Knowl Inf Syst
DOI 10.1007/s10115-016-0988-y

REGULAR PAPER

Weighted-object ensemble clustering: methods and analysis

Yazhou Ren[1,2] · **Carlotta Domeniconi**[3] · **Guoji Zhang**[4] · **Guoxian Yu**[5]

Received: 6 June 2015 / Accepted: 25 August 2016

Abstract Ensemble clustering has attracted increasing attention in recent years. Its goal is to combine multiple base clusterings into a single consensus clustering of increased quality. Most of the existing ensemble clustering methods treat each base clustering and each object as equally important, while some approaches make use of weights associated with clusters, or to clusterings, when assembling the different base clusterings. Boosting algorithms developed for classification have led to the idea of considering weighted objects during the clustering process. However, not much effort has been put toward incorporating weighted objects into the consensus process. To fill this gap, in this paper, we propose a framework 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 with 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 experimentally demonstrate the gain in performance that our WOEC methodology achieves with respect to state-of-the-art ensemble clustering methods, as well as its stability and robustness.

Keywords Ensemble clustering · Consensus clustering · Graph partition · Weighted objects

✉ Yazhou Ren
yazhou.ren@uestc.edu.cn

1 School of Computer Science and Engineering, Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 611731, China

2 School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China

3 Department of Computer Science, George Mason University, Fairfax, VA 22030, USA

4 School of Sciences, South China University of Technology, Guangzhou 510640, China

5 College of Computer and Information Science, Southwest University, Chongqing 400715, China

Published online: 06 September 2016

1 Introduction

Clustering is a key step for many exploratory tasks in data mining. Clustering seeks to partition data into groups, or clusters, according to a certain similarity measure. The overall goal is to place data points that are similar to one another in the same cluster, and points that are dissimilar in different clusters. It is well known that off-the-shelf clustering methods may discover different patterns when applied to the same set of data. This is because each algorithm has its own bias due to the optimization of different criteria. An additional challenge with clustering is the absence of ground truth to validate the results.

In recent years, clustering ensembles have emerged as a technique to overcome some of the challenges related to clustering [10,25]. A clustering ensemble technique is characterized by two components: a mechanism to generate diverse partitions, and a consensus function to combine the 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.

Some work has been done to investigate 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 that affects data in high dimensional spaces [1,4]. A subspace clustering can be seen as 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 [18]. To the best of our knowledge, our work [22] was the first that investigated how to use weights associated with *objects* within the clustering ensemble framework.

Researchers have studied the benefits of weighting objects in iterative clustering methods combined with boosting techniques [11,26,30]. 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 toward 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 toward the modes of the distribution modified by the weights. Nock and Nielsen [21] formulated clustering as a constrained minimization using 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 a framework 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 with objects. Different from boosting [23], 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.

This article is a major extension of our previous work [22]. As new material, this paper provides motivations for our WOEC methodology; a variety of simulated experiments to gain a deeper understanding of the conditions under which each of the three proposed consensus techniques leads in performance; new experiments with real data; and an extensive analysis of the results.

Specifically, the main contributions of this paper are summarized as follows:

(i) We formulate a framework called *Weighted-Object Ensemble Clustering* (WOEC), which consists of a one-shot scheme to assign weights to objects based on the co-association matrix. We introduce three consensus techniques that leverage the weight information associated with objects, and a weighted version of the classic k-means clustering algorithm.
(ii) Extensive experiments on three synthetic data sets and fifteen real data sets are conducted to demonstrate the effectiveness, robustness, and stability of the WOEC approach. A comprehensive comparison between the proposed weighted-object k-means algorithm and k-means is also given.
(iii) An investigation of the conditions under which each of the proposed WOEC algorithms is expected to work well is provided via the analysis of the results on simulated data.

2 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 [31]. This result has inspired researchers to further investigate the development of ensemble techniques for unsupervised problems, namely clustering.

Fred and Jain [7,8] 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 [25] proposed three graph-based methods, namely Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph Partitioning Algorithm (HGPA), and Meta-CLustering Algorithm (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 [15] 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 [14] 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 [15] to partition the meta-graph.

Fern and Brodley [6] introduced Hybrid Bipartite Graph Formulation (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 [20,24] and METIS to get the final partitions. To handle missing values, Wang et al. [28] proposed Bayesian Cluster Ensembles (BCE), 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 [1,4] 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) [5], and the resulting clustering techniques are Weighted Similarity Partitioning Algorithm (WSPA) and Weighted Bipartite Partitioning Algorithm (WBPA). Li and Ding [17] 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) [18].

More recently, a method called Ensemble Clustering by Matrix Completion (ECMC) was proposed [29]. 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. To reduce the time and space complexity of the existing ensemble clustering methods, Liu et al. [19] proposed a spectral ensemble clustering approach, where spectral clustering is applied on the obtained co-association matrix to compute the final clustering result. For ensemble clustering of high dimensional data, the work in [13] first partitions the features into groups, and then randomly selects several features from each group to generate the component data sets.

All the methods mentioned above 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.

3 Weighted object ensemble clustering

This section introduces in detail our weighted-object ensemble clustering methodology.

3.1 Ensemble clustering problem formulation

An ensemble clustering process contains 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 $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ denote the data set , where n is the number of objects and each object $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T \in \Re^d$.

Lets consider a set of R clustering solutions $C = \{C^1, C^2, \ldots, C^R\}$, where each clustering component $C^r = \{C_1^r, C_2^r, \ldots, C_{k_r}^r\}$, $r = 1, 2, \ldots, R$, partitions the data set $\mathcal{X}$ into k_r disjoint clusters, i.e., $C_i^r \cap C_j^r = \emptyset$ $(\forall i \neq j, i, j = 1, 2, \ldots, k_r)$, 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^*.

3.2 One-shot weight assignment

Boosting [23,31] is a supervised technique which seeks to create a strong learner based on a set of weak learners. Adaboost [9] 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:

$$A_{ij} = \frac{V_{ij}}{R} \tag{1}$$

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 $A_{ii} = 1, \forall i = 1, 2, \ldots, n$. $A_{ij} \approx 1$ 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. 1.[1] Hence, we can measure the level of uncertainty in clustering two points $\mathbf{x}_i$ and $\mathbf{x}_j$ as follows:

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

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:

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

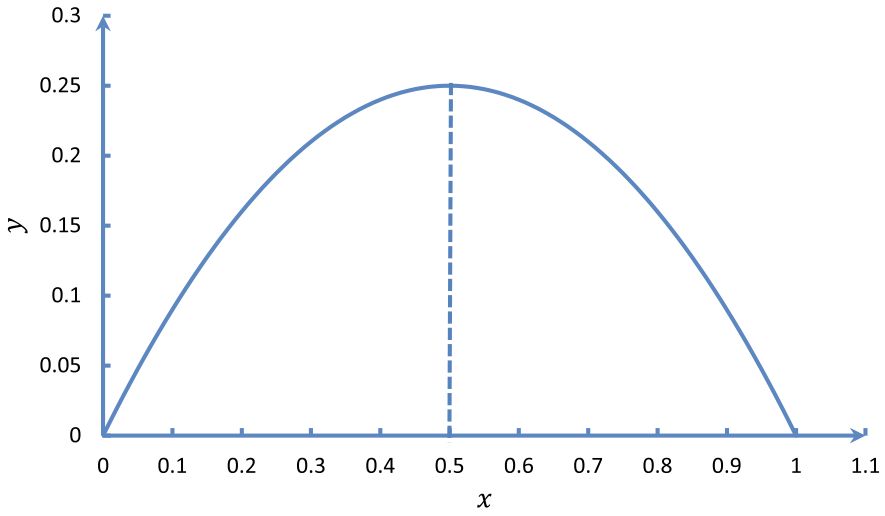

Fig. 1 Quadratic function: $y(x) = x(1 - x)$

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

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:

$$w_i = \frac{w_i' + e}{1 + e} \tag{4}$$

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 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 [23,31].

3.3 Algorithms

In this subsection, 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.

3.3.1 Weighted-object meta-clustering algorithm (WOMC)

The first approach we introduce is a weighted version of the Meta-Clustering Algorithm (MCLA) introduced in [25]. 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 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.

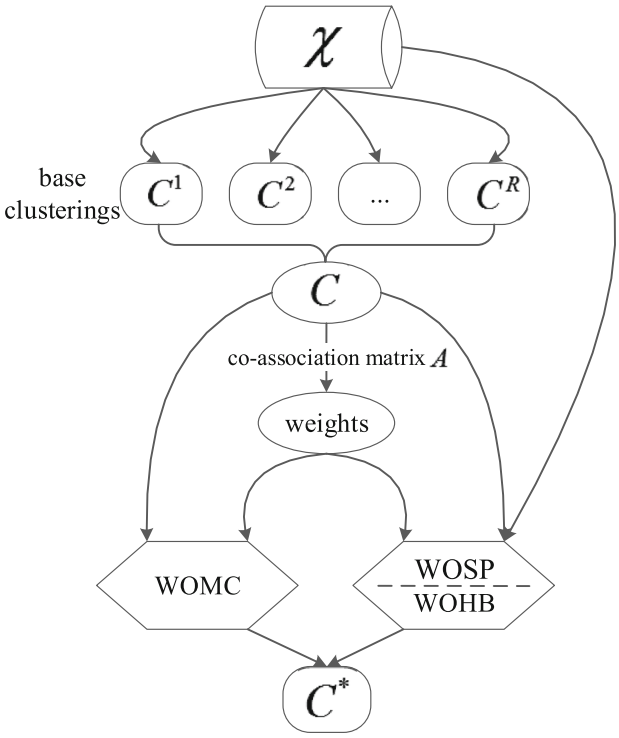

Fig. 2 The process map of Weighted-Object Ensemble Clustering (WOEC)

We explicit the components of each C^r in C. This gives: $C = \{C_1^1, \ldots, C_{k_1}^1, C_1^2, \ldots, C_{k_2}^2, \ldots, C_1^R, \ldots, C_{k_R}^R\}$, where each 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:

$$S_{ij} = \frac{\sum_{\mathbf{x}_k \in C_i \cap C_j} w_k}{\sum_{\mathbf{x}_k \in C_i \cup C_j} w_k} \tag{5}$$

where w_k is the weight associated with the k-th object and is defined in Eqs. (1–4). The definition of the similarity between two feature sets has been discussed in [27] and was adapted for the semantic web by [3]. Here, the approach is different in that Eq. (5) incorporates the weight of each object into the similarity between the two sets.

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 [15]. 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 with 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. An illustration of this process can be found in Sect. 3.4 of [25]. 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. Algorithm 1 describes the WOMC algorithm. Here k^* is predefined by the user and METIS [15] is a software package for graph partitioning.

Input:
C; METIS; 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 with the edges are computed using Eq. (5)
3: $MetaClusters = \text{METIS}(G, k^*)$ [*Partitions the meta-graph into k^* meta-clusters*]
4: $C^* = Assign(MetaClusters)$ [*Assigns each object to the meta-cluster in which it participates most strongly*]
5: **return** C^*

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

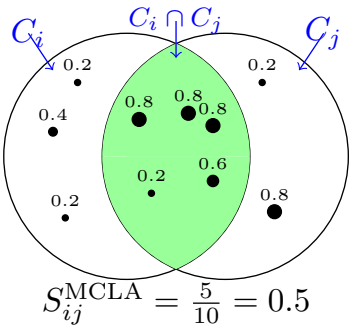

Fig. 3 Illustration of similarity measures between clusters in WOMC and in MCLA

The similarity measure defined in Eq. (5) computes the similarity of two clusters, not only based on the fraction of the shared points, but also on the values of the weights associated with 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.

An example is shown in Fig. 3. Clusters C_i and C_j share 5 out of 10 data points. The number associated with each point represents the corresponding object's weight. According to the MCLA algorithm, the similarity between C_i and C_j is $\frac{5}{10} = 0.5$. At the same time, we observe that the intersection of C_i and C_j contains some of the points with larger weights, i.e., three points with weight 0.8 and one with point 0.6. This means that C_i and C_j agree on most of the hard-to-cluster objects. The larger similarity score computed by the WOMC algorithm (i.e., 0.64) is able to capture, unlike MCLA, this aspect of the data.

Lets see how a change of the weights assigned to the points affects the value of S_{ij}. Suppose C_i and C_j have fixed points assigned to them. Thus, their Jaccard similarity is also fixed. Lets 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. Lets now fix the weights assigned to the points is $(C_i \cap C_j)$. 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.

3.3.2 Weighted-object similarity partitioning algorithm (WOSP)

Previous work on clustering [11,26,30] suggested to assign large weights to objects that are hard to be clustered, and iteratively move cluster centers toward areas that contain objects with large weights. Inspired by this work, we proceed in a similar way 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 $C^r (r = 1, 2, \ldots, R)$, we compute the corresponding weighted center as follows:

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

where $l = 1, \ldots, k_r$. The basic idea behind this step is to move the center of each cluster toward the region consisting of the objects that are hard-to-cluster. We call it *shifting center technique*. The experiments presented in Sect. 5 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:

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

where $t > 0$ is a parameter. The probability of cluster C_l^r, given $\mathbf{x}_i$, can then be defined as:

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

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 $\mathbf{x}_i$:

$$P_i^r = \left(P\left(C_1^r|\mathbf{x}_i\right), P\left(C_2^r|\mathbf{x}_i\right), \ldots, P\left(C_{k_r}^r|\mathbf{x}_i\right)\right) \tag{9}$$

where $\sum_{l=1}^{k_r} P\left(C_l^r|\mathbf{x}_i\right) = 1$. 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.

Figure 4 shows an example. $C^r = \{C_1^r, C_2^r, C_3^r\}$ represents one base clustering with three clusters, C_1^r, C_2^r, and C_3^r. The red points $\mathbf{m}_l^r$ $(l = 1, \ldots, k_r)$ correspond to the weighted centers. In C_2^r, the gray point denotes the original center of this cluster. $\mathbf{x}_i$ is a data point and its representation in the space of coordinates relative to the three cluster centroids is $P_i^r = (P(C_1^r|\mathbf{x}_i), P(C_2^r|\mathbf{x}_i), P(C_3^r|\mathbf{x}_i))$.

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:

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

Input:
$\mathcal{X}$; C; METIS; 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: $C^* = \text{METIS}(G, k^*)$ [*Partitions the graph into k^* parts, each representing a final cluster*]
4: **return** C^*

Algorithm 2: Weighted-Object Similarity Partitioning Algorithm (WOSP)

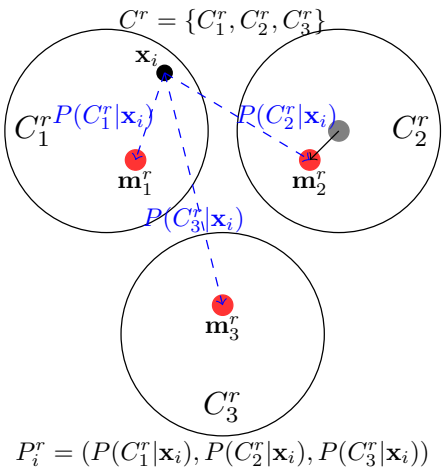

Fig. 4 Illustration of the similarity measure used by WOSP

where T denotes the transpose of a vector. Each base clustering produces one similarity matrix. Thus, given R base clusterings, we obtain R similarity matrices $S^r (r = 1, \ldots, R)$. We combine these matrices into one final similarity matrix S:

$$S = \frac{1}{R} \sum_{r=1}^{R} S^r \tag{11}$$

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.

3.3.3 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

Input:
$\mathcal{X}$; C; METIS; k^*
Output:
The final consensus clustering C^*.
1: Assign weights to objects using the one-shot weight assignment
2: Construct the hybrid bipartite graph $G = (V, E)$
3: $C_{\text{hybrid}} = \text{METIS}(G, k^*)$ [*Partitions the graph into k^* parts*]
4: **return** $C^* =$ the partition of objects in C_{hybrid}

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

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:

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

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

$$S = \begin{bmatrix} 0 & B^{\mathrm{T}} \\ B & 0 \end{bmatrix}$$

where the $n \times n_c$ matrix B is defined as

$$B = \begin{bmatrix} P_1^1 & P_1^2 & \cdots & P_1^R \\ P_2^1 & P_2^2 & \cdots & P_1^R \\ \cdots & \cdots & \cdots & \cdots \\ P_n^1 & P_n^2 & \cdots & P_n^R \end{bmatrix}$$

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.

3.4 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 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. (13). The main idea is to move cluster centers toward regions with larger weights iteratively. Compared to the weighted version of k-means in [21], WOK-means computes all objects' weights in advance using the co-association matrix provided by the base clusterings, while [21] assigns weights to objects iteratively.

3.5 Computational complexity

The computational and storage complexities of the one-shot weight assignment technique are $O(n^2)$ (where n is the total number of data points). In fact, the technique needs to construct the $n \times n$ co-association matrix and measure the level of uncertainty between each

Input:
$\mathcal{X}$; C; k^*
Output:
The final clustering C^*
1: Assign weights to objects using the one-shot weight assignment
2: Randomly generate k^* cluster centers $\mathbf{c}_1, \ldots, \mathbf{c}_{k^*}$, each corresponding to a cluster $C_k = \emptyset$ ($k = 1, \ldots, k^*$)
3: **repeat**
4: [*assigning each object to its closest weighted center*] $\forall i, C_{\overline{k}} = C_{\overline{k}} \cup \mathbf{x}_i$, where $\overline{k} = arg\min_k \|\mathbf{x}_i - \mathbf{c}_k\|$
5: [*generating new weighted centers*]

$$\mathbf{c}_k = \frac{\sum_{\mathbf{x}_i \in C_k} w_i \mathbf{x}_i}{\sum_{\mathbf{x}_i \in C_k} w_i}, k = 1, \ldots, k^* \tag{13}$$

6: **until** All weighted centers do not change or the maximum number of iterations is reached
7: **return** $C^* = \{C_1, C_2, \ldots, C_{k^*}\}$

Algorithm 4: Weighted-Object k-means (WOKmeans)

pair of objects. The objects' weights are computed only once and can then be used in the three proposed consensus clustering algorithms and in the weighted version of k-means. The computational and storage complexities of CSPA are both quadratic in n, and those of MCLA are near-linear in n [25]. Similarly to CSPA, WOSP also needs to construct a $n \times n$ similarity matrix among the objects. Thus its complexity is $O(n^2)$. Once the objects' weights are obtained, the computing complexity of WOMC is equal to MCLA, since the only difference between the two is the way the meta-graph is constructed. WOHB constructs a $(n_c + n) \times (n_c + n)$ ($n_c = \sum_{r=1}^{R} k_r$) hybrid bipartite graph and then partitions it into several hybrid parts. Typically $n_c << n$, thus the computing complexity of WOHB is also $O(n^2)$. In summary, the computational complexities of the three proposed ensemble clustering methods are $O(n^2)$. Given the weight for each object, the complexity of WOKmeans is linear in n.

4 Experimental setup

4.1 Data sets

To compare the performance of our WOEC approach against the state-of-the-art techniques, we conducted experiments on various kinds of data, including three toy examples and fifteen real data sets. Table 1 gives the characteristics of all the data sets used in our experiments.

The toy examples are shown in Fig. 5 (after standardization). Toy example 1 (Toy1) consists of three classes with 150 data points each. The three classes were all generated according to multivariate Gaussian distributions and all share the same covariance matrix [10 0; 0 10]. The mean vectors are (−10, 10), (10, 10) and (0, 0). Toy example 2 (Toy2) contains three circle-shaped classes. Each class contains 300 points uniformly distributed within the respective circle. The centers of the two upper circles are (−4, 6) and (4, 6); both circles have a radius of 3. The center and radius of the third circle are (0, 0) and 4, respectively. Toy example 3 (Toy3) consists of five classes with 300 points each. All five classes were generated according to multivariate Gaussian distributions and all share the same covariance matrix [5 0; 0 5]. The mean vectors are (−5.29, 0), (−8.559, 10.062), (0, 16.281), (8.559, 10.062), and (5.29, 0).

Table 1 Data sets used in the experiments

	Data sets	#Objects	#Features	#Classes
Toy examples	Toy1	450	2	3
	Toy2	900	2	3
	Toy3	1500	2	5
Real data	AustralianCredit	690	14	2
	BalanceScale	625	4	3
	Beef	60	470	5
	BreastTissue	106	9	6
	Glass	214	9	6
	HayesRoth	132	4	3
	Iris	150	4	3
	Leukemia	72	3571	2
	Satimage	420	36	6
	Semeion	1593	256	10
	Sony	980	65	2
	Spambase	500	57	2
	Vowel	990	10	11
	Wall	4302	2	2
	Yeast	1136	8	3

Beef and Sony (SonyAIBORobotSurfaceII) are data sets taken from the UCR time series repository.[2] Leukemia is a gene expression data set.[3] The other data sets are from the UCI repository.[4] In particular, Satimage originally contained 6435 objects; we randomly selected 420 images equally distributed among the six classes for our experiments. Spambase had 4601 objects; we randomly sampled 500 objects (250 for each class). The two largest classes of Wall (Wall-Following Robot Navigation Data) and the three largest classes of Yeast were used in the experiments. The other eight UCI data sets were used in their original form. For each toy example and each real data set, features are standardized to have zero mean and unit variance.

4.2 Evaluation measures

Various metrics exist to evaluate clustering results [2,16]. Since the labels of the data are known, we use Rand Index (RI) [12], Adjusted Rand Index (ARI) [12], and Normalized Mutual Information (NMI) [25] as validity indices. Both RI and NMI ranges between 0 and 1, while ARI values belong to the interval $[-1, 1]$. A larger value of RI (ARI or NMI) is indicative of a better clustering result.

[2] http://www.cs.ucr.edu/~eamonn/time_series_data/.

[3] http://stat.ethz.ch/~dettling/bagboost.html.

[4] http://archive.ics.uci.edu/ml/index.html.

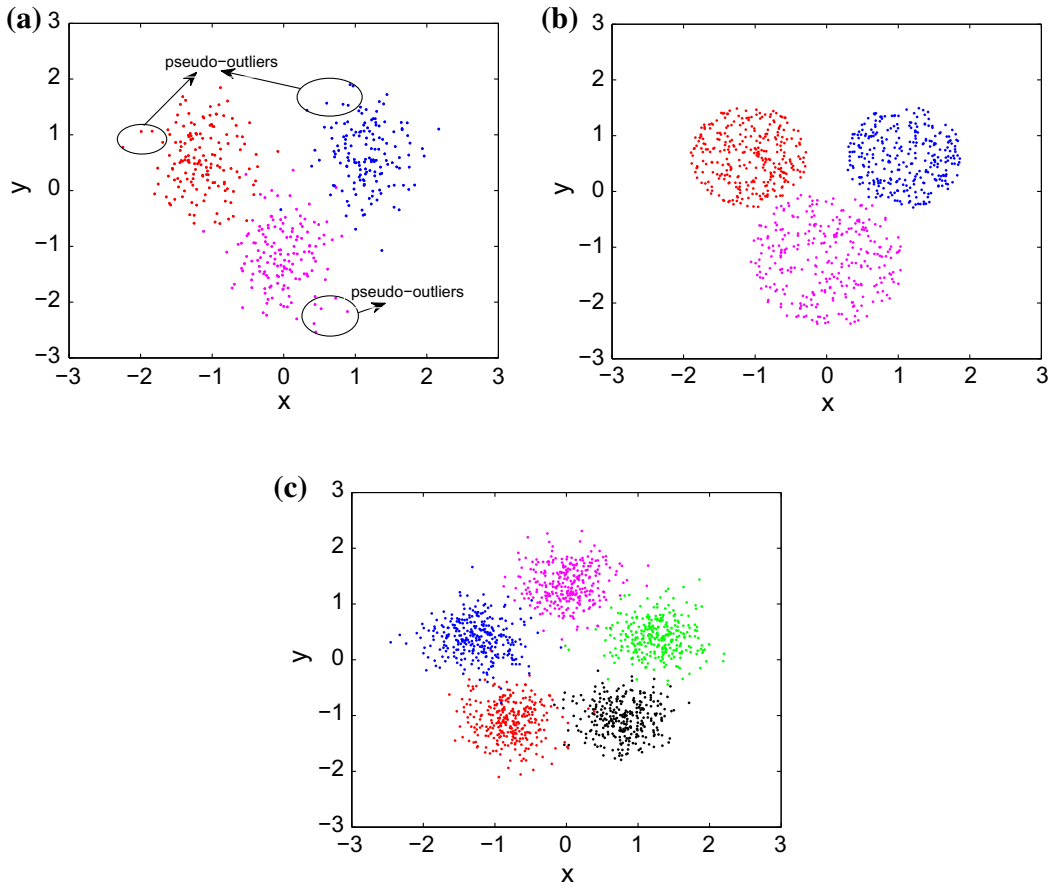

Fig. 5 Toy examples

4.3 Base clusterings

To generate diverse base clusterings, we used k-means combined with random sampling of data and features. 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. Each remaining object was assigned to the cluster with the closest center (based on Euclidean distance). 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. Each run of k-means was initialized with randomly selected centers.

5 Results and analysis

For a fair comparison, we used the METIS package [15] 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 sets and all methods.[5] In our experiments, the value of t [see Eq. (7)], for each base clustering, is set to the average of the squared

[5] In 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.

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 compared the proposed WOEC methods against eight state-of-the-art ensemble clustering algorithms, namely: CSPA [25], MCLA [25], HBGF [6], BCE [28], WCC [17], ECMC [29], WSPA[1,4] and WBPA[1,4]. (HGPA [25] was not included in our experiments since it typically performs worse than CSPA and MCLA.)

5.1 Results on toy examples

To illustrate the effectiveness of the proposed three WOEC algorithms and of WOKmeans, we first conducted experiments on the three toy examples. We set the ensemble size $R = 40$. RI, ARI, and NMI are used to evaluate the performance of all algorithms. 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 2. Best performances that are statistically significant are highlighted in boldface.

WOSP and WOHB outperform the other ensemble methods and exceed the max index on Toy1 and Toy2. On Toy3, WOMC (along with some of the competitive techniques) gives the best performance. In the following, we analyze this resulting behavior.

To investigate the effectiveness of the *shifting center technique* used by the WOSP and WOHB methods, we recorded the status of one of the base clusterings in one of the independent runs. Figure 6 shows our findings for each of the Toy examples. In each subfigure, the hollow circles denote the centers and the solid circles are the weighted centers [see Eq. (6)] of clusters. Points that receive the largest weights are highlighted with squares. As expected, between cluster points tend to gain large weights and thus have larger contributions to the shifting centers' procedure. As Fig. 6a shows, the *shifting center technique* forces all the centers to move toward between cluster areas. As such, the influence of the pseudo-outliers[6] is reduced, thereby leading to more robust ensemble clustering results. In Fig. 6b, the center of the lower cluster moves by a big step toward the region with hard-to-cluster points. We observe that the three clusters contain the same number of points, but the lower cluster has a bigger radius and is therefore sparser. The shift of its center reduces the negative influence that clusters with different densities can cause.

For Toy3 (see Fig. 6c), each of the five clusters has a boundary in common with other two clusters. In this scenario, the size of the centers' shifting is small and the benefit of this procedure is therefore limited. Although WOSP and WOHB are still effective as an ensemble strategy, WOMC, along with MCLA, BCE, and WCC, give the best performance on Toy3 (see Table 2).

We also compared the performance of k-means and WOKmeans on the synthetic data sets. Results in Table 3 show that WOKmeans significantly outperforms k-means on Toy2 and Toy3. On Toy1, WOKmeans works significantly better w.r.t. NMI, and the average values of RI and ARI of WOKmeans are still larger than those of k-means. k-means is negatively affected by the pseudo-outliers. Although WOKmeans improves upon k-means, its perfor-

[6] As shown in Fig. 5a, the circled points are far away from the mean points of the classes and their density is considerably lower than that of the other points. These points can bias the computation of the mean vector. They are generated from the same distribution as the other points in the same class, 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".

Table 2 Results on toy examples

Data	Index	Min	Max	Mean	CSPA	MCLA	HBGF	BCE	WCC	ECMC	WOEC		
											WOMC	WOSP	WOHB
Toy1	RI	0.9316	0.9651	0.9613	0.9606	0.9624	0.9601	0.9602	0.9624	0.8382	0.9624	**0.9751**	0.9655
	ARI	0.8524	0.9214	0.9130	0.9111	0.9152	0.9099	0.9111	0.9152	0.6816	0.9152	**0.9438**	0.9221
	NMI	0.8289	0.8831	0.8728	0.8690	0.8747	0.8680	0.8720	0.8747	0.7235	0.8747	**0.9072**	0.8861
Toy2	RI	0.9722	0.9775	0.9754	0.9718	0.9753	0.9658	0.9712	0.9753	0.8541	0.9753	0.9944	**0.9980**
	ARI	0.9373	0.9494	0.9446	0.9365	0.9444	0.9230	0.9367	0.9444	0.7069	0.9444	0.9875	**0.9955**
	NMI	0.9142	0.9279	0.9222	0.9006	0.9219	0.8837	0.9177	0.9219	0.7694	0.9219	0.9757	**0.9914**
Toy3	RI	0.8704	0.9837	0.9585	0.9815	**0.9832**	0.9813	**0.9832**	**0.9832**	0.8799	**0.9832**	0.9814	0.9825
	ARI	0.6218	0.9490	0.8765	0.9420	**0.9474**	0.9413	**0.9474**	**0.9474**	0.7060	**0.9474**	0.9418	0.9452
	NMI	0.7321	0.9315	0.8863	0.9209	**0.9294**	0.9205	**0.9294**	**0.9294**	0.8193	**0.9294**	0.9218	0.9251

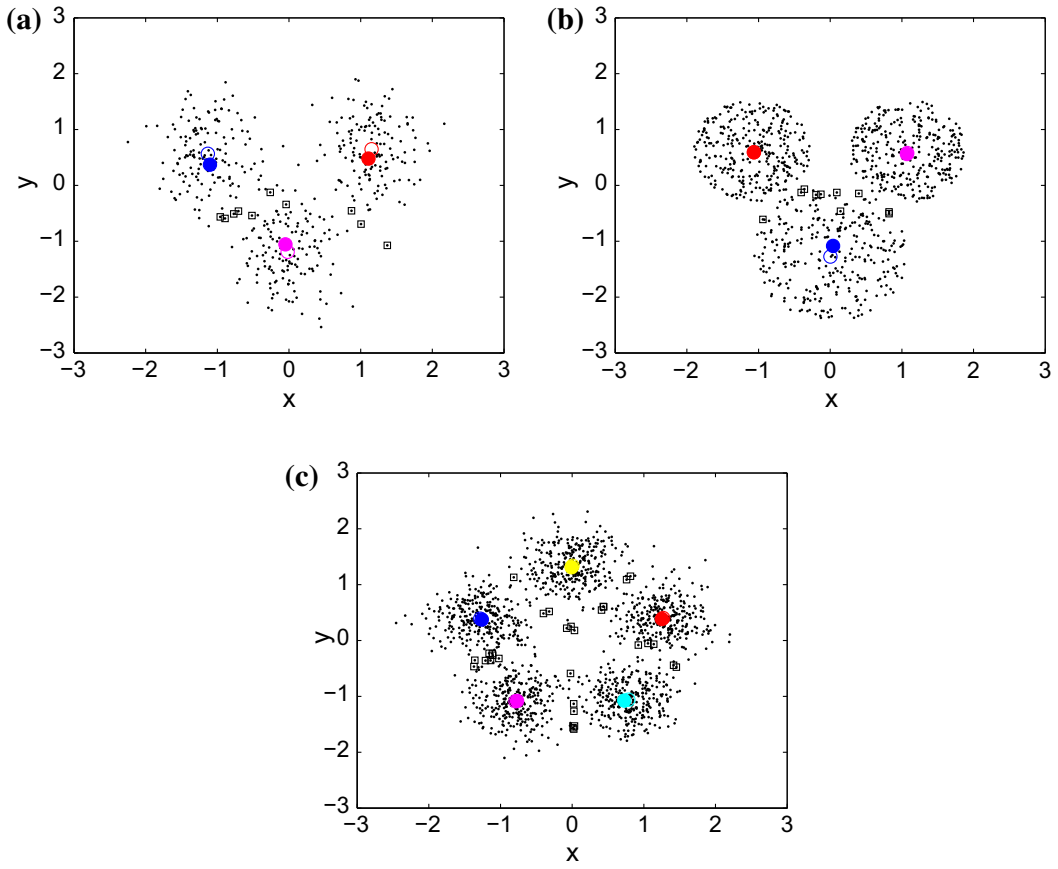

Fig. 6 Illustration of the *shifting center technique* used by WOSP and WOHB

Table 3 k-means versus WOKmeans on toy examples

Data	Index	k-means	WOKmeans
Toy1	RI	$\mathbf{0.9598 \pm 0.0255}$	$\mathbf{0.9628 \pm 0.0029}$
	ARI	$\mathbf{0.9100 \pm 0.0515}$	$\mathbf{0.9161 \pm 0.0065}$
	NMI	0.8711 ± 0.0357	$\mathbf{0.8807 \pm 0.0074}$
Toy2	RI	0.9753 ± 0.0000	$\mathbf{0.9780 \pm 0.0102}$
	ARI	0.9444 ± 0.0000	$\mathbf{0.9504 \pm 0.0220}$
	NMI	0.9219 ± 0.0000	$\mathbf{0.9313 \pm 0.0269}$
Toy3	RI	0.9633 ± 0.1207	$\mathbf{0.9734 \pm 0.0297}$
	ARI	0.8903 ± 0.0727	$\mathbf{0.9191 \pm 0.0852}$
	NMI	0.8945 ± 0.0721	$\mathbf{0.9109 \pm 0.0515}$

mance is generally worse than that of the WOEC methods, demonstrating the effectiveness of ensemble techniques.

In Fig. 7, we visualize the final clustering results of k-means and WOKmeans, and the final consensus clusterings of the three WOEC methods (in a given run) for Toy2. This data set presents a challenge due to the larger radius and sparsity of the lower cluster. It's easy to

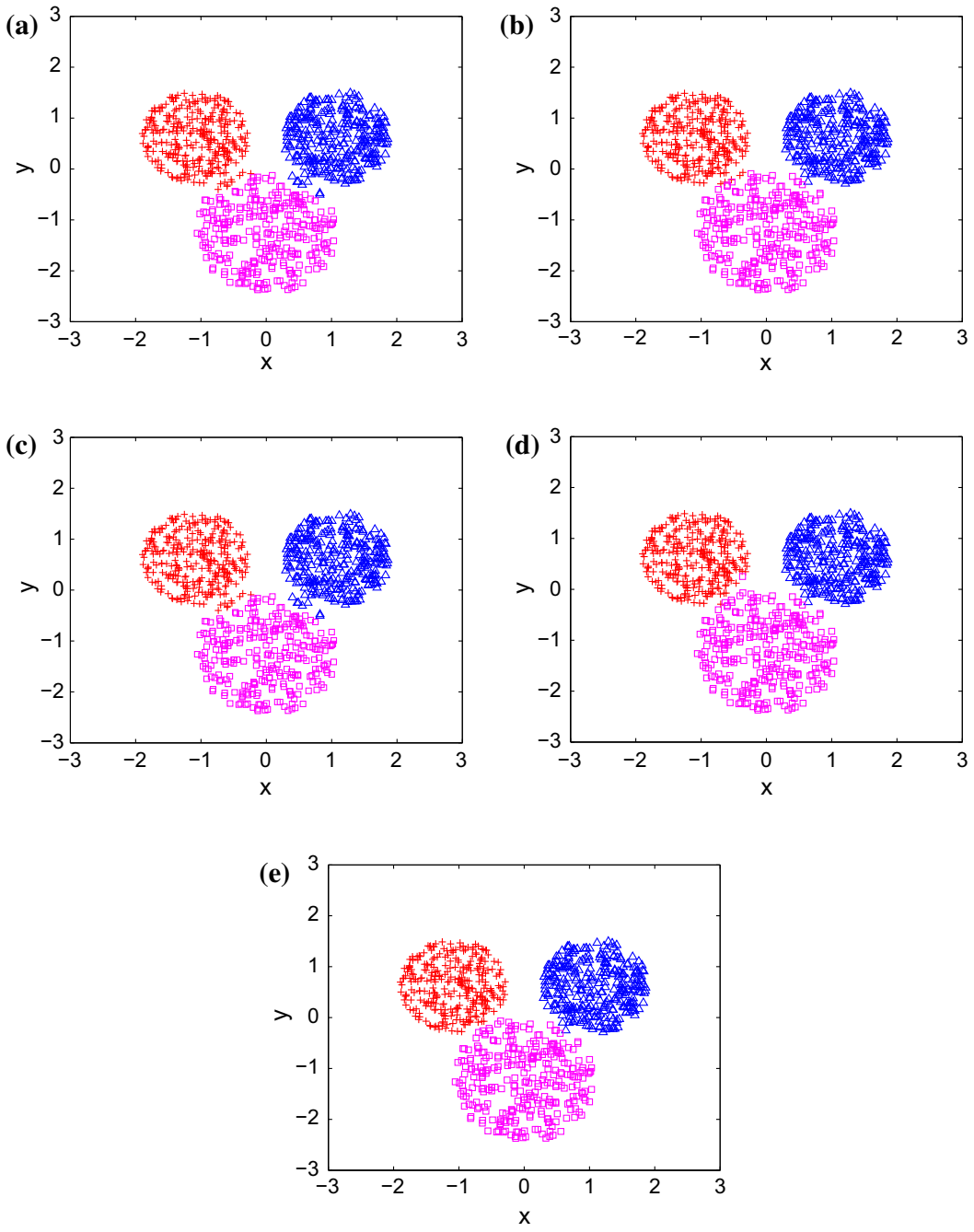

Fig. 7 Toy2: Final (or final consensus) clustering results of **a** k-means; **b** WOKmeans; **c** WOMC; **d** WOSP; and **e** WOHB

see that WOKmeans incorrectly clusters fewer points (located in between-cluster areas) than k-means. Furthermore, comparing the clusterings provided by the three WOEC methods, one can see that WOMC assigns some of the boundary points to the incorrect cluster, while WOSP and WOHB benefit from the effect of the *shifting center technique* and almost achieve a perfect clustering.

From Tables 2 and 3, we can see that in general a better RI value corresponds to a better ARI and NMI values. Therefore, we only report ARI values for the experiments on real data.

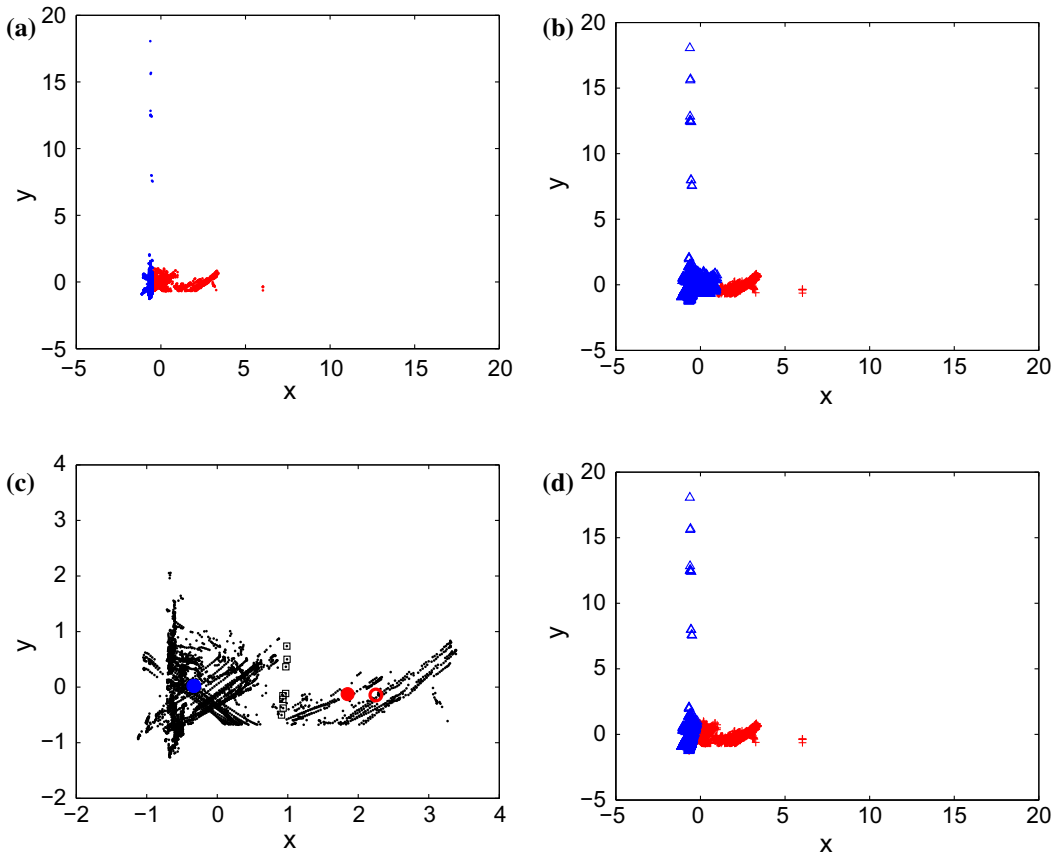

Fig. 8 **a** Wall data; **b** clustering result of WOMC; **c** illustration of the *shifting center technique*; and **d** clustering result of WOSP

5.2 Results on real data

Extensive experiments were conducted on fifteen real data sets to evaluate the effectiveness of the proposed WOEC methods. In the experiments, we chose three different ensemble sizes R: 20, 40, and 60. The results in terms of ARI are shown in Table 4. Each ARI value reported is the average of 100 independent runs. In each row, statistically significant values are highlighted in boldface. We also run k-means 100 times for baseline comparisons; its average ARI is given in Table 4 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. Small fluctuations in performance are generally observed for the WOEC methods for different values of R. In some cases, a steady improvement is observed with an increase in ensemble size.

For each data set, at least one of the WOEC methods achieves the best performance, with the only exception of the Breast data for which CSPA outperforms the other methods. As expected, the ensemble clustering techniques achieve superior performance with respect to the baseline for most of the problems. On Iris, Leukemia, Sony, Spambase, and Wall, both WOSP and WOHB provide a considerable improvement compared to all the other ensemble clustering methods, indicating that the *shifting center technique* works well on real data sets

Table 4 Comparison against existing methods (ARI)

Data	R	Min	Max	Mean	CSPA	MCLA	HBGF	BCE	WCC	ECMC	WOEC		
											WOMC	WOSP	WOHB
Australian	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
(0.4189)	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	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**
(0.1336)	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
Beef	20	0.1544	0.2454	0.1968	0.2506	0.2015	0.2030	0.1860	0.2021	0.1517	0.2280	**0.2538**	0.2302
(0.1888)	40	0.1463	0.2532	0.1949	**0.2547**	0.2030	0.1988	0.1861	0.2181	0.1305	0.2301	**0.2551**	0.2242
	60	0.1359	0.2593	0.1955	0.2518	0.2017	0.1972	0.1918	0.2116	0.1329	0.2218	**0.2538**	0.2353
Breast	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
(0.2952)	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	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
(0.1680)	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	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
(0.0601)	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	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**
(0.5784)	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

Table 4 continued

Data	*R*	Min	Max	Mean	CSPA	MCLA	HBGF	BCE	WCC	ECMC	WOEC		
											WOMC	WOSP	WOHB
Leukemia	20	0.0951	0.5358	0.2006	0.1974	0.1805	0.1940	0.1805	0.1805	0.1645	0.1805	0.2043	**0.2171**
(0.1203)	40	0.0623	0.4259	0.1856	0.1897	0.1793	0.1988	0.1712	0.1805	0.1633	0.1793	0.2120	**0.2221**
	60	0.0241	0.6463	0.1918	0.1911	0.1855	0.1958	0.1668	0.1855	0.1621	0.1830	**0.2093**	**0.2108**
Satimage	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
(0.6099)	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
Semeion	20	0.3113	0.4728	0.3821	0.3873	0.4057	0.3816	0.4025	0.3826	0.3004	**0.4156**	0.2878	0.2262
(0.3614)	40	0.3060	0.4759	0.3829	0.3988	0.4147	0.3904	0.4041	0.3650	0.2965	**0.4183**	0.2892	0.3073
	60	0.2947	0.4815	0.3827	0.3910	**0.4024**	0.3907	0.3807	0.3905	0.3735	**0.4024**	0.2862	0.3224
Sony	20	0.2677	0.3355	0.3123	0.2721	0.3180	0.2587	0.3187	0.3189	0.3181	0.3180	**0.3802**	**0.3792**
(0.3184)	40	0.2463	0.3499	0.3134	0.2974	0.3187	0.2944	0.3024	0.3187	0.3151	0.3190	**0.3832**	0.3822
	60	0.2637	0.3534	0.3142	0.3032	0.3195	0.2918	0.3181	0.3181	0.3134	0.3196	**0.3835**	0.3825
Spambase	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**
(0.1412)	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**
Vowel	20	0.1205	0.2053	0.1646	0.1578	0.1698	0.1586	0.1660	**0.1745**	0.1175	**0.1767**	0.1534	0.1402
(0.1671)	40	0.1143	0.2107	0.1645	0.1607	0.1714	0.1644	0.1711	0.1702	0.0847	**0.1735**	0.1526	0.1431
	60	0.1046	0.2140	0.1635	0.1602	**0.1720**	0.1650	0.1709	0.1702	0.0702	**0.1734**	0.1510	0.1427
Wall	20	0.0295	0.0548	0.0527	0.0260	0.0538	0.5974	0.0538	0.0176	0.0538	0.0538	0.7130	**0.7237**
(0.0528)	40	0.0159	0.0585	0.0529	0.0303	0.0538	0.2700	0.0538	0.0172	0.0538	0.0538	0.6911	**0.7105**
	60	0.0084	0.0554	0.0525	0.0284	0.0538	0.3781	0.0538	0.0105	0.0538	0.0538	0.7103	**0.7130**
Yeast	20	0.0168	0.1254	0.0736	0.0616	0.0826	0.0627	0.0726	0.0751	0.0661	0.0737	**0.0912**	0.0286
(0.0675)	40	0.0147	0.1296	0.0717	0.0637	0.0751	0.0537	0.0703	0.0997	0.0826	0.0732	0.0896	**0.1137**
	60	0.0146	0.1375	0.0734	0.0616	0.0745	0.0714	0.0880	0.0574	0.0401	0.0882	0.0834	**0.1052**

as well. In particular, the performance of WOSP and WOHB is superior to the max index on Sony and Wall. Compared to all the competitive methods, WOSP and WOHB achieve an outstanding improvement on Wall. To investigate this behavior, in Fig. 8a, b we plotted the ground truth labels of the Wall data and the clustering result of WOMC, respectively. The WOMC technique, along with all the other comparing methods, performs poorly on this data set. Figure 8c shows the status of one base clustering in one independent run. Here, we focus on the x and y values in the range $[-2, 4]$. Points that obtain the largest weight values are also highlighted with squares. We can see that the *shifting center technique* causes the right center (hollow red circle) to move by a big step toward the hard-to-cluster area. As a consequence, the final boundary generated by WOSP also shifts to the left, closer to the ground truth boundary, as shown in Fig. 8d, thereby giving the observed improved performance. A similar behavior was observed for WOHB as well.

As pointed out earlier, our WOMC technique is a variation of MCLA. Compared to MCLA, WOMC achieves a superior performance on Balance, Beef, Breast, Iris, Satimage, Semeion, and Vowel. On the remaining data sets, MCLA and WOMC have a comparable performance. This shows the benefits of considering objects' weights when measuring the similarity of two clusters in a meta-clustering algorithm. With respect to the most recent ECMC technique, the WOEC methods can achieve 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 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 [29]. Furthermore, ECMC also needs to find a suitable value for a parameter C that minimizes $\mathbf{1}^T M \mathbf{1}$, where M is a pairwise similarity matrix (details are in [29]), 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 the parameter t, as the analysis in Sect. 5.5 shows.

5.3 Comparison against WSPA and WBPA

WSPA and WBPA are two weighted ensemble clustering methods which make use of weights associated with clusters. To perform comparisons against them, we generated the base clusterings using the LAC algorithm, as described in [1,4]. 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 5. As before, statistically significant results are given in boldface. At least one of the WOEC methods gives the best performance in each data set, except for Beef; WSPA is the winner in this case. Again, WOSP and WOHB offer an outstanding improvement on the Wall data set.

5.4 Evaluation of WOKmeans

In Sect. 5.1, 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 perform more comparisons using the real data sets. In each run, the maximum number of iterations for 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 size is always 40. Note that only WOKmeans is affected by the ensemble size. Figure 9 gives the results. We observe that the average ARI

Table 5 Comparison against WSPA and WBPA (ARI)

Data	*R*	WSPA	WBPA	WOEC		
				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
Beef	20	**0.2656**	0.2233	0.2379	0.2538	0.2319
	40	**0.2656**	0.2096	0.2420	0.2564	0.2295
	60	**0.2656**	0.2051	0.2435	0.2551	0.2298
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
Leukemia	20	0.1924	0.2111	**0.2159**	0.1898	0.2113
	40	0.1951	0.1961	**0.2013**	0.1871	0.2000
	60	0.2004	0.1897	**0.2026**	0.1871	0.1922
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
Semeion	20	0.2476	0.1429	**0.3964**	0.2784	0.2198
	40	0.2475	0.2544	**0.3940**	0.2936	0.3052
	60	0.2494	0.3098	**0.4083**	0.2807	0.3211
Sony	20	**0.3714**	0.3700	0.3090	0.3602	0.3578
	40	0.3716	0.3704	0.3156	**0.3875**	0.3849
	60	0.3767	0.3764	0.3182	**0.3934**	**0.3937**
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**
Vowel	20	0.1192	0.1169	**0.1679**	0.1465	0.1426
	40	0.1168	0.1418	**0.1733**	0.1507	0.1443
	60	0.1177	0.1423	**0.1664**	0.1477	0.1433

Table 5 continued

Data	R	WSPA	WBPA	WOEC		
				WOMC	WOSP	WOHB
Wall	20	0.3226	0.3221	0.0561	**0.8449**	**0.8448**
	40	0.3257	0.3252	0.0544	**0.8361**	0.8311
	60	0.3255	0.3248	0.0541	**0.8150**	0.8105
Yeast	20	**0.0826**	0.0286	0.0653	**0.0839**	0.0321
	40	0.0843	0.0513	0.0589	0.0868	**0.0980**
	60	0.0863	0.0344	0.0604	0.0841	**0.0964**

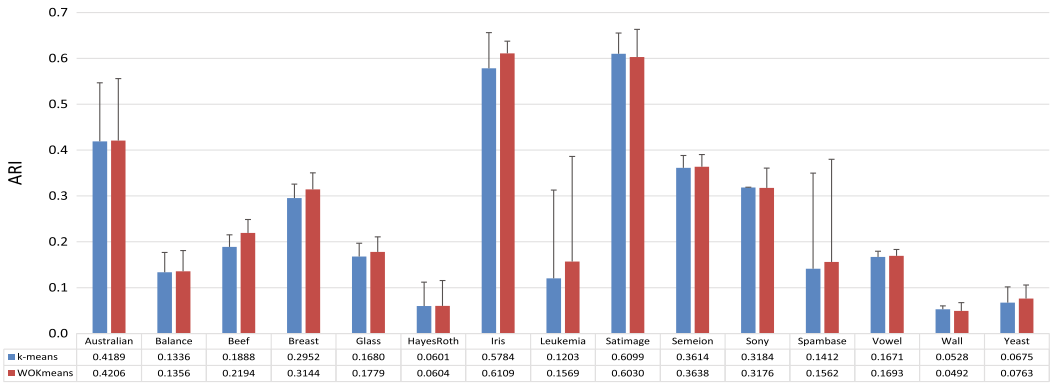

Fig. 9 Comparisons between k-means and WOKmeans (ARI)

values of WOKmeans are always higher than those of k-means except for Satimage, Sony, and Wall. In particular, WOKmeans improves significantly upon k-means on Beef, Breast, Glass, and Iris. This result confirms again the benefit of the *shifting center technique*. As suggested by previous analysis [11,26,30], moving the centers toward the hard-to-cluster data can improve the clustering of the same. However, WOKmeans is a variation of k-means, and as such it's still sensitive to the initial setting (i.e., the choice of initial cluster centers) and likely to converge to local minima. As a consequence, the difference in performance between WOKmeans and k-means is not significant on some data sets. In comparison with the WOEC methods, typically WOKmeans' performance is lower, demonstrating again the effectiveness of ensemble techniques.

5.5 Sensitivity analysis of parameters

From Fig. 9, we can see that the performance of k-means is unstable (i.e., large standard derivation values) on Australian, Leukemia, and Spambase. In this subsection, we test how sensitive the WOEC methods are with respect to parameters on these three data sets.

We analyzed first the sensitivity on the t parameter used in Eq. (7). We set the ensemble size $R = 40$ and change the parameter t from 1 to 100. Figure 10 shows the results. We can see that both WOSP and WOHB are stable when $t > 20$ for Leukemia and when $t > 5$ for Spambase. The performance is stable for the entire range of tested values for Australian. 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 affect the sequence of probability values. This explains

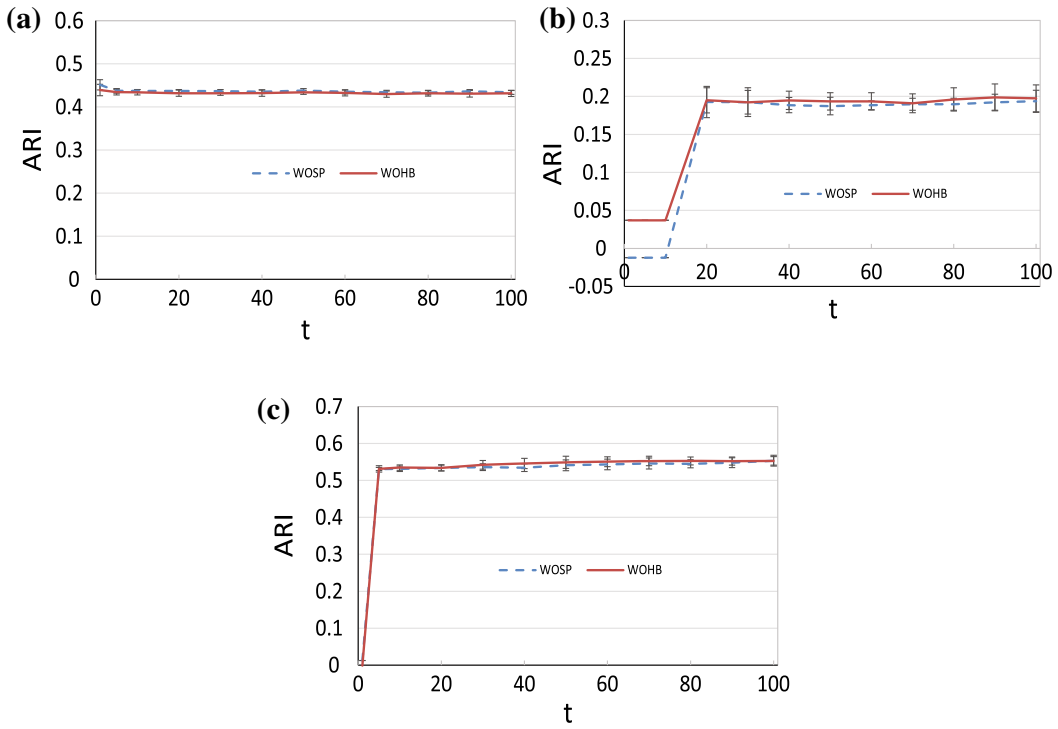

Fig. 10 Sensitivity analysis of parameter t (ARI)

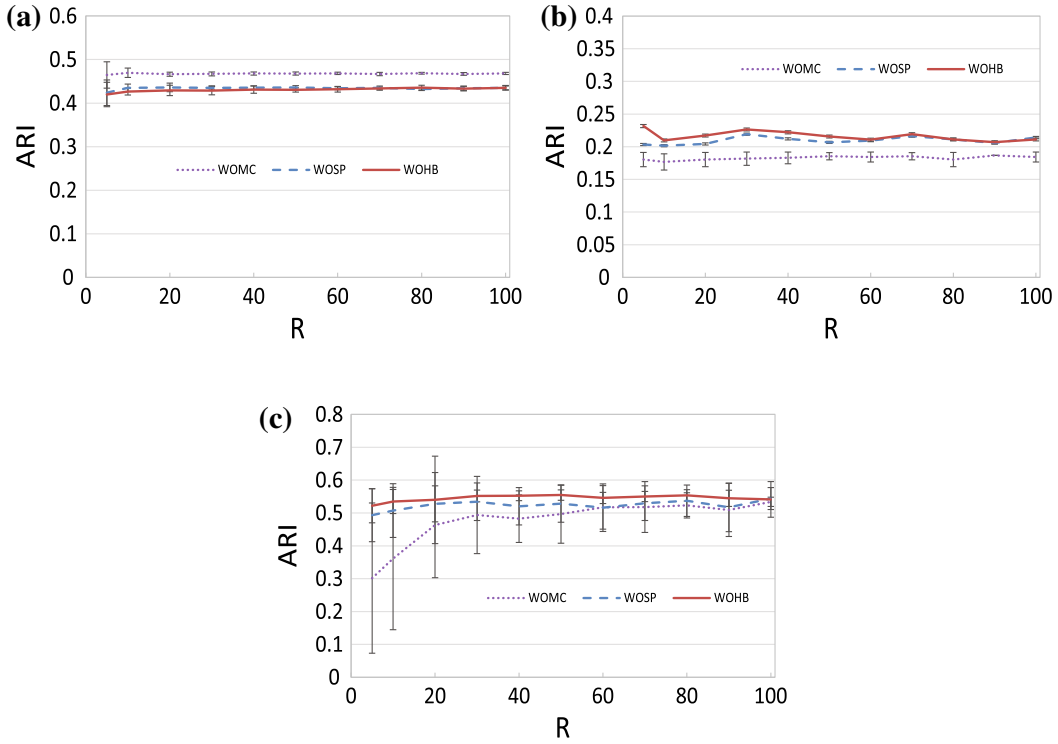

Fig. 11 Sensitivity analysis of ensemble size R (ARI)

why both WOSP and WOHB are robust with respect 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, $\forall k$, $d(\mathbf{x}_i, \mathbf{m}_k^r) \to 0$ and $d(\mathbf{x}_i, \mathbf{m}_k^r) \to 1$, 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.

Second, we investigated the sensitivity of the three WOEC methods with respect to the ensemble size R. We change R from 5 to 100, and Fig. 11 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, 10, and 30 on Australian, Leukemia, and Spambase, respectively. On Australian and Leukemia, all three WOEC methods can achieve a good performance when $R = 5$. WOSP and WOHB achieve a stable performance also on Spambase in correspondence of a small ensemble size.

In terms of stability, we can see from Fig. 10 that WOSP and WOHB consistently achieve a stable performance on the three data sets. Figure 11 shows that the standard derivations of the WOEC methods reduce in value when $R > 20$ on Spambase, and they are consistently small on Australian and Leukemia. This indicate that although the base clustering method (k-means in our experiments) is unstable, the WOEC methods are still able to provide stable and good clustering results.

6 Conclusions and future work

This paper introduces a framework to perform *Weighted-Object Ensemble Clustering* (WOEC) which performs a one-shot weight assignment to objects. We discussed three variants of WOEC and one weighted version of k-means. All three WOEC algorithms reduce the ensemble clustering problem to a graph partitioning one. We conducted extensive experiments to investigate the effectiveness and stability of the WOEC approach, and further analyzed the conditions under which the three different consensus techniques are most effective. Like most existing ensemble techniques, the time complexity of the WOEC methods is quadratic in the number of points. This makes it difficult to use them for clustering large scale data. Our next step is to design a novel ensemble clustering approach which takes into account both the weight information of objects and efficiency.

Acknowledgments This paper was in part supported by Grants from the Fundamental Research Funds for the Central Universities of China (No. A03012023601042), the Natural Science Foundation of China (Nos. 61572111, 61402378), and the Natural Science Foundation of CQ CSTC (Nos. cstc2014jcyjA40031, cstc2016jcyjA0351).

References

1. Al-Razgan M, Domeniconi C (2006) Weighted clustering ensemble. In: Proceedings of the 6th SIAM international conference on data mining, pp 258–269
2. Arbelaitz O, Gurrutxaga I, Muguerza J, Perez JM, Perona I (2013) An extensive comparative study of cluster validity indices. Pattern Recognit 46(1):243–256
3. d'Amato C, Fanizzi N, Esposito F (2009) A semantic similarity measure for expressive description logics. CoRR. arXiv:0911.5043
4. Domeniconi C, Al-Razgan M (2009) Weighted cluster ensembles: methods and analysis. ACM Trans Knowl Discov Data 2(4):17:1–17:40
5. Domeniconi C, Papadopoulos D, Gunopulos D, Ma S (2004) Subspace clustering of high dimensional data. In: Proceedings of the SIAM international conference on data mining, pp 517–521
6. Fern XZ, Brodley CE (2004) Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the 21th international conference on machine learning, pp 281–288

7. Fred A, Jain AK (2002a) Data clustering using evidence accumulation. In: Proceedings of the 16th international conference of pattern recognition, pp 276–280
8. Fred A, Jain AK (2002b) Evidence accumulation clustering based on the k-means algorithm. In: Proceedings of the joint IAPR international workshop on structural, syntactic, and statistical pattern recognition, pp 442–451
9. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55:119–139
10. Ghosh J, Acharya A (2011) Cluster ensembles. WIREs Data Min Knowl Discov 1(4):305–315
11. Hamerly G, Elkan C (2002) Alternatives to the k-means algorithm that find better clusterings. In: Proceedings of the 11th international conference on information and knowledge management, pp 600–607
12. Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218
13. Jing L, Tian K, Huang JZ (2015) Stratified feature sampling method for ensemble clustering of high dimensional data. Pattern Recognit 48(11):3688–3702
14. Karypis G, Aggarwal R, Kumar V, Shekhar S (1997) Multilevel hypergraph partitioning: application in vlsi domain. In: Proceedings of the design and automation conference, pp 526–529
15. Karypis G, Kumar V (1998) A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J Sci Comput 20(1):359–392
16. Legany C, Juhasz S, Babos A (2006) Cluster validity measurement techniques. In: Proceedings of the 5th WSEAS international conference on artificial intelligence, knowledge engineering and data bases, pp 388–393
17. Li T, Ding C (2008) Weighted consensus clustering. In: Proceedings of the 8th SIAM international conference on data mining, pp 798–809
18. Li T, Ding C, Jordan MI (2007) Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In: Proceedings of the 7th IEEE international conference on data mining, pp 577–582
19. Liu H, Liu T, Wu J, Tao D, Fu Y (2015) Spectral ensemble clustering. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 715–724
20. Ng AY, Jordan MI, Weiss Y (2001) On spectral clustering: analysis and an algorithm. In: Dietterich TG, Becker S, Ghahramani Z (eds) Advances in neural information processing systems. MIT Press, Vancouver, British Columbia, pp 849–856
21. Nock R, Nielsen F (2006) On weighting clustering. IEEE Trans Pattern Anal Mach Intell 28(8):1223–1235
22. Ren Y, Domeniconi C, Zhang G, Yu G (2013) Weighted-object ensemble clustering. In: Proceedings of the IEEE 13th international conference on data mining, pp 627–636
23. Schapire RE (1990) The strength of weak learnability. Mach Learn 5(2):197–227
24. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
25. Strehl A, Ghosh J (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617
26. Topchy A, Minaei-Bidgoli B, Jain AK, Punch WF (2004) Adaptive clustering ensembles. In: Proceedings of the 17th international conference on pattern recognition, pp 272–275
27. Tversky A (1977) Features of similarity. Psychol Rev 84(4):327–352
28. Wang H, Shan H, Banerjee A (2009) Bayesian cluster ensembles. In: Proceedings of the 9th SIAM international conference on data mining, pp 209–220
29. Yi J, Yang T, Jin R, Jain AK, Mahdavi M (2012) Robust ensemble clustering by matrix completion. In: Proceedings of the 12th IEEE international conference on data mining, pp 1176–1181
30. Zhang B, Hsu M, Dayal U (2001) K-harmonic means—a spatial clustering algorithm with boosting. In: Roddick JF, Hornsby K (eds) Temporal, spatial, and spatio-temporal data mining, vol 2007. Springer, Berlin, Heidelberg, pp 31–45
31. Zhou Z (2012) Ensemble methods: foundations and algorithms. Chapman & Hall, London

Yazhou Ren is a Lecturer in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. He received his B.Sc. degree in Information and Computation Science and Ph.D. degree in Computer Science from South China University of Technology, Guangzhou, China in 2009 and 2014, respectively. He visited the Data Mining Laboratory at George Mason University, USA, for 2 years from 2012 to 2014. His current research interests include machine learning, data mining and pattern recognition.

Carlotta Domeniconi is an Associate Professor in the Department of Computer Science at George Mason University, VA, USA. She received a B.Sc. degree in Computer Science from the University of Milan, Italy, in 1992, and a Ph.D. degree in Computer Science from the University of California, Riverside, in 2002. Her research interests include pattern recognition, machine learning, data mining, and feature relevance estimation. She has published extensively in premier data mining and machine learning conferences and journals. She has been the Principal Investigator (PI) and co-PI on a number of projects sponsored by the NSF, the US Army, the Air Force, the DoD, and FINRA. She is a recipient of an NSF CAREER Award.

Guoji Zhang is a Professor in the School of Sciences, South China University of Technology, Guangzhou, China. He received his B.Sc. degree in Computer Application and Ph.D. degree in Circuit and System from South China University of Technology, in 1977 and 1999, respectively. His research interests include computational intelligence, computational electromagnetic and cryptology. He has published over 50 research papers.

Guoxian Yu is an Associate Professor in the College of Computer and Information Science, Southwest University, Chongqing, China. He received the Ph.D. in Computer Science from South China University of Technology, Guangzhou, China in 2013. He visited the Data Mining Laboratory at George Mason University, VA, USA for 2 years from 2011 to 2013, he was a Postdoc Research Fellow in the Department of Computer Science, Hong Kong Baptist University, Hong Kong from 2014 to 2015. His current research interests include data mining and bioinformatics. He serves as PC member for KDD, CDM, SDM and other conferences. He is a recipient of Best Poster Award of SDM2012 Doctral Forum and Best Student Paper Award of IEEE International Conference on Machine Learning and Cybernetics, 2011.