

Weighted-Object Ensemble Clustering

Yazhou Ren

School of Computer Science and Engineering
South China University of Technology
Guangzhou, 510006, China
Email: yazhou.ren@mail.scut.edu.cn

Guoji Zhang

School of Sciences
South China University of Technology
Guangzhou, 510640, China
Email: magjzh@scut.edu.cn

Carlotta Domeniconi

Department of Computer Science
George Mason University
Fairfax, 22030, USA
Email: carlotta@cs.gmu.edu

Guoxian Yu

College of Computer and Information Science
Southwest University
Chongqing, 400715, China
Email: gxyu@swu.edu.cn

Abstract—Ensemble clustering, also known as consensus clustering, aims to generate a stable and robust clustering through the consolidation of multiple base clusterings. In recent years many ensemble clustering methods have been proposed, most of which treat each clustering and each object as equally important. Some approaches make use of weights associated with clusters, or with clusterings, when assembling the different base clusterings. Boosting algorithms developed for classification have also led to the idea of considering weighted objects during the clustering process. However, not much effort has been put towards incorporating weighted objects into the consensus process.

To fill this gap, in this paper we propose an approach called *Weighted-Object Ensemble Clustering* (WOEC). We first estimate how difficult it is to cluster an object by constructing the co-association matrix that summarizes the base clustering results, and we then embed the corresponding information as weights associated to objects. We propose three different consensus techniques to leverage the weighted objects. All three reduce the ensemble clustering problem to a graph partitioning one. We present extensive experimental results which demonstrate that our WOEC approach outperforms state-of-the-art consensus clustering methods and is robust to parameter settings.

Keywords—Ensemble clustering, consensus clustering, graph partition, weighted objects.

I. INTRODUCTION

Clustering is the key step for many tasks in data mining. It is well known that off-the-shelf clustering methods may discover different patterns in a given set of data. This is because each algorithm has its own bias due to the optimization of different criteria. Furthermore, there is no ground truth to validate the result. Recently the use of clustering ensembles has emerged as a technique for overcoming problems with clustering algorithms [1], [2]. A clustering ensemble technique is characterized by two components: the mechanism to generate diverse partitions, and the consensus function to combine the input partitions into a final clustering. A clustering ensemble consists of different clusterings, obtained from multiple applications of any single algorithm with different initializations, or on various bootstrap samples of the available data, or from the application of different algorithms to the same data set. Clustering ensembles offer a solution to challenges inherent to clustering arising from its ill-posed nature: they can provide more robust and stable solutions by making use of the

consensus across multiple clustering results, while averaging out emergent spurious structures that arise due to the various biases to which each participating algorithm is tuned, or to the variance induced by different data samples.

Researchers have investigated how to combine clustering ensembles with subspace clusterings in an effort to address both the ill-posed nature of clustering and the curse-of-dimensionality in high dimensional spaces [3], [4]. A subspace clustering is a collection of weighted clusters, where each cluster has a weight vector representing the relevance of features for that cluster. In a subspace clustering ensemble, the consensus function makes use of both the clusters and the weight vectors provided by the base subspace clusterings. Work has also been done to evaluate the relevance of each base clustering (and assign weights accordingly), in an effort to improve the final consensus clustering [5]. To the best of our knowledge, no previous work has investigated how to use weights associated to *objects* within the clustering ensemble framework.

Several approaches have studied the benefits of weighting the objects in iterative clustering methods combined with boosting techniques [6], [7], [8]. Empirical observations suggest that large weights should be assigned to the objects that are hard to be clustered. As a result, centroid-based clustering methods move the centers towards the regions where the objects whose cluster membership is hard to be determined are located. In boosting, the weights bias the data distribution, thus making the region around the difficult points denser. With this interpretation of the weights, shifting the centers of the clusters corresponds to moving them towards the modes of the distribution modified by the weights. Nock and Nielsen [9] formulated clustering as a constrained minimization using a Bregman divergence, and recommended that the hard objects should be given large weights. They analyzed the benefits of using boosting techniques in clustering and introduced several weighted versions of classical clustering algorithms.

Inspired by this work, we propose an approach called *Weighted-Object Ensemble Clustering* (WOEC). We first estimate how difficult it is to cluster an object by constructing the co-association matrix that summarizes the base clustering results, and we then embed the corresponding information as

weights associated to objects. Different from boosting [10], the weights are not subject to iterative changes. We propose three different consensus techniques to leverage the weighted objects. All three reduce the ensemble clustering problem to a graph partitioning one. We present extensive experimental results which demonstrate that our WOEC approach outperforms state-of-the-art consensus clustering methods and is robust to parameter settings. The main contributions of this paper are as follows:

- i) To the best of our knowledge, for the first time weights associated to objects are used within clustering ensembles. In WOEC, a one-shot method is proposed to assign weights to objects, and then three weighted-object techniques are presented to compute the consolidated clustering.
- ii) A Weighted-Object k -means algorithm (WOKmeans) is proposed. Its superiority to k -means demonstrates the effectiveness of the one-shot weights' assignment.
- iii) Extensive experiments are conducted to analyze the performance of WOEC against existing consensus clustering algorithms, on several data sets and using three popular clustering evaluation measures.

The rest of this paper is organized as follows. In Section II, we review related work on ensemble clustering. In Section III, we introduce the WOEC methodology. Section IV gives the experimental settings and Section V analyzes the experimental results. Conclusions and future work are provided in Section VI.

II. RELATED WORK

Ensemble techniques were first developed for supervised settings. Empirical results have shown that they are capable of improving the generalization performance of the base classifiers [11]. This result has inspired researchers to further investigate the development of ensemble techniques for unsupervised problems, namely clustering.

Fred and Jain [12], [13] captured the information provided by the base clusterings in a co-association matrix, where each entry is the frequency according to which two objects are clustered together. The co-association matrix was used as a similarity matrix to compute the final clustering.

Strehl and Ghosh [2] proposed three graph-based methods, namely CSPA, HGPA, and MCLA. CSPA constructs a co-association matrix, which is again viewed as a pairwise similarity matrix. A graph is then constructed, where each object corresponds to a node, and each entry of the co-association matrix gives the weight of the edge between two objects. The METIS [14] package is used to partition the objects into k clusters. HGPA combines multiple clusterings by solving a hypergraph partitioning problem. In the hypergraph, each hyperedge represents a cluster, and it connects all the objects that belong to the corresponding cluster. The package HMETIS [15] is utilized to generate the final clustering. MCLA considers each cluster as a node in a meta-graph, and sets the pairwise similarity between two clusters as the ratio between the number of shared objects and the number of total objects in the two clusters. In the meta-graph, each cluster is represented by a hyperedge. MCLA groups and collapses related hyperedges, and it assigns an object to the collapsed hyperedge to which it participates most strongly. MCLA also uses METIS [14] to partition the meta-graph.

Fern and Brodley [16] introduced HBGF to integrate base clusterings. HBGF constructs a bipartite graph that models both objects and clusters simultaneously as vertices. In the bipartite graph, an object is connected to several clusters, whereas there is no edge between pairwise objects or pairwise clusters. HBGF applies two graph partitioning algorithms: spectral graph partitioning [17], [18] and METIS to get the final partitions. To handle missing values, Wang, Shan and Banerjee proposed Bayesian cluster ensembles (BCE) [19], which considers all base clustering results as feature vectors, and then learns a Bayesian mixed-membership model from this feature representation.

Domeniconi and Al-Razgan [3], [4] combined the clustering ensemble framework with subspace clustering. A subspace clustering is a collection of weighted clusters, where each cluster has a weight vector representing the relevance of features for that cluster. The input to the consensus function is a collection of subspace clusterings. The subspace clustering method used is Locally Adaptive Clustering (LAC) [20], and the resulting clustering techniques are WSPA and WBPA. Li and Ding [21] specified different weights for different clusterings and proposed an approach called Weighted Consensus Clustering (WCC). The optimal weights are sought iteratively through optimization and then used to compute the consensus result within the framework of nonnegative matrix factorization (NMF) [5].

More recently, a method called ensemble clustering by matrix completion (ECMC) was proposed [22]. ECMC uses the *reliable* pair of objects to construct a partially observed co-association matrix, and exploits the matrix completion algorithm to replenish the missing entries of the co-association matrix. Two objects are reliable if they are often clustered together or seldom clustered together. ECMC then uses spectral clustering on the completed matrix to get the final clustering. However, ECMC has two disadvantages: (i) the notion of reliability between objects is hard to define, and (ii) the matrix completion process may result in information loss.

All the aforementioned methods do not assign weights to objects. In contrast, our proposed WOEC algorithms use the information provided by the base clusterings to define objects' weights which reflect how difficult it is to cluster them. The weights are then embedded within the consensus function. Specifically, we propose three different WOEC approaches: (i) *Weighted-Object Meta Clustering* (WOMC), (ii) *Weighted-Object Similarity Partition* (WOSP) Clustering, and (iii) *Weighted-Object Hybrid Bipartite* (WOHB) Graph Partition Clustering.

III. WEIGHTED OBJECT ENSEMBLE CLUSTERING

This section introduces the ensemble clustering problem and presents the details of our weighted-object ensemble clustering.

A. Ensemble Clustering Problem Formulation

An ensemble clustering approach consists of two steps. First, multiple base clusterings are generated; second, the base clusterings are consolidated into the final clustering result. We assume here that the base clusterings have already been generated, and we only discuss hard clustering; however, the methods proposed can be extended to solve soft clustering problems as well.

Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ denote the data set and $\mathbf{X} = [\mathbf{x}_1^T; \mathbf{x}_2^T; \dots; \mathbf{x}_n^T] \in \mathbb{R}^{n \times d}$ be its matricial representation, where n is the number of objects and d is the dimensionality of each object $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$. Lets consider a set of R clustering solutions $C = \{C^1, C^2, \dots, C^R\}$, where each clustering component $C^r = \{C_{k_1}^r, C_{k_2}^r, \dots, C_{k_r}^r\}$, $r = 1, 2, \dots, R$, partitions the data set \mathcal{X} into k_r disjoint clusters, i.e. $C_i^r \cap C_j^r = \emptyset$ ($\forall i \neq j, i, j = 1, 2, \dots, k_r$), and $\cup_{k=1}^{k_r} C_k^r = \mathcal{X}$. The ensemble clustering problem consists in defining a consensus function Γ that maps the set of base clusterings C into a single consolidated clustering C^* .

B. One-shot Weight Assignment

Boosting [10], [11] is a supervised technique which seeks to create a strong learner based on a set of weak learners. Adaboost (short for Adaptive Boosting) [23] is the most commonly used boosting algorithm. It iteratively generates a distribution over the data, which is then used to train the next weak classifier. In each run, objects that are hard to be classified gain more weight, while easy-to-classify objects lose weight. The newly trained classifier focuses on the points with larger weights. At termination, Adaboost combines the weak classifiers into a strong one.

The effectiveness of boosting has been proven both theoretically and experimentally. Boosting algorithms iteratively assign weights to objects using label information, which is likely to be unknown in clustering applications. This makes difficult to apply boosting techniques to clustering, and explains why little research has been performed in the context of weighted-object ensemble clustering methods. This paper aims at reducing this gap. Our goal is to enable difficult-to-cluster points to play a bigger role when producing the final consolidated clustering result. Following this objective, we propose a one-shot weight assignment to objects using the results of all base clusterings, and then embed the weights into the successive consensus clustering process. Similar to boosting, points that are hard to cluster receive larger weights, while easy-to-cluster points are given smaller weights. The difference is that boosting is an iterative process, while our weight assignment scheme is performed in one-shot. The details are given below.

Let A be the $n \times n$ co-association matrix built from the clustering solutions C :

$$A_{ij} = \frac{V_{ij}}{R} \quad (1)$$

where V_{ij} is the number of times objects \mathbf{x}_i and \mathbf{x}_j co-occur in the same cluster, and $R = |C|$ is the ensemble size. Obviously, $A_{ij} \in [0, 1]$. We set $A_{ii} = 1, \forall i = 1, 2, \dots, n$. $A_{ij} \approx 1$ means that \mathbf{x}_i and \mathbf{x}_j are often placed in the same cluster. $A_{ij} \approx 0$ means that \mathbf{x}_i and \mathbf{x}_j are often placed in different clusters. In both cases, the base clusterings show a high level of agreement. When $A_{ij} \approx 0.5$, roughly half of the clusterings group \mathbf{x}_i and \mathbf{x}_j together, and the other half place them in different clusters. This scenario is the most uncertain one, since the clustering components show no agreement on how to cluster points \mathbf{x}_i and \mathbf{x}_j . We can capture this trend by mapping the A_{ij} values through a quadratic function: $y(x) = x(1-x), x \in [0, 1]$. This function achieves the peak at $x = 0.5$, and the minima at $x = 0$ and $x = 1$, as illustrated in Fig. 1¹. Hence, we can measure

¹Other functions satisfying these properties can be used as well.

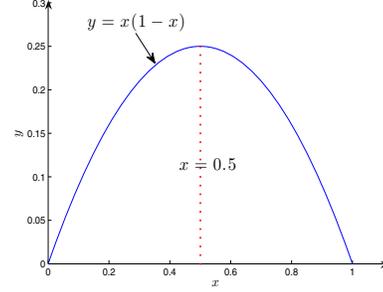


Fig. 1. Quadratic function: $y(x) = x(1-x)$

the level of uncertainty in clustering two points \mathbf{x}_i and \mathbf{x}_j as follows:

$$\text{confusion}(\mathbf{x}_i, \mathbf{x}_j) = A_{ij}(1 - A_{ij}) \quad (2)$$

The confusion index reaches its maximum of 0.25 when $A_{ij} = 0.5$, and its minimum of 0 when $A_{ij} = 0$ or $A_{ij} = 1$. We use this confusion measure to define the weight associated with each object as follows:

$$w'_i = \frac{4}{n} \sum_{j=1}^n \text{confusion}(\mathbf{x}_i, \mathbf{x}_j) \quad (3)$$

The normalization factor $\frac{4}{n}$ is used to guarantee that $w'_i \in [0, 1]$. To avoid a value of 0 for a weight, which can lead to instability, we add a smoothing term:

$$w_i = \frac{w'_i + e}{1 + e} \quad (4)$$

where e is a small positive number ($e = 0.01$ in our experiments). As a result, $w_i \in (0, 1]$. A large w_i value means that $\text{confusion}(\mathbf{x}_i, \mathbf{x}_j)$ is large for different \mathbf{x}_j values. As such it measures how hard it is to cluster \mathbf{x}_i . The assignment of large weights to points that are hard to cluster is consistent with the way boosting techniques operate [10], [11].

C. Algorithms

In this section we introduce three different *Weighted-Object Ensemble Clustering* (WOEC) algorithms. They all make use of the weights associated with objects (computed as explained above) to determine the consensus clustering. The overall process is shown in Fig. 2. As illustrated in the Figure, the consensus functions of WOSP and WOHB make use of the feature vectors, while WOMC does not.

1) Weighted-Object Meta Clustering Algorithm (WOMC):

The first approach we introduce is a weighted version of the Meta Clustering Algorithm (MCLA) introduced in [2]. We call the resulting algorithm *Weighted-Object Meta Clustering* (WOMC). The key step of meta clustering algorithms is the clustering of clusters. When measuring the similarity between two clusters, MCLA treats all the points present in both clusters equally. In contrast, the proposed WOMC technique distinguishes the contribution of hard-to-cluster and easy-to-cluster points. Specifically, a point in the intersection of two clusters contributes to the clusters' similarity in a way that is proportional to its weight. The details of the approach follow.

We explicit the components of each C^r in C . This gives: $C = \{C_{k_1}^1, \dots, C_{k_1}^1, C_{k_2}^1, \dots, C_{k_2}^1, \dots, C_{k_R}^R, \dots, C_{k_R}^R\}$, where each

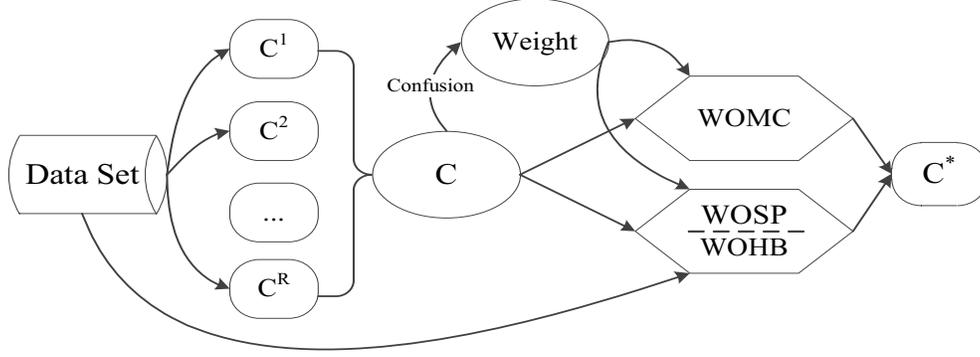


Fig. 2. The process map of Weighted-object Ensemble Clustering (WOEC)

component now corresponds to a cluster. The WOMC algorithm groups the clusters in C . To this end, it proceeds as follows. It constructs an undirected meta-graph $G = (V, E)$ with $|V| = n_c = \sum_{r=1}^R k_r$ vertices, each representing a cluster in C . The edge E_{ij} connecting vertices V_i and V_j is assigned the weight value S_{ij} defined as follows:

$$S_{ij} = \frac{\sum_{x_k \in C_i \cap C_j} w_k}{\sum_{x_k \in C_i \cup C_j} w_k} \quad (5)$$

Clearly, $S_{ij} \in [0, 1]$. If $S_{ij} = 0$, there is no edge between C_i and C_j in G . $S_{ij} = 1$ indicates that C_i and C_j contain the same points. WOMC partitions the meta-graph G into k^* (number of clusters in the final clustering result) meta-clusters, each representing a group of clusters, by using the similarity measure S_{ij} and by applying the well-known graph partitioning algorithm METIS [14]. Similarly to MCLA, WOMC also assigns each point to the meta-cluster in which it participates most strongly. Specifically, a data point may occur in different meta-clusters, and it is associated to a given meta-cluster according to the ratio of clusters it belongs to. A point is eventually assigned to the meta-cluster with the highest ratio for that point. Ties are broken randomly. There might be a situation in which no object is assigned to a meta-cluster. Thus, the final combined clustering C^* may contain less than k^* clusters.

The similarity measure defined in Eq. (5) computes the similarity of two clusters, not only based on the fraction of shared points, but also on the values of the weights associated to the points. In contrast, the binary Jaccard measure $\frac{|C_i \cap C_j|}{|C_i \cup C_j|}$ computes the similarity as the proportion between the size of the intersection and the size of the union of C_i and C_j . Suppose C_i and C_j have fixed points assigned to them. Thus, their Jaccard similarity is also fixed. Let's see how changing the weights assigned to the points affects the value of S_{ij} . Consider $D = (C_i \cup C_j) \setminus (C_i \cap C_j)$, i.e. the set of points not in the intersection. Let the weights assigned to the points in D be fixed. Then, increasing (decreasing) the weights assigned to the points in $(C_i \cap C_j)$ will cause an increase (decrease) of S_{ij} . As such, the more hard-to-cluster points C_i and C_j share, the more similar they are. Let's now fix the weights assigned to the points in $(C_i \cap C_j)$. Increasing (decreasing) the weights assigned to the points in D will cause a decrease (increase) of S_{ij} . As a consequence, the more hard-to-cluster points C_i and C_j do not share, the smaller their similarity is. **Algorithm 1** describes the WOMC algorithm.

Algorithm 1 Weighted-Object Meta Clustering Algorithm (WOMC)

Input:

- Set of base clusterings C ;
- Graph partitioning algorithm METIS;
- Number of clusters k^* .

Output:

The final consensus clustering C^* .

- 1: Assign weights to objects using the one-shot weight assignment.
 - 2: Construct the meta-graph $G = (V, E)$, where the weights associated to the edges are computed using Eq. (5).
 - 3: $MetaClusters = METIS(G, k^*)$. //Partition the meta-graph into k^* meta-clusters.
 - 4: $C^* = Assign(MetaClusters)$ //Assign each object to the meta-cluster in which it participates most strongly.
 - 5: **return** C^* .
-

2) *Weighted-Object Similarity Partitioning Algorithm (WOSP)*: Previous work on clustering [6], [7], [8] suggested to assign large weights to objects that are hard to be clustered, and iteratively move cluster centers towards areas that contain objects with large weights. Inspired by this work, we proceed similarly while leveraging the information provided by the ensemble to estimate the weights. The motivation stems from boosting. The difference is that clustering methods combined with a boosting technique assign weights to objects and move cluster centers iteratively, while we only move centers once for each base clustering by making use of the original data and objects' weight information. Clustering with a boosting technique aims to generate a better single clustering result while we seek to find a better consolidated consensus solution. To this end, for each cluster in the base clusterings C^r ($r = 1, 2, \dots, R$), we compute the corresponding weighted center as follows:

$$\mathbf{m}_l^r = \frac{\sum_{\mathbf{x}_i \in C_l^r} w_i \mathbf{x}_i}{\sum_{\mathbf{x}_i \in C_l^r} w_i} \quad (6)$$

where $l = 1, \dots, k_r$. The basic idea behind this step is to move the center of each cluster towards the region consisting of the objects that are hard to cluster. We call it *shifting center technique*. The experiments presented in Section V demonstrate the rationale and the improvement achieved by such transformation. The similarity between a point \mathbf{x}_i and a

weighted center \mathbf{m}_l^r is calculated using the following exponential function:

$$d(\mathbf{x}_i, \mathbf{m}_l^r) = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{m}_l^r\|^2}{t}\right\} \quad (7)$$

The probability of cluster C_l^r , given \mathbf{x}_i , can then be defined as:

$$P(C_l^r|\mathbf{x}_i) = \frac{d(\mathbf{x}_i, \mathbf{m}_l^r)}{\sum_{l'=1}^{k_r} d(\mathbf{x}_i, \mathbf{m}_{l'}^r)} \quad (8)$$

The smaller $\|\mathbf{x}_i - \mathbf{m}_l^r\|^2$ is, the larger $P(C_l^r|\mathbf{x}_i)$ will be. We can now define the vector P_i^r of posterior probabilities associated with \mathbf{x}_i :

$$P_i^r = (P(C_1^r|\mathbf{x}_i), P(C_2^r|\mathbf{x}_i), \dots, P(C_{k_r}^r|\mathbf{x}_i)) \quad (9)$$

where $\sum_{l=1}^{k_r} P(C_l^r|\mathbf{x}_i) = 1$. P_i^r provides a new representation of \mathbf{x}_i in a space of relative coordinates with respect to cluster centroids, where each dimension corresponds to one cluster. This new representation embeds information from both the original input data and the clustering ensemble. We use the cosine similarity to measure the similarity S_{ij}^r between \mathbf{x}_i and \mathbf{x}_j with respect to the base clustering C^r :

$$S_{ij}^r = \frac{P_i^r (P_j^r)^T}{\|P_i^r\| \|P_j^r\|} \quad (10)$$

where T denotes the transpose of a vector. There are R similarity matrices $S^r (r = 1, \dots, R)$, each one corresponding to a base clustering. We combine these matrices into one final similarity matrix S :

$$S = \frac{1}{R} \sum_{r=1}^R S^r \quad (11)$$

S represents the average similarity between \mathbf{x}_i and \mathbf{x}_j (through vectors P_i and P_j), across the R contributing clusterings. Hence we can construct an undirected graph $G = (V, E)$, where $|V| = n$ and each vertex represents an object in \mathcal{X} . The edge E_{ij} connecting vertices V_i and V_j is assigned the weight value S_{ij} . A graph partitioning algorithm (METIS in our experiments) can be applied to graph G to compute the partitioning of the n vertices that minimizes the edge weight-cut. This gives the consensus clustering we seek. The WOSP algorithm is illustrated in **Algorithm 2**.

Algorithm 2 Weighted-Object Similarity Partitioning Algorithm (WOSP)

Input:

- Data set \mathcal{X} ;
- Set of base clusterings C ;
- Graph partitioning algorithm METIS;
- Number of clusters k^* .

Output:

- The final consensus clustering C^* .
 - 1: Assign weights to objects using the one-shot weight assignment.
 - 2: Construct the corresponding graph $G = (V, E)$, with weights computed through Eqs. (6)-(11).
 - 3: $C^* = \text{METIS}(G, k^*)$. //Partition the graph into k^* parts, each representing a final cluster.
 - 4: **return** C^* .
-

3) *Weighted-Object Hybrid Bipartite Graph Partitioning Algorithm (WOHB)*: WOSP measures pairwise similarities which are solely instance-based, thus ignoring cluster similarities. We now introduce our third technique, WOHB, which attempts to capture both kinds of similarities. This approach reduces the clustering consensus problem to a bipartite graph partitioning problem, which partitions both cluster vertices and instance vertices simultaneously. Thus, it also accounts for similarities between clusters. As WOSP, WOHB makes use of the shifting center technique and computes probabilities as in Eq. (8).

WOHB constructs an undirected hybrid bipartite graph $G = (V, E)$, where V contains $n_c + n$ vertices ($n_c = \sum_{r=1}^R k_r$). The first n_c vertices represent all the clusters in C and the last n vertices represent objects. The edge E_{ij} connecting the vertices V_i and V_j is assigned the weight value S_{ij} defined as follows:

$$S_{ij} = \begin{cases} S_{ji} = 0, & \text{if both vertices } i \text{ and } j \text{ are} \\ & \text{clusters or objects} \\ S_{ji} = P(C_j|\mathbf{x}_i) & \end{cases} \quad (12)$$

where $P(C_j|\mathbf{x}_i)$ is the probability of cluster C_j given \mathbf{x}_i , and can be computed from Eq. (8). The matrix S of elements S_{ij} can be written as $S = \begin{bmatrix} 0 & B^T \\ B & 0 \end{bmatrix}$, where the $n \times n_c$ matrix B is defined as follows:

$$B = \begin{bmatrix} P_1^1 & P_1^2 & \dots & P_1^R \\ P_2^1 & P_2^2 & \dots & P_2^R \\ \dots & \dots & \dots & \dots \\ P_n^1 & P_n^2 & \dots & P_n^R \end{bmatrix} \quad (13)$$

where P_i^r is a row vector defined in Eq. (9).

A graph partitioning algorithm (METIS in our experiments) is then used to partition this graph into k^* hybrid parts (each containing objects and clusters), so that the edge weight-cut is minimized. The partition of the objects provides the final clustering result. The WOHB algorithm is given in **Algorithm 3**.

Algorithm 3 Weighted-Object Hybrid Bipartite Graph Partitioning Algorithm (WOHB)

Input:

- Data set \mathcal{X} ;
- Set of base clusterings C ;
- Graph partitioning algorithm METIS;
- Number of clusters k^* .

Output:

- The final consensus clustering C^* .
 - 1: Assign weights to objects using the one-shot weight assignment.
 - 2: Construct the hybrid bipartite graph $G = (V, E)$ as described in this subsection.
 - 3: $C_{\text{hybrid}} = \text{METIS}(G, k^*)$. //Partition the graph into k^* parts.
 - 4: **return** the partition of objects in C_{hybrid} as C^* .
-

D. Weighted-Object k-means

To verify the effect of moving the center of each cluster according to Eq. (6), we introduce here a modified version of

Algorithm 4 Weighted-Object k -means (WOKmeans)

Input:

Data set \mathcal{X} ;
Set of base clusterings C ;
Number of clusters k^* .

Output:

The final clustering C^* .

- 1: Assign weights to objects using the one-shot weight assignment.
- 2: Randomly generate k^* weighted cluster centers $\mathbf{c}_1, \dots, \mathbf{c}_{k^*}$, each corresponding to a cluster $C_k = \emptyset$ ($k = 1, \dots, k^*$).
- 3: **repeat**
- 4: //assign each object to its closest weighted center
 $\forall i, C_{\bar{k}} = C_k \cup \mathbf{x}_i$, where $\bar{k} = \arg \min_k \|\mathbf{x}_i - \mathbf{c}_k\|$.
- 5: //generate new weighted centers

$$\mathbf{c}_k = \frac{\sum_{\mathbf{x}_i \in C_k} w_i \mathbf{x}_i}{\sum_{\mathbf{x}_i \in C_k} w_i}, k = 1, \dots, k^* \quad (14)$$

- 6: **until** All weighted centers do not change or reach the maximum of iteration
 - 7: **return** $C^* = \{C_1, C_2, \dots, C_{k^*}\}$.
-

k -means that makes use of such weighted means. The resulting technique is called Weighted-Object k -means (WOKmeans), and it's illustrated in **Algorithm 4**. WOKmeans computes the weighted centers after assigning objects to clusters in each iteration, as shown in Eq. (14). The main idea is to move cluster centers towards regions with larger weights iteratively. Compared to the weighted version of k -means in [9], WOKmeans computes all objects' weights in advance using the co-association matrix provided by the base clusterings, while [9] assigns weights to objects iteratively.

IV. EMPIRICAL EVALUATION

A. Data Sets

We conducted experiments on one simulated data and eight UCI² data sets to compare the performance of the WOEC methods against existing state-of-the-art techniques. A description of the data sets used is given in Table I. The simulated data is discussed in V-A.

The Satimage data set originally contained 6435 objects; we randomly selected 420 images equally distributed among the six classes for our experiments. The Spambase data set had 4601 objects; we randomly sampled 500 objects (250 for each class). The other six data sets were used in their original form. For each data set, each feature is preprocessed to have zero mean value and unit variance.

TABLE I. DATA SETS USED IN THE EXPERIMENTS

Data sets	#Objects	#Dimensions	#Classes
Simulated data	300	2	2
AustralianCredit	690	14	2
BalanceScale	625	4	3
BreastTissue	106	9	6
Glass	214	9	6
HayesRoth	132	4	3
Iris	150	4	3
Satimage	420	36	6
Spambase	500	57	2

²<http://archive.ics.uci.edu/ml/index.html>

B. Evaluation Measures

There exist various metrics to evaluate clustering results [24], [25]. Since the labels of the data are known, we use the Rand Index (RI) [26], the Adjusted Rand Index (ARI) [26], and the Normalized Mutual Information (NMI) [2] as the validity indices. Let $C^* = \{C_1^*, C_2^*, \dots, C_{k^*}^*\}$ be the consensus clustering result (k^* is the number of clusters in the final consensus clustering), and let $\bar{C} = \{\bar{C}_1, \bar{C}_2, \dots, \bar{C}_{\bar{k}}\}$ be the ground-truth clustering (\bar{k} is the number of classes).

- 1) RI measures the similarity between C^* and \bar{C} as follows:

$$RI(C^*, \bar{C}) = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}} \quad (15)$$

where

- a is the number of pairs of points that are in the same cluster in both C^* and \bar{C} .
- b is the number of pairs of points that are in different clusters in both C^* and \bar{C} .
- c is the number of pairs of points that are in the same cluster in C^* and in different clusters in \bar{C} .
- d is the number of pairs of points that are in different clusters in C^* and in the same cluster in \bar{C} .

RI ranges from 0 to 1.

- 2) ARI defines the similarity between C^* and \bar{C} as follows:

$$ARI(C^*, \bar{C}) = \frac{\sum_{i,j} \binom{n_{ij}}{2} - (S^* \times \bar{S}) / \binom{n}{2}}{\frac{1}{2}(S^* + \bar{S}) - (S^* \times \bar{S}) / \binom{n}{2}} \quad (16)$$

where $S^* = \sum_i \binom{n_i^*}{2}$, $\bar{S} = \sum_j \binom{\bar{n}_j}{2}$, and n_i^* and \bar{n}_j denote the number of objects in cluster C_i^* and \bar{C}_j ($i = 1, \dots, k^*$; $j = 1, \dots, \bar{k}$), respectively. $n_{ij} = |C_i^* \cap \bar{C}_j|$. The ARI values range from -1 to +1.

3) NMI [2] treats the clustering labels of C^* and \bar{C} as two random variables. The mutual information of these two random variables is evaluated and then normalized within the $[0, 1]$ range.

A larger value of RI (ARI or NMI) means that C^* and \bar{C} are more similar.

C. Base Clusterings

To generate diverse clusterings we used k -means. Random sampling of the data and feature subset selection techniques were applied. Specifically, for the first half of the base clusterings, in each run we randomly selected 70% of the objects and performed k -means on this subset of the data. The remaining objects were assigned to the closest cluster based on the Euclidean distances between the object and the cluster centers. For the second half of the base clusterings, in each run we selected 70% of the features and conducted k -means in the corresponding subspace. When using k -means, we randomly generated initial centers in each independent run.

V. RESULTS AND ANALYSIS

For a fair comparison, we used the METIS package [14] for all the clustering ensemble methods that need to partition a graph. The number of clusters k^* in the consensus clustering is set equal to the number of classes, for all data and all

TABLE II. RESULTS ON SIMULATED DATA

Data	Index	min	max	mean	CSPA	MCLA	HBGF	BCE	WCC	ECMC	WOEC		
											WOMC	WOSP	WOHB
Simulated Data	RI	0.5000	0.9854	0.8769	0.9736	0.9743	0.9762	0.9261	0.9736	0.9762	0.9743	0.9828	0.9828
	ARI	0.0000	0.9708	0.7537	0.9472	0.9485	0.9524	0.8525	0.9472	0.9524	0.9485	0.9656	0.9656
	NMI	0.0024	0.9449	0.7185	0.8978	0.9023	0.9082	0.8103	0.9003	0.9082	0.9023	0.9289	0.9294

TABLE III. COMPARISON BETWEEN k -MEANS AND WOKMEANS ON SIMULATED DATA

Data	Index	k -means	WOKmeans
Simulated Data	RI	0.8329±0.2161	0.8825±0.1888
	ARI	0.6657±0.4322	0.7651±0.3776
	NMI	0.6329±0.4106	0.7301±0.3611

methods³. In our experiments, the value of t (see Eq. (7)), for each base clustering, is set to the average of the squared Euclidean distances between the data and the weighted means. All the experimental results are the average of 100 independent runs. We also performed a paired t -test to assess the statistical significance of the results; the significance level was always set to 0.05. We compare the proposed WOEC methods against eight state-of-the-art ensemble clustering algorithms, namely: CSPA [2], MCLA [2], HBGF [16], BCE [19], WCC [21], ECMC [22], WSPA[3], [4] and WBPA[3], [4]. (HGPA [2] was not included in our experiments since it typically performs worse than CSPA and MCLA.)

A. Simulated Data

To illustrate the effectiveness of the proposed three WOEC algorithms and of WOKmeans, we first generated a simulated data. The simulated data consists of $d = 2$ input features and 2 classes, each consisting of 150 points. Both classes are normally distributed according to multivariate Gaussian distributions, as shown in Fig. 3. The mean vector and the covariance matrix of class 1 are $(5, 10)$ and $\begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$, respectively. For class 2, the mean vector is $(20, 10)$ and the covariance matrix is again $\begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$. We set the ensemble size $R = 40$. RI, ARI and NMI are applied to assess the performance of all algorithms. For each of the 40 base clusterings, the data sampling technique was used. That is, we randomly selected 70% of the points and performed k -means on this subset of the data. The remaining points were assigned to the closest cluster based on the Euclidean distance between the points and the cluster centers. Each feature value was also restricted to have zero mean and unit variance. For each independent run, we record the minimum (min), maximum (max), and mean value (of RI, ARI and NMI) for all the base clusterings. The corresponding averages of 100 runs are also reported. The results are shown in Table II. Significantly better values are given in boldface.

WOSP and WOHB achieve the best performance for all the evaluation indices (RI, ARI, and NMI). This is because the hard-to-cluster points (i.e. the points between the clusters' boundaries) have the largest contribution to the shifting center procedure. This causes a shift of the clusters' centers towards the region with hard-to-cluster points. As such, the influence of

³In real applications, classes may be multimodal, and thus k^* should be larger than the number of classes. In other cases, there may be less clusters than classes. These scenarios are not considered in this paper and we simply set k^* equal to the number of classes.

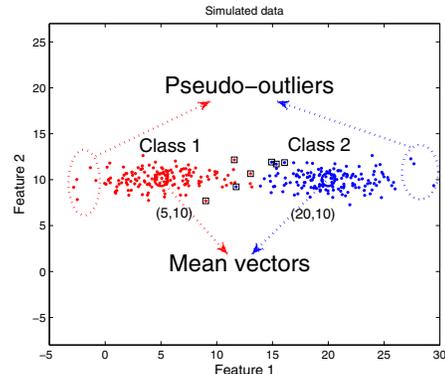


Fig. 3. Simulated Data

the pseudo-outliers⁴ is reduced, thereby leading to more robust ensemble clustering results. BCE performs the worst. WOMC achieves comparable performance to the other methods, but loses to WOSP and WOHB, showing that the technique of shifting centers can work better on this data.

To further demonstrate the effectiveness of shifting centers, we conducted experiments on the simulated data to compare the performance of k -means and WOKmeans. The experimental setup is the same as before. Table III shows the results. Clearly, WOKmeans achieves a considerable improvement with respect to k -means and its standard deviations are also smaller. k -means is negatively affected by the pseudo-outliers. Although WOKmeans improves upon k -means, its performance is considerably worse than that of ensemble clustering methods, demonstrating the effectiveness of ensemble techniques. Points that receive the largest weights are highlighted in squares in Fig. 3, showing that between-cluster points tend to gain large weights. From Table II and III, we can see that generally a better RI value corresponds to a better ARI and NMI values. Therefore, we only report the ARI values for the rest of the experiments due to the limitation in space.

B. Evaluation of the WOEC Techniques

We conducted experiments to evaluate the effectiveness of the proposed WOEC methods on eight UCI data sets.

Base clusterings were generated as explained in Section IV-C. We experimented with three different ensemble sizes R : 20, 40, and 60. The results in terms of ARI are shown in Table IV. Each ARI value reported is the average of 100 independent runs. In each row, the values in boldface are the ARI values that

⁴As shown in Fig. 3, the left circled points are far away from the mean point of class 1 and the density around them is much lower than that around the other points in class 1. These points can bias the computation of the mean vector. They are generated from the same distribution as the other red points in class 1, and should be grouped in the same cluster, but behave like outliers for the purpose of this discussion. For this reason we call them "pseudo-outliers". Similarly, the points in the right circle are the pseudo-outliers of class 2.

TABLE IV. COMPARISON AGAINST EXISTING METHODS (ARI VALUES)

Data	R	min	max	mean	CSPA	MCLA	HBGF	BCE	WCC	ECMC	WOEC		
											WOMC	WOSP	WOHB
Australian (0.4189)	20	0.0104	0.5184	0.3775	0.4300	0.4661	0.4173	0.4433	0.4658	0.4658	0.4668	0.4341	0.4279
	40	-0.0069	0.5274	0.3763	0.4446	0.4674	0.4209	0.4485	0.4666	0.4668	0.4674	0.4357	0.4316
	60	-0.0075	0.5327	0.3766	0.4482	0.4676	0.4295	0.4430	0.4674	0.4660	0.4676	0.4350	0.4329
Balance (0.1336)	20	0.0573	0.2300	0.1374	0.1432	0.1348	0.1358	0.1322	0.1402	0.1331	0.1510	0.1215	0.1481
	40	0.0384	0.2531	0.1366	0.1396	0.1198	0.1264	0.1241	0.1315	0.1132	0.1322	0.1214	0.1290
	60	0.0313	0.2625	0.1365	0.1347	0.1238	0.1300	0.1360	0.1359	0.1197	0.1367	0.1181	0.1194
Breast (0.2952)	20	0.2193	0.3584	0.2850	0.3390	0.3052	0.2837	0.2893	0.3343	0.2292	0.3268	0.3340	0.3127
	40	0.1989	0.3691	0.2832	0.3393	0.3021	0.2656	0.2938	0.3370	0.2248	0.3176	0.3323	0.3130
	60	0.1972	0.3752	0.2833	0.3422	0.3046	0.2695	0.2953	0.3318	0.2207	0.3232	0.3424	0.3149
Glass (0.1680)	20	0.1149	0.2484	0.1737	0.1197	0.1422	0.1630	0.1736	0.1079	0.1333	0.1392	0.1710	0.1479
	40	0.1021	0.2642	0.1742	0.1213	0.1414	0.1497	0.1727	0.1172	0.1439	0.1338	0.1748	0.1365
	60	0.0902	0.2694	0.1739	0.1165	0.1443	0.1495	0.1715	0.1189	0.1383	0.1463	0.1755	0.1426
HayesRoth (0.0601)	20	-0.0073	0.1699	0.0703	0.0725	0.0518	0.0538	0.0539	0.0424	0.0489	0.0524	0.1470	0.0390
	40	-0.0113	0.1745	0.0697	0.0903	0.0526	0.0547	0.0492	0.0377	0.0644	0.0521	0.1617	0.0576
	60	-0.0129	0.1826	0.0699	0.0911	0.0524	0.0589	0.0456	0.0390	0.0384	0.0636	0.1636	0.0849
Iris (0.5784)	20	0.4434	0.6722	0.5979	0.6107	0.6070	0.5961	0.6053	0.6118	0.5966	0.6096	0.6314	0.6394
	40	0.4256	0.6734	0.5991	0.6149	0.6080	0.5931	0.6074	0.6125	0.6015	0.6126	0.6292	0.6263
	60	0.4217	0.6734	0.5971	0.6144	0.6084	0.5954	0.6049	0.6118	0.6023	0.6117	0.6273	0.6221
Satimage (0.6099)	20	0.4881	0.6670	0.6158	0.6382	0.6747	0.6254	0.6109	0.6481	0.6087	0.6766	0.6544	0.6218
	40	0.4383	0.6746	0.6143	0.6280	0.6752	0.6190	0.6187	0.6458	0.5692	0.6807	0.6493	0.6169
	60	0.4117	0.6778	0.6130	0.6247	0.6777	0.6323	0.6249	0.6505	0.6065	0.6768	0.6538	0.6179
Spambase (0.1412)	20	0.0001	0.5643	0.1656	0.5345	0.4058	0.5398	0.2943	0.5049	0.3584	0.4054	0.5494	0.5561
	40	0.0001	0.5761	0.1683	0.5342	0.4839	0.5320	0.3842	0.5200	0.3666	0.4825	0.5478	0.5579
	60	0.0000	0.5869	0.1673	0.5318	0.4977	0.5306	0.3557	0.4779	0.4067	0.5027	0.5502	0.5593

are statistically significantly better than the remaining ones. We also run k -means 100 times for baseline comparison; its average ARI is given in Table IV in parenthesis, following the name of each data set. For each independent run, we record the minimum, maximum and mean ARI for all the base clusterings. The corresponding averages are also reported in the Table. We can see that the mean ARI of the base clusterings is very close to the baseline for each data set.

From Table IV we can see that at least one of the WOEC methods can always achieve the best performance on all data sets, except for $R = 20$ and $R = 40$ on BreastTissue. To be more specific, WOMC gives the best performance on AustralianCredit, BalanceScale, and Satimage; WOSP achieves the best performance on Glass and HayesRoth; while WOHB significantly outperforms all other methods on Spambase. On Iris data, WOSP and WOHB achieve comparable performance and both outperform all other methods. As expected, the ensemble clustering methods achieve superior performance with respect to the baseline for most of the problems. In particular, on the Spambase data set, the ensemble techniques provide a considerable improvement. The poor performance of the baseline may be due to the large dimensionality of the data in this case. There are exceptions to this general trend. For instance, most of the ensemble clustering methods are inferior to the baseline on BalanceScale, Glass, and HayesRoth. Nevertheless, at least one of the WOEC methods performs well also for these data sets, i.e. WOMC on BalanceScale, and WOSP on Glass and HayesRoth. An interesting result is that the performance of WOSP and WOHB is close to the max index on HayesRoth and Spambase respectively, and WOMC can achieve better ARI values than the max index on Satimage.

As pointed out earlier, our WOMC technique is a variation of MCLA. We can see that these two methods have comparable performance on AustralianCredit, Glass, HayesRoth, and Spambase, while WOMC outperforms MCLA on the other 4 data sets. This shows the benefits of considering objects' weights when measuring the similarity of two meta clusters in a meta clustering algorithm. Compared against the most recent ECMC technique, the WOEC methods can achieve much better

results in general. The main reason is that ECMC assigns a value of one to the entries of the co-association matrix that are larger than a threshold d_1 , and a value of zero to the entries that are smaller than a threshold value d_0 . This leads to information loss; in particular, the loss of information regarding the hard-to-cluster objects. In addition, the setting of thresholds like d_0 and d_1 is always problematic. In our experiments we set $d_0 = 0.2$ and $d_1 = 0.8$ as suggested in [22]. ECMC needs also to choose a suitable value for a parameter C that minimizes $1^T M 1$, where M is a pairwise similarity matrix (details are in [22]), which is time consuming. In contrast, WOMC does not need to perform any parameter selection, while WOSP and WOHB are robust to the setting of its parameter t , as the analysis in Section V-E shows.

C. Comparison against WSPA and WBPA

To perform comparisons against WSPA and WBPA, we generated the base clusterings using the LAC algorithm, as described in [3], [4]. LAC assigns weights to features, locally at each cluster. Both WSPA and WBPA require such weights, while our WOEC techniques have a wider applicability. The weight vectors computed by the LAC algorithm are discarded when combined with the WOEC approaches. Results are shown in Table V. As before, statistically significant results are in boldface. We can see that the WOEC methods can always achieve the best performance, while WSPA and WBPA give comparable to best results only on BreastTissue and Glass data sets.

D. Evaluation of WOKmeans

In Section V-A, we have discussed the superiority of WOKmeans with respect to k -means on the simulated data. To further investigate the behavior of WOKmeans, in this section we add more comparisons on three data sets: BreastTissue, Glass, and Iris. The reported results are the average of 100 individual runs. In each run, the maximum number of iterations of both methods is set to 100. The number of clusters is set to the number of classes given by the ground truth labels. The ensemble sizes are again 20, 40 and 60. Note that only WOKmeans is affected by the ensemble size. Table VI gives

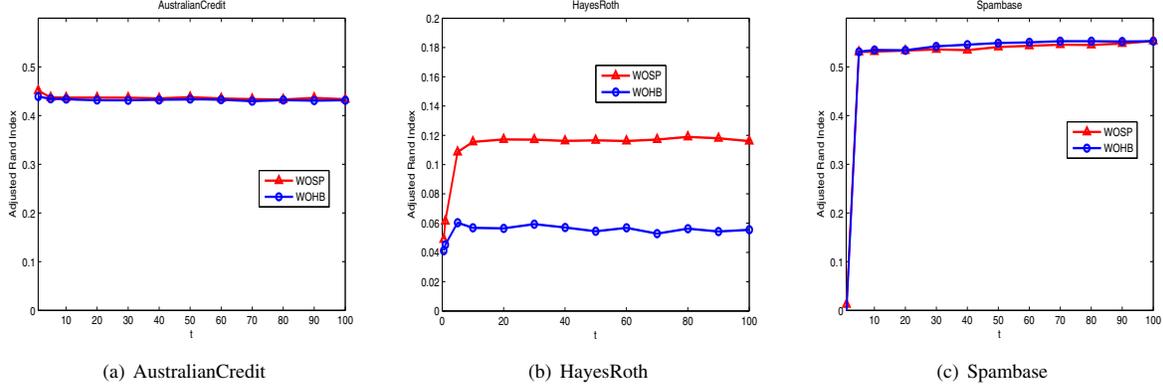


Fig. 4. Sensitivity analysis of parameter t

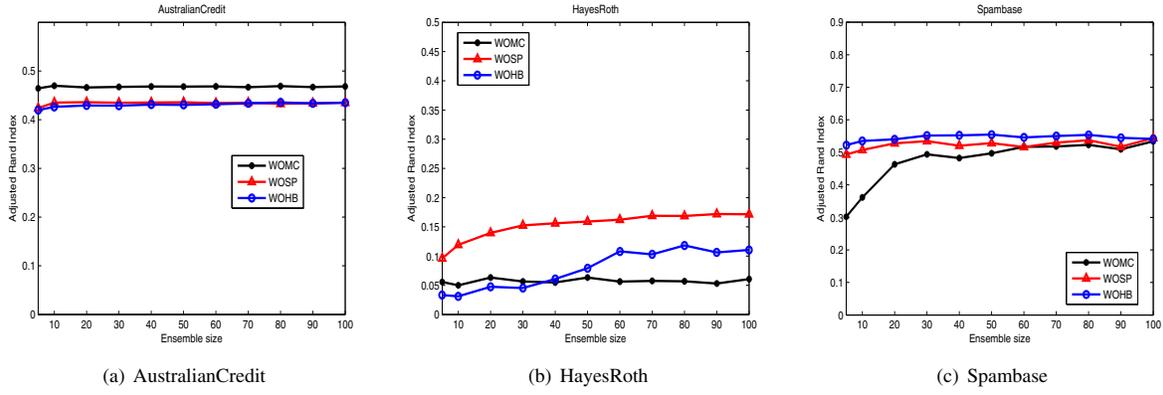


Fig. 5. Sensitivity analysis of ensemble size R

TABLE V. COMPARISON AGAINST WSPA AND WBPA (ARI)

Data	R	WSPA	WBPA	WOC		
				WOMC	WOSP	WOHB
Australian	20	0.4196	0.4172	0.4662	0.4015	0.4036
	40	0.4238	0.4203	0.4666	0.4220	0.4234
	60	0.4228	0.4202	0.4667	0.4286	0.4310
Balance	20	0.1213	0.1272	0.1193	0.1215	0.1454
	40	0.1270	0.0392	0.1427	0.1185	0.1203
	60	0.1241	0.1058	0.1331	0.1199	0.1177
Breast	20	0.2941	0.3275	0.3377	0.3273	0.3187
	40	0.2931	0.2936	0.3276	0.3266	0.3029
	60	0.2940	0.2628	0.3272	0.3259	0.3102
Glass	20	0.1762	0.1508	0.1445	0.1854	0.1357
	40	0.1788	0.1341	0.1433	0.1836	0.1386
	60	0.1807	0.1481	0.1471	0.1808	0.1435
HayesRoth	20	0.0286	0.0276	0.0422	0.1121	0.0311
	40	0.0244	0.0371	0.0352	0.1210	0.0372
	60	0.0303	0.0515	0.0315	0.1467	0.0505
Iris	20	0.6385	0.6368	0.6183	0.6430	0.6305
	40	0.6372	0.6197	0.6134	0.6440	0.6163
	60	0.6372	0.6163	0.6104	0.6496	0.6129
Satimage	20	0.6456	0.5979	0.6578	0.6606	0.6164
	40	0.6447	0.6375	0.6677	0.6608	0.6139
	60	0.6465	0.6454	0.6740	0.6495	0.6179
Spambase	20	0.5362	0.5248	0.5006	0.5436	0.5500
	40	0.5341	0.5181	0.5204	0.5600	0.5666
	60	0.5349	0.5062	0.5107	0.5629	0.5736

the results. We observe that WOKmeans can significantly outperform k -means on the three data sets in all cases. For Iris data, WOKmeans not only achieves a better performance, but it's also more stable with much smaller standard deviations.

TABLE VI. EXPERIMENTAL RESULTS OF k -MEANS AND WOKMEANS (ARI)

Data	R	k -means	WOKmeans
BreastTissue	20		0.3163±0.0375
	40	0.2952±0.0305	0.3144±0.0359
	60		0.3187±0.0365
Glass	20		0.1792±0.0350
	40	0.1680±0.0288	0.1779±0.0329
	60		0.1759±0.0320
Iris	20		0.6092±0.0294
	40	0.5784±0.0776	0.6109±0.0267
	60		0.6099±0.0329

These results empirically demonstrate again the effect of Eq. (6) on the resulting clusters. As suggested by previous analysis [6], [7], [8], moving the centers towards the hard-to-cluster data can improve the clustering of the same.

E. Sensitivity Analysis of Parameter t

Here we test how sensitive the WOSP and the WOHB methods are with respect to the values of the parameter t in Eq. (7). We set the ensemble size $R = 40$ and change the parameter t from 0.5 to 100. We use the AustralianCredit, HayesRoth, and Spambase data sets to observe the performance of the methods for the different values of t . Fig. 4 shows the results. We can see that both WOSP and WOHB tend to be stable when $t > 5$ on HayesRoth and Spambase, while the performance is almost the same for different values of t on AustralianCredit. In fact, the probability of a cluster

C_k^r given an object \mathbf{x}_i , computed as in Eq. (8), is essentially determined by the Euclidean distances between \mathbf{x}_i and all the cluster centers in clustering C^r . Different values of t will not change the sequence of probability values. This explains why both WOSP and WOHB are robust to different values of the parameter t . Nevertheless, we should avoid setting t to values that are too small or too large. In these two cases, in fact, $\forall k, d(\mathbf{x}_i, \mathbf{m}_k^r) \rightarrow 0$ and $d(\mathbf{x}_i, \mathbf{m}_k^r) \rightarrow 1$, respectively, in Eq. (7), and this will influence the consensus clustering process of WOSP and WOHB. Therefore, the average of the squared Euclidean distances between the data and the weighted means is a reasonable default value of t for each base clustering.

F. Sensitivity Analysis of Ensemble Size R

In this section we investigate the sensitivity of the three WOEC methods with respect to the ensemble size R . We change R from 5 to 100, and we use the same three data sets: AustralianCredit, HayesRoth, and Spambase. Fig. 5 gives the results. In general, with an increase of the ensemble size, the performance of each WOEC method improves. WOEC tends to be stable when R is larger than 10, 60, and 30 on AustralianCredit, HayesRoth, and Spambase, respectively. On AustralianCredit, all three WOEC methods can achieve a good performance when $R = 5$. Also WOSP and WOHB achieve a stable performance on Spambase in correspondence of a small ensemble size. Each method gives the highest ARI for a different data set.

VI. CONCLUSIONS AND FUTURE WORK

We have introduced an approach called *Weighted-Object Ensemble Clustering* (WOEC). Three different consensus techniques have been proposed to leverage weighted objects. Experimental results indicate that each of the three techniques leads in performance in different data sets. This suggests that the three methods are most effective under different conditions. The achievement of a better understanding of such conditions is an interesting question we'll investigate. Like most existing ensemble methods, the WOEC approaches are quadratic in the number of points and linear in the dimensionality. How to apply WOEC methods effectively and efficiently with large scale and high dimensional data is also a challenging future work.

VII. ACKNOWLEDGMENTS

This paper is partially supported by grants from the Natural Science Foundation of China (Project Nos. 61003174, 61101234, 61372138), the Fundamental Research Funds for the Central Universities of China (Project Nos. XDJK2010B002, XDJK2013C123), the Doctoral Fund of Southwest University (Nos. SWU110063 and SWU113034) and the China Scholarship Council (CSC).

REFERENCES

- [1] J. Ghosh and A. Acharya, "Cluster ensembles," *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 4, pp. 305–315, 2011.
- [2] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002.
- [3] M. Al-Razgan and C. Domeniconi, "Weighted clustering ensemble," in *Proceedings of The 6th SIAM International Conference on Data Mining*, April 2006, pp. 258–269.
- [4] C. Domeniconi and M. Al-Razgan, "Weighted cluster ensembles: Methods and analysis," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 4, pp. 17:1–40, January 2009.
- [5] T. Li, C. Ding, and M. I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," in *Proceedings of the 7th IEEE International Conference on Data Mining*, 2007, pp. 577–582.
- [6] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proceedings of the 11th International Conference on Information and Knowledge Management*, 2002, pp. 600–607.
- [7] A. Topchy, B. Minaei-Bidgoli, A. K. Jain, and W. F. Punch, "Adaptive clustering ensembles," in *Proceedings of the 17th International Conference on Pattern Recognition*, August 2004, pp. 272–275.
- [8] B. Zhang, M. Hsu, and U. Dayal, "K-harmonic means—a spatial clustering algorithm with boosting," *Temporal, Spatial, and Spatio-Temporal Data Mining*, pp. 31–45, 2001.
- [9] R. Nock and F. Nielsen, "On weighting clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1223–1235, August 2006.
- [10] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.
- [11] Z. Zhou, *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall, 2012.
- [12] A. Fred and A. Jain, "Data clustering using evidence accumulation," in *Proceedings of the 16th International Conference of Pattern Recognition*, 2002, pp. 276–280.
- [13] A. Fred and A. K. Jain, "Evidence accumulation clustering based on the k-means algorithm," in *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, 2002, pp. 442–451.
- [14] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, August 1998.
- [15] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Application in vlsi domain," in *Proceedings of the Design and Automation Conference*, 1997, pp. 526–529.
- [16] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proceedings of the 21th International Conference on Machine Learning*, 2004, pp. 281–288.
- [17] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2001, pp. 849–856.
- [18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August 2000.
- [19] H. Wang, H. Shan, and A. Banerjee, "Bayesian cluster ensembles," in *Proceedings of the 9th SIAM International Conference on Data Mining*, 2009, pp. 209–220.
- [20] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma, "Subspace clustering of high dimensional data," in *Proceedings of the SIAM International Conference on Data Mining*, April 2004, pp. 517–521.
- [21] T. Li and C. Ding, "Weighted consensus clustering," in *Proceedings of the 8th SIAM International Conference on Data Mining*, 2008, pp. 798–809.
- [22] J. Yi, T. Yang, R. Jin, A. K. Jain, and M. Mahdavi, "Robust ensemble clustering by matrix completion," in *Proceedings of the 12th IEEE International Conference on Data Mining*, December 2012, pp. 1176–1181.
- [23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.
- [24] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Prez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, vol. 46, no. 1, pp. 243–256, January 2013.
- [25] C. Legny, S. Juhsz, and A. Babos, "Cluster validity measurement techniques," in *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, 2006, pp. 388–393.
- [26] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.