dataset in Table 3, where the datasets are arranged in ascending order of their sample size. The results show that on all the datasets, *WECR k-means* is superior to other semi-supervised clustering ensemble algorithms, especially on the larger datasets. Compared with the two unsupervised clustering ensemble algorithms, it outperforms *U-SENC* on the Microarray datasets, and outperforms *KCC* on all the datasets except Drift.

Table 6 and Fig.7 in Appendix B show the average time cost of generating a base partition for the clustering ensemble algorithms with constraint sets of size N on twelve datasets respectively. The results show why *WECR k-means* is fast. It is much more efficient in ensemble generation than the others except *U-SENC*, while its time cost in weighting operation of a base partition is also very small. Detailed discussions are given in Appendix B.

4.5 The Effect of Hyperparameters

In order to explore the effect of hyperparameters of *WECR k-means*, extensive experiments are conducted on random constraint set of size N. For each dataset, several libraries are generated with varying sampling rates r_i (on sample) and r_f (on features). The results show that for the five Microarray datasets, the best results are obtained with $r_i \in [0.5, 0.7]$ and $r_f \in [0.2, 0.3]$, while for the other datasets with more instances and fewer features, the best sampling rates settings are $r_i \in [0.2, 0.3]$ and $r_f \in [0.6, 0.7]$. The reason may lie in the fact that for small-sample and high-dimensional data, more instances should be sampled to capture the distribution of the original data, while there exists great redundancy in the high dimensional feature space so that a smaller proportion of features can achieve good performance. The situation is just the opposite for other datasets. In the following experiments, we will use these parameter settings for *WECR k-means*. The hyperparameter γ, which is the scaling factor between *clustering-level* and *cluster-level* weights, is tuned in $[0, 1]$ with 0.1 as the step size. Limited by the length, the tuning method of γ is described in Appendix C.

4.6 Experiment I: Comparison on Incremental Constraints

Experiment I was carried out ten times. The average ranks of different algorithms over twelve datasets are displayed in Table 4 (Fig. 8 in Appendix B), which shows that *WECR k-means* with CSPA obtains the highest average ranking with all sizes of constraints set. *WECR k-means* with SPEC ranks first when the constraints size is $0.5N$, and ranks second in the other cases. Under each constraint set size, to evaluate the statistical significance , we perform Friedman Test on the averaged ranks of twelve algorithms on the twelve datasets in the ten random trials. The results show that the performances of *WECR k-means* is significantly different from that of other algorithms (at significant level $p < 0.05$), while there are no significant differences between *WECR k-means* with CSPA and *WECR k-means* with SPEC. Look into the single semi-supervised clustering algorithms, the performances of *COP-Kmeans* are always relatively poor, while E^2CP often obtains better partitions, especially on Optdigits and MNIST4000. Table 4 also indicates the effectiveness of *DCECP* and E^2CP, both of which have high average rankings. The performance of *SSKKE* is not stable in different run of the ten experiments. It achieves better performances in some cases, while perform much worse in other cases.

We also report the average and standard deviation of NMI scores of all the algorithms over all datasets in Table 7 in Appendix B, which provides a clear overview of the advantages of *WECR k-means* over other methods.

To more intuitively show the significant differences among the algorithms, Fig.5 and Fig.6 display the result of one of the experiments. They show the *NMI* value of the twelve algorithms on twelve datasets respectively, with the horizontal axis representing different number of constraints. For a clear comparison, Fig.5 displays the results of ten algorithms, i.e. two *WECR k-means* algorithms, three E^2CP related algorithms, three *COP-Kmeans* related algorithms, and two unsupervised methods, *KCC* and *U-SENC*. Fig.6 displays the results of two *WECR k-means* algorithms, two semi-supervised ensemble methods *SSKKE* and *DCECP*, and two unsupervised methods, *KCC* and *U-SENC*. The results also reveal some interesting findings as follows.

(1) Ensemble approaches do not always improve the performances of individual semi-supervised clusterings. It can be concluded that *COP-Kmeans* is likely to get promoted with ensemble approaches while E^2CP is less likely to be enhanced. This can also be illustrated by the average ranks displayed in Table 4.

Specifically, using CSPA as the consensus method, *COP-Kmeans* ensemble perform much better than *COP k-means*, on Optdigits, MNIST4000, COIL20, Segmentation, SRBCT, and Leukemia2. Using SPEC, *COP-Kmeans* ensemble perform much better than *COP-Kmeans* on Optdigits, MNIST4000, LungCancer, Leukemia1 and Leukemia2. However, the performances of *COP-Kmeans* ensemble are unstable. They are able to achieve very good partitions on some datasets, but have very poor performances on other datasets, e.g. LungCancer. The case is quite different when performing E^2CP ensemble. In most cases, the performances of E^2CP ensemble drop sharply compared to E^2CP, except for being applied on Prostate and LungCancer with SPEC as the consensus approach, and on COIL20 with CSPA or SPEC. It indicates that E^2CP is less likely to be enhanced by ensemble approaches. The difference in performance between E^2CP and *COP-Kmeans* coincides with the principal of ensemble learning: a set of weak learners can create a single strong learner, since *COP-Kmeans* usually performs much worse than E^2CP, the ensemble approaches are capable to boost the former rather than the latter.

(2) The data characteristics, e.g. curse of dimensionality, has great impact on the performances. In terms of overall performance, the clustering algorithms work better on normal datasets than Microarray datasets. E^2CP ensemble is an exception because it does not improve the performance of E^2CP. The results also show higher variances on Microarray datasets (except Prostate) compared with normal datasets, which is probably due to their high number of features and small sample size. Besides, it is worth noting that all algorithms perform poorly on Prostate, which provides a great challenge for clustering.

(3) When the number of constraints increases, the performances of algorithms might not get promoted accordingly, on the contrary they exhibit some degree of fluctuations, especially on the Microarray datasets. Among them, *WECR k-means* and *DCECP* are relatively robust and show less fluctuations. As mentioned in Section 2, the main reason may lie in the fact that the utility of constraints is demonstrated not consistent for different semi-supervised algo-

TABLE 3
Average time cost(in seconds) of the entire algorithms on twelve datasets

Methods	WECR k-means (CSPA)	WECR k-means (SPEC)	COP-kmeans (CSPA)	COP-kmeans (SPEC)	E^2CP (CSPA)	E^2CP (SPEC)	DCECP	SSKKE	U-SENC	KCC
Leukemia2	7.59	7.46	14.59	14.45	50.58	50.56	65.59	90.10	11.92	48.59
Leukemia1	3.80	3.45	6.51	6.16	35.94	35.89	46.61	92.42	8.37	19.72
SRBCT	2.94	2.59	4.02	3.66	34.57	33.52	44.83	153.05	6.32	12.33
Prostate	8.27	7.87	54.09	53.68	61.58	60.52	79.93	265.66	15.21	36.48
LunCancer	24.04	23.29	42.40	41.65	154.78	151.67	200.98	2137.35	26.18	165.26
COIL20	39.05	38.21	974.13	973.29	668.99	666.87	869.17	345289.43	18.33	259.38
ISOLET-5	87.74	86.39	1010.54	1013.88	1081.51	1080.31	1403.98	416088.29	12.54	286.20
Segmentation	11.56	8.56	745.24	744.75	699.90	698.48	907.59	1890166.43	4.91	35.64
OPTDIGITS	31.30	25.86	1071.98	1076.94	1661.28	1660.50	2157.76	1627264.84	6.66	92.25
MNIST4000	74.40	68.59	1332.05	1326.23	2139.63	2136.79	2779.50	1479335.51	26.32	613.05
Fashion-MNIST	205.37	194.60	1509.00	1512.30	12476.39	12472.85	16212.36	1775200.49	92.17	1578.53
Drift	700.11	650.18	2297.11	2316.12	29130.65	29123.52	37848.95	1739699.07	662.37	429.31

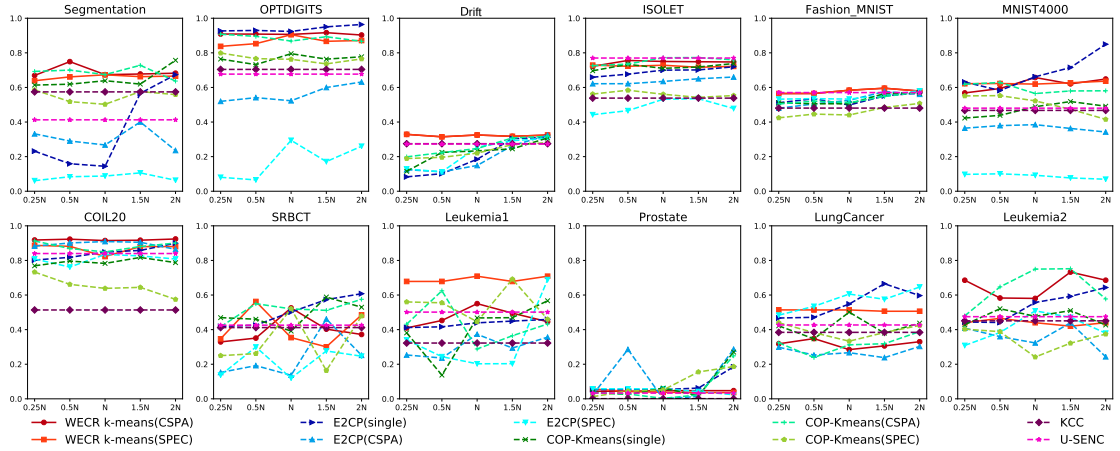


Fig. 5. Comparison of ten algorithms on twelve datasets with incremental constraints ($0.25N \sim 2N$): to show the impact of curse of dimensionality, the graphs are arranged in ascending order of the number of features of each datasets. The horizontal axis represents different number of constraints, and the vertical axis represents the *NMI* value between the clustering result and the ground-truth partition.

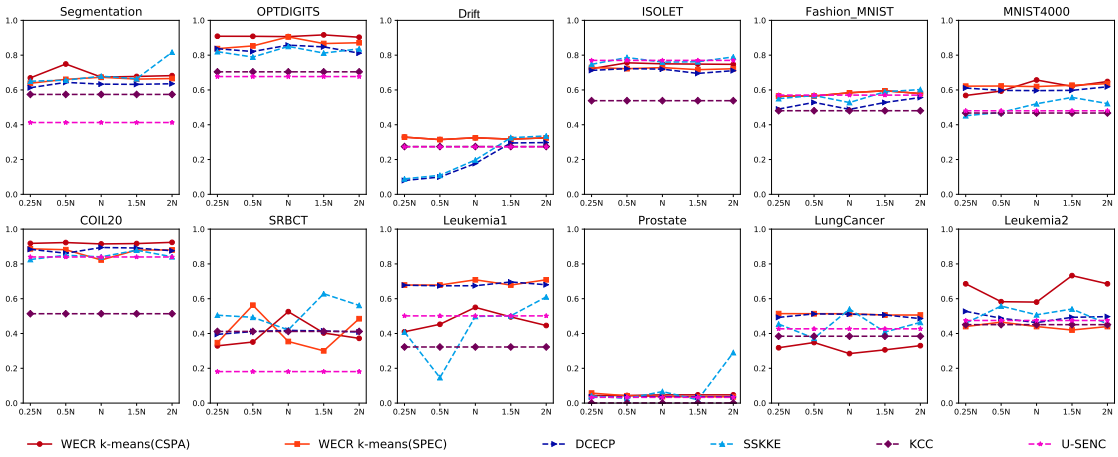


Fig. 6. Comparison of six algorithms on twelve datasets with incremental constraints ($0.25N \sim 2N$): to show the impact of curse of dimensionality, the graphs are arranged in ascending order of the number of features of each datasets. The horizontal axis represents different number of constraints, and the vertical axis represents the *NMI* value between the clustering result and the ground-truth partition.

rithms [8]. The high variance of the experimental results of the same algorithm on the same data set stems from the effect of random constraints, which have great impact on the performances of semi-supervised algorithms, and do not always improve the performance. Since in real applications the utility of priori knowledge or user feedbacks is unlikely guaranteed in advance, it once again raises the question of whether it is necessary to perform time-consuming semi-supervised algorithm in ensemble generation, especially when the dataset is very large and constraint information is very limited.

(4) Consensus methods, i.e. CSPA and SPEC, play an important role on the performances, when the ensemble approach works. It is worth noting that the performance of the two consensus methods, CSPA and SPEC, is highly dependent on the datasets. It is quite plausible that CSPA

TABLE 4
The average ranks of performances of the twelve algorithms over twelve datasets with different number of constraints

Methods	0.25N	0.5N	N	1.5N	2N
WECR k-means(CSPA)	2.3	2.0	2.3	2.3	2.2
WECR k-means(SPEC)	2.3	2.2	2.4	2.6	3.3
DCECP	4.2	4.3	5.0	4.8	5.0
SSKKE	8.8	8.4	8.7	7.8	6.8
U-SENC	6.3	6.6	6.8	7.1	7.4
KCC	8.4	8.9	8.8	9.1	8.9
E2CP(single)	6.1	5.6	4.3	4.5	3.7
E2CP(CSPA)	9.7	9.0	9.0	8.3	7.4
E2CP(SPEC)	8.4	8.1	8.1	7.3	6.5
COP-Kmeans(single)	10.5	10.4	9.7	9.7	10.7
COP-Kmeans(CSPA)	7.8	7.9	8.1	8.2	8.5
COP-Kmeans(SPEC)	7.9	8.3	8.2	8.3	8.5

might outperform SPEC in one dataset, while SPEC might outperform CSPA in another dataset. As a result, the clustering approaches also inherit their dependency on specific datasets. For example, for *WECR k-means* and *COP-Kmeans* , when using CSPA their performances are better than using SPEC on Leukemia2, while the case is just the opposite on Leukemia1. The ensemble approach does not work for *E2CP* on the two datasets. On LungCancer, using CSPA the three ensemble methods perform poorly, while all of them perform much better using SPEC. If we compare them using the same consensus function, we can see more clearly the effectiveness of our approach.

(5) Compared with unsupervised ensemble clustering methods, the semi-supervised algorithms even achieve worse performances in many cases. This is a relatively rare phenomenon on the normal datasets, e.g. applying E^2CP and its ensemble on Segmentation, and applying E^2CP ensemble on Optdigits and MNIST4000. Note that *U-SENC* achieve very good performances. On ISOLET, it always outperform all the semi-supervised algorithms. On the Microarray datasets, while all the semi-supervised algorithms exhibit relatively higher variance, they perform worse than *KCC* and *U-SENC* in many cases. This also illustrates once again the impact of curse of dimensionality on the performances. It can be observed that *WECR k-means* and *DCECP* always perform better than the unsupervised methods, except in few cases.

In order to confirm the above observations, Experiment I was carried out ten times. In each trial, a new constraint pool of size $2N$ was generated randomly. For each constraint set, we then repeated the experiment five times respectively. The results are similar to each other in terms of *NMI* when the experiments are conducted on the same constraint set. This shows that the performance of different algorithms is consistent when the constraint set remains the same. For each of ten experiments, we compare the algorithms with the same incremental setting. To more clearly show the changes of the performances of twelve algorithms, we draw Fig.9 in Appendix B to display the average performances over ten random experiments of each algorithms on the datasets. It shows that although the results are different from each other in terms of *NMI*, we can still draw similar conclusions on the comparison of the different algorithms.

4.7 Experiment II: Comparison on Different Constraint Sets

To intuitively show the adaptability of *WECR k-means* to different constraints, clustering results with respect to ten different random constraints sets with the same size N are displayed in Fig.12 in Appendix B. The horizontal axis represents ten different constraint sets and the vertical axis denotes *NMI* value. It shows that the results of mutual comparisons of these algorithms are quite consistent with Experiment I.

Due to the randomness of constraints, the results of all algorithms exhibit some degree of variances. Similarly, the curse of dimensionality may increase the variance on Microarray datasets. Overall, the performances of *WECR k-means* are stable and they are demonstrated to be more adaptive to different constraints through comparisons on standard deviations of *NMI* (see Table 8 in Appendix B). The results of remaining four size of constraint sets, $0.25N, 0.5N, 1.5N, 2N$, are displayed in Fig.10, Fig.11, Fig.13, and Fig.14 in Appendix B.

4.8 Experiment III: Comparison on Constraints with Noise

Fig.15 in Appendix B displays the performances of the algorithms with respect to different levels of noise (0, 5%,10%,15%,and 20%) in the random constraint sets of size N. Generally, the results could be divided into two parts to analyze according to the characteristics of the datasets.

(1) Generally speaking, for the seven normal datasets, the algorithms except *WECR k-means*, $E^2CP(single)$ and *DCECP* show varying degrees of downward trend or fluctuations in the performance curves when the noise level gradually increases. When the noise level is 20%, even on the normal datasets, their performances can be degraded to close to that of the two unsupervised algorithms. *WECR k-means* is stable and has the best overall performances.

(2) For the five Microarray datasets, the algorithms tend to exhibit relatively larger fluctuations with the increasing of noise level. It is mainly due to the data characteristics of high dimension and small sample. In this case, *WECR k-means* and *DCECP* also perform better than other algorithms in terms of robustness. It can be observed that *WECR k-means* always achieves large *NMI* values even with noisy constraints, except on Prostate. In this case, E^2CP shows slightly worse performances than it does in the first case, except on SRBCT.

The standard deviations of the results (see Table 9 in Appendix B) also demonstrate that *WECR k-means* are the most robust to noisy constraints among all the ten semi-supervised algorithms while keeping a high quality. The semi-supervised algorithms are more susceptible to noisy constraints, which has great impact on the qualities of base partitions and subsequently deteriorates their ensemble. The mechanism of E^2CP which does not use constraints directly may protect it from the negative influences of noisy constraints to some degree, so E^2CP is less affected. Note that *DCECP* also employs E^2CP to generate base partitions, so it is also less affected. Similarly, in order to confirm the above observations, Experiment III was carried out ten times, with ten different random constraint sets of size

N, and the conclusions are consistent. The box plots of the average ranks over ten experiments are displayed in Fig.16 in Appendix B. According to the results of average ranks, we also perform Friedman Test which shows that the performances of *WECR k-means* are significantly different from that of other algorithms (at significant level $p < 0.05$), while there are no significant differences between *WECR k-means* with CSPA and *WECR k-means* with SPEC.

5 Conclusion

In this paper, we have introduced a "Constraint-first" framework of using constraints in semi-supervised cluster ensemble and demonstrated its effectiveness. It is much more computational efficient than the traditional framework, as it avoids performing time-consuming semi-supervised clustering. Moreover, in a dynamic learning environment with an increasing or changing supervisory information like user feedbacks, *WECR k-means* does not require reproducing a cluster ensemble and re-calculating the *Silhouette Coefficients* of base partitions. Only the *clustering-level* and *cluster-level consistencies* are required to be re-examined according to the new constraint set. In this case, the main time cost of *WECR k-means* rests with the adopted consensus function. Also, this framework is much more robust to noisy constraints while semi-supervised clustering algorithms are usually susceptible to noise.

The co-association matrix based consensus approach used in this work is inefficient with a complexity of $O(N^2)$. In the future, we will investigate how to combine *WECR k-means* with other scalable consensus approaches. We will also investigate more scalable internal validation measurements.

Acknowledgments

This work is supported by the Natural Science Foundation of China under Grant NO.61672441 and NO.61673324, the Natural Science Foundation of Fujian under Grant NO.2018J01097 and NO.2018J01577, the Shenzhen Basic Research Program No. JCYJ20170818141325209, the State Scholarship Fund of China Scholarship Council under Grant NO.201706315020, and the Fundamental Research Funds for the Central Universities of China under Grant No.20720160085.

References

[1] Z. Yu, L. Li, J. Liu, J. Zhang, and G. Han, "Adaptive noise immune cluster ensemble using affinity propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 12, pp. 3176–3189, 2015.

[2] Z. Yu, Z. Kuang, J. Liu, H. Chen, J. Zhang, J. You, H. Wong, and G. Han, "Adaptive ensembling of semi-supervised clustering solutions," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1577–1590, 2017.

[3] Z. Yu, H. Chen, J. You, H. Wong, J. Liu, L. Li, and G. Han, "Double selection based semi-supervised clustering ensemble for tumor clustering from gene expression profiles," *IEEE/ACM Trans. Comput. Biology Bioinform.*, vol. 11, no. 4, pp. 727–740, 2014.

[4] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[5] Z. Yu, P. Luo, J. You *et al.*, "Incremental semi-supervised clustering ensemble for high dimensional data clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 701–714, 2016.

[6] A. M. Iqbal, A. Moh'd, and Z. A. Khan, "Semi-supervised clustering ensemble by voting," *CoRR*, vol. abs/1208.4138, 2012.

[7] F. Yang, T. Li, Q. Zhou, and H. Xiao, "Cluster ensemble selection with constraints," *Neurocomputing*, vol. 235, pp. 59–70, 2017.

[8] I. Davidson, K. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms," in *Knowledge Discovery in Databases: PKDD 2006, 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, September 18-22, 2006, Proceedings*, 2006, pp. 115–126.

[9] I. Davidson and S. S. Ravi, "The complexity of non-hierarchical clustering with instance and cluster level constraints," *Data Min. Knowl. Discov.*, vol. 14, no. 1, pp. 25–61, 2007.

[10] ——, "Intractability and clustering with constraints," in *Proceedings of the Twenty-Fourth International Conference on Machine Learning , Corvallis, Oregon, USA, June 20-24*, 2007, pp. 201–208.

[11] B. J. Jain, "Condorcet's Jury Theorem for Consensus Clustering and its Implications for Diversity," *ArXiv e-prints*, Apr. 2016.

[12] V. B. Berikov and I. Pestunov, "Ensemble clustering based on weighted co-association matrices: Error bound and convergence properties," *Pattern Recognition*, vol. 63, pp. 427–436, 2017.

[13] N. Russell, T. B. Murphy, and A. E. Raftery, "Bayesian model averaging in model-based clustering and density estimation," *arXiv preprint arXiv:1506.09035*, 2015.

[14] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical science*, pp. 382–401, 1999.

[15] A. Fred and A. Jain, "Data clustering using evidence accumulation," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 4. IEEE, 2002, pp. 276–280.

[16] A. Topchy, A. Jain, and W. Punch, "Combining multiple weak clusterings," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003, pp. 331–338.

[17] X. Fern and C. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proceedings of the 20th International Conference on Machine Learning*, 2003, pp. 186–193.

[18] B. Fischer and J. Buhmann, "Path-based clustering for grouping of smooth curves and texture segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 4, pp. 513–518, 2003.

[19] B. Minaei-Bidgoli, A. Topchy, and W. Punch, "Ensembles of partitions via data resampling," in *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, vol. 2. IEEE, 2004, pp. 188–192.

[20] S. Dudoit and J. Fridlyand, "Bagging to improve the accuracy of a clustering procedure," *Bioinformatics*, vol. 19, no. 9, pp. 1090–1099, 2003.

[21] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," *Machine learning*, vol. 52, no. 1, pp. 91–118, 2003.

[22] G. A. Hanan and S. K. Mohamed, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 160–173, 2008.

[23] A. Topchy, A. Jain, and W. Punch, "A mixture model of clustering ensembles," in *Proc. SIAM Intl. Conf. on Data Mining*, 2004.

[24] H. Wang, H. Shan, and A. Banerjee, "Bayesian cluster ensembles," *Statisical Analysis and Data Mining*, vol. 4, no. 1, pp. 54–70, 2011.

[25] A. P. Topchy, M. H. C. Law, A. K. Jain, and A. L. N. Fred, "Analysis of consensus partition in cluster ensemble," in *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK*, 2004, pp. 225–232.

[26] X. Fern and W. Lin, "Cluster ensemble selection," *Statistical Analysis and Data Mining*, vol. 1, no. 3, pp. 128–141, 2008.

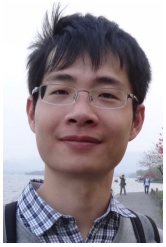
[27] F. Yang, X. Li, Q. Li, and T. Li, "Exploring the diversity in cluster ensemble generation: Random sampling and random projection," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4844–4866, 2014.

[28] S. Hadjitodorov, L. Kuncheva, and L. Todorova, "Moderate diversity for better cluster ensembles," *Information Fusion*, vol. 7, no. 3, pp. 264–275, 2006.

[29] M. Pividori, G. Stegmayer, and D. H. Milone, "Diversity control for improving the analysis of consensus clustering," *Information Sciences*, vol. 361-362, pp. 120–134, 2016.

[30] L. I. Kuncheva and S. T. Hadjitodorov, "Using diversity in cluster ensembles," in *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics: The Hague, Netherlands, 10-13 October 2004*, 2004, pp. 1214–1219.

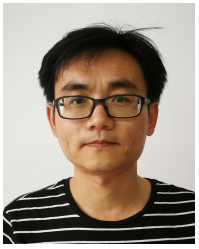
[31] T. Li and C. Ding, "Weighted consensus clustering," *Proceedings of SDM*, 2008.
[32] L. Lam and C. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *IEEE Trans. Systems, Man, and Cybernetics, Part A*, vol. 27, no. 5, pp. 553–568, 1997.
[33] F. Wang, X. Wang, and T. Li, "Generalized cluster aggregation," in *Proceedings of the 21st international jont conference on Artifical intelligence*. Morgan Kaufmann Publishers Inc., 2009, pp. 1279–1284.
[34] C. Domeniconi and M. Al-Razgan, "Penta-training: Clustering ensembles with bootstrapping of constraints," in *Workshop on Supervised and Unsupervised Ensemble Methods and their Applications*, 2008.
[35] D. Greene and P. Cunningham, "Constraint selection by committee: An ensemble approach to identifying informative constraints for semi-supervised clustering," *Machine Learning: ECML 2007*, pp. 140–151, 2007.
[36] Y. Yang, H. Wang, C. Lin, and J. Zhang, "Semi-supervised clustering ensemble based on multi-ant colonies algorithm," *Rough Sets and Knowledge Technology*, pp. 302–309, 2012.
[37] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *Proc. of the 18th International Conference on Machine Learning*, 2001, pp. 577–584.
[38] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," in *Proc. of the 19th International Conference on Machine Learning*, 2002, pp. 19–26.
[39] Z. Lu and Y. Peng, "Exhaustive and efficient constraint propagation: A graph-based learning approach and its applications," *International Journal of Computer Vision*, vol. 103, no. 3, pp. 306–325, 2013.
[40] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. C, pp. 53–65, 1987.
[41] M. Lichman, "UCI machine learning repository," 2013.
[42] A. R. Statnikov, I. Tsamardinos, Y. Dosbayev, and C. F. Aliferis, "GEMS: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data," *International Journal of Medical Informatics*, vol. 74, no. 7-8, pp. 491–503, 2005.
[43] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.
[44] H. Xiao, K. Rasul, and R. Vollgraf, "Fashon-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[45] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.
[46] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," *Machine learning*, vol. 74, no. 1, pp. 1–22, 2009.
[47] Z. Yu, P. Luo, J. Liu, H.-S. Wong, J. You, G. Han, and J. Zhang, "Semi-supervised ensemble clustering based on selected constraint projection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2394–2407, 2018.
[48] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen, "K-means-based consensus clustering: A unified view," *IEEE transactions on knowledge and data engineering*, vol. 27, no. 1, pp. 155–169, 2015.
[49] D. Huang, C.-D. Wang, J. Wu, J.-H. Lai, and C. K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2019.
[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[51] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.



Yongxuan Lai received the PhD degree in computer science from Renmin University of China in 2009. He is currently an associate professor in Software School, Xiamen University, China. He is an visiting scholar during Sep. 2017 - Sep. 2018 at University of Queensland, Australia. His research interests include network data management, vehicular ad-hoc networks, and big data management and analysis.

Songyao He Received B.S degree in Department of Automation from Xiamen University in 2018. She is currently a master student from Department of Automation at Xiamen University.Her research interests is ensemble learning and data mining.

Zhijie Lin received B.S degree in Software Engineering from Xiamen University in 2015. He is currently a master student from Software School, Xiamen University. His research interests is data mining.



Fan Yang received his Ph.D. degree in Control Theory and Control Engineering from Xiamen University in 2009. He is currently an associate professor in the Department of Automation at Xiamen University. His research interests include feature selection, ensemble learning, and bioinformatics.

Qifeng Zhou received her Ph.D. degree in Control Theory and Control Engineering from Xiamen University in 2007. She is currently an associate professor in the Department of Automation at Xiamen University. Her research interests are machine learning, intelligence system, and digital image processing.

Xiaofang Zhou received the bachelor's and master's degrees in computer science from Nanjing University in 1984 and 1987, respectively, and the PhD degree in computer science from the University of Queensland, in 1994. He is a professor of computer science at the University of Queensland, and also head of Data Engineering and Pattern Recognition Research Division at UQ. He is a specially appointed adjunct professor of computer science at Soochow University, China. His research interests include spatial and multimedia databases, high performance query processing, web information systems, data mining, and data quality management. He is a fellow of the IEEE.