Transfer Clustering Ensemble Selection

Yifan Shi, Zhiwen Yu, *Senior Member, IEEE*, C. L. Philip Chen, *Fellow, IEEE*, Jane You, Hau-San Wong, Yide Wang, and Jun Zhang, *Fellow, IEEE*

***Abstract*—Clustering ensemble (CE) takes multiple clustering solutions into consideration in order to effectively improve the accuracy and robustness of the final result. To reduce redundancy as well as noise, a CE selection (CES) step is added to further enhance performance. Quality and diversity are two important metrics of CES. However, most of the CES strategies adopt heuristic selection methods or a threshold parameter setting to achieve tradeoff between quality and diversity. In this paper, we propose a transfer CES (TCES) algorithm which makes use of the relationship between quality and diversity in a source dataset, and transfers it into a target dataset based on three objective functions. Furthermore, a multiobjective self-evolutionary process is designed to optimize these three objective functions. Finally, we construct a transfer CE framework (TCE-TCES) based on TCES to obtain better clustering results. The experimental results on 12 transfer clustering tasks obtained from the 20newsgroups dataset show that TCE-TCES can find a better tradeoff between quality and diversity, as well as obtaining more desirable clustering results.**

***Index Terms*—Clustering ensemble selection (CES), machine learning, multiobjective, transfer learning.**

I. Introduction

WITH the successful application of ensemble theory in the domain of classification, such as Adaboost [1], GBDT [2], Randomforest [3], andso on [52]–[55], [73], researchers are turning their attention to clustering ensemble (CE). Different CE approaches have been widely used in the area of bioinformatics [4]–[6], multimedia [7], and pattern recognition [8]–[16]. These research works show that CE strategies contribute to better and more robust results due to the fusing of multiple base clustering members strategically.

According to Kleinberg's theorem, it is not possible to find a single clustering algorithm suitable for all datasets [17]. Inspired by the idea of ensemble learning, the CE framework is proposed [18]. It mainly consists of two steps: 1) clustering members generation and 2) ensemble consensus. In the first step, diverse clustering members are generated based on different techniques, for example, different single clustering algorithms, random initialization, resampling [19]–[21], and feature subspaces [22]. We usually require the clustering members to be as diverse as possible since we can obtain more information from distinct views. In the second step, all clustering members in step one are taken into consideration, and a consensus function is designed to produce the final clustering result. A number of consensus methods have been proposed based on different perspectives to observe clustering members. Voting-based methods, which are directly inherited from ensemble classification, are the most straightforward techniques for result aggregation, expect for the need of an alignment process. Zhou and Tang [23] proposed four types of voting strategies: 1) simple voting; 2) weighted-voting; 3) selective voting; and 4) selective weighted-voting. Co-association matrix-based methods have been designed by Fred and Jain [24]. For these types of method, a co-association matrix extracts the association information of two data points in the same cluster from the clustering members, then hierarchical agglomerative clustering algorithm is applied to the matrix to obtain the final result. Graph-based methods also take advantage of the co-association matrix to construct a hypergraph whose nodes are data points or clusters, and the graph cut algorithm can be applied for partitioning the nodes. CSPA, HGPA, and MCLA are three graph-based consensus algorithms proposed by Strehl and Ghosh [18]. They produce hypergraphs in different ways, and METIS is used as the final graph cut algorithm.

In later studies, researchers found that better clustering results could be obtained without using all clustering members [26]–[35], because redundancy and noise also exist in these members. Redundancy not only affects the ensemble efficiency but also leads us to ignore other clustering members' contributions. The presence of noise also disturbs the intrinsic structure of the dataset and degrades the performance.

Manuscript received June 16, 2018; accepted December 3, 2018. Date of publication December 25, 2018; date of current version May 7, 2020. This work was supported in part by the Guangdong Province Higher Vocational Colleges and Schools Pearl River Scholar Funded Scheme 2018, in part by NSFC under Grant 61722205, Grant 61751205, Grant 61751202, Grant 61572199, Grant 61502174, and Grant 61872148, in part by the Natural Science Foundation of Guangdong Province under Grant 2017A030313355, in part by the Guangzhou Science and Technology Planning Project under Grant 201704030051, in part by the Research Grants Council of the Hong Kong under Grant CityU 11300715, in part by the Hong Kong General Research under Grant 152202/14E, and in part by the Hong Kong Polytechnic University under Grant G-YM05 and Grant G-YN39. This paper was recommended by Associate Editor P. N. Suganthan. *(Corresponding author: Zhiwen Yu.)*

Y. Shi, Z. Yu, and J. Zhang are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: zhwyu@scut.edu.cn).

C. L. P. Chen is with the Department of Computer and Information Science, University of Macau, Macau 999078, China.

J. You is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong.

H.-S. Wong is with the Department of Computer Science, City University of Hong Kong, Hong Kong.

Y. Wang is with the Department of Electronic and Digital Technology, Ecole Polytechnique de l'Universite de Nantes, 44000 Nantes, France.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TCYB.2018.2885585

As a result, a CE selection (CES) step is proposed to solve these two problems [26], [27]. Instead of using all clustering members which may contain redundant members, CES selects a subset of clustering members before the consensus step. It is obvious that how to evaluate a subset of clustering members is the main consideration of a CES algorithm. Quality and diversity [25] are two of the most common metrics. They indicate the degree of agreement and disagreement between clustering members in the subset, respectively. Higher quality indicates that the data partition is generally agreed to by clustering members, while higher diversity can take advantage of different types of contributions. As a result, how to determine the tradeoff relationship between quality and diversity becomes a challenge. Previously, this relationship was chosen according to experience or specific parameter settings [14], [25]–[30], which is too dependent on the datasets. Based on the above observation, an algorithm that can adaptively search for the suitable tradeoff relationship is desired.

In traditional clustering tasks, all the data come from the same domain. There is not sufficient information to obtain the tradeoff relationship between quality and diversity in an unsupervised environment. In order to solve the problem of information deficiency, transfer clustering is proposed [37]. Unlike traditional clustering approaches, transfer clustering takes the data in similar domains into consideration. It aims at learning clustering knowledge from a source domain with or without supervision, and then transferring the knowledge to a target domain. However, to the best of our knowledge, almost all the existing transfer clustering algorithms focus on the improvement of single clustering algorithms, and transfer CE has not been studied.

In this paper, the idea of transfer learning is utilized. We consider selecting clustering member subset in the target domain with the help of the dataset in the source domain. Then three objective functions are formulated as transfer objectives, and a multiobjective self-evolutionary process (MOSEP) is designed to optimize them. Finally, we construct a transfer CE framework (TCE-TCES) based on the transfer CES (TCES) algorithm. The main contributions of this paper are as follows.

1) We propose a TCES algorithm that can learn the tradeoff relationship between quality and diversity, and adaptively select the clustering members.
2) To our best knowledge, we are the first to combine transfer learning with CE to obtain better clustering results.
3) We design an MOSEP to optimize the selection of clustering members.
4) A transfer CE framework is constructed based on TCES and tested on 12 transfer clustering tasks.

The rest of this paper is organized as follows. Section II introduces the related work on CES and transfer clustering. Section III presents the proposed TCE-TCES framework. Section IV applies a series of experiments on 20newsgroups (20NG) to verify the performance of the proposed method. Section V concludes the proposed work and provides some future research directions.

II. Related Work

Given a dataset $X = \{x_1, \ldots, x_n\}$ that has n data points for a CE task. The clustering members generation step applies base clustering algorithms with different strategies to X to produce a clustering member set $\Pi = \{\pi_1, \ldots, \pi_m\}$. Each clustering member π_i represents a partition for X, where π_{ij} denotes that the jth data point is assigned to the ith partition. The aim of the CES step is to choose a subset $\Pi^* = \{\pi_1, \ldots, \pi_{m^*}\}(\Pi^* \subseteq \Pi, m^* \leq m)$ that can be integrated to produce a better clustering result.

The key problem of CES is how to evaluate the member subset. Quality and diversity-based methods [14], [25]–[32] are the most common ones. First, NMI [18] or ARI [46] is used to measure the similarity between two clustering members without an alignment process. The normalized mutual information $\mathrm{NMI}(\pi, \pi')$ is defined as follows:

$$\mathrm{NMI}(\pi, \pi') = \frac{2\varphi(\pi, \pi')}{\varphi(\pi) + \varphi(\pi')} \tag{1}$$

$$\varphi(\pi, \pi') = \sum_{h=1}^{k}\sum_{l=1}^{k'} \frac{C_h \bigcap C'_l}{n} \log \frac{n|C_h \bigcap C'_l|}{|C_h||C'_l|} \tag{2}$$

$$\varphi(\pi) = -\sum_{h=1}^{k} \frac{|C_h|}{n} \log \frac{|C_h|}{n}. \tag{3}$$

The adjusted Rand index $\mathrm{ARI}(\pi, \pi')$ is defined as follows:

$$\mathrm{ARI}(\pi, \pi') = \frac{\sum_{h=1}^{k}\sum_{l=1}^{k'} \binom{|C_h \bigcap C'_l|}{2} - \xi_3}{0.5(\xi_1 + \xi_2) - \xi_3} \tag{4}$$

$$\xi_1 = \sum_{h=1}^{k} \binom{|C_h|}{2},\ \xi_2 = \sum_{l=1}^{k'} \binom{|C'_l|}{2}, \xi_3 = \frac{2\xi_1\xi_2}{n(n-1)} \tag{5}$$

where π and π' are two clustering members. k and k' are the number of clusters in π and π', respectively. C_h denotes the data points in the hth cluster of π, while C'_l denotes the data points in the lth cluster of π'. $|\cdot|$ denotes the number of data points. We can see that $\mathrm{NMI}(\pi, \pi') \in [0, 1]$ and $\mathrm{ARI}(\pi, \pi') \in [-1, 1]$, the greater $\mathrm{NMI}(\pi, \pi')$ or $\mathrm{ARI}(\pi, \pi')$ is, the more similar π and π' are.

Next, $\mathrm{NMI}(\pi, \pi')$ or $\mathrm{ARI}(\pi, \pi')$ is calculated to serve as a similarity measure $\mathrm{Sim}(\pi, \pi')$. Quality and diversity are defined based on NMI or ARI according to [25] as follows:

$$\mathrm{Quality}(\Pi^*) = \sum_{i=1}^{m^*} \mathrm{SNMI}(\pi_i, \Pi) \tag{6}$$

$$\mathrm{SNMI}(\pi_i, \Pi) = \sum_{j=1\&j\neq i}^{m} \mathrm{Sim}(\pi_i, \pi_j) \tag{7}$$

$$\mathrm{Diversity}(\Pi^*) = \sum_{\pi_i \neq \pi_j \& \pi_i, \pi_j \in \Pi^*} \left(1 - \mathrm{Sim}(\pi_i, \pi_j)\right). \tag{8}$$

Now, we can evaluate a clustering member subset in two ways. On the one hand, a high quality subset indicates that most of the members agree with the result. On the other hand, a subset with high diversity can absorb a wide variety of views. Therefore, there were different CES strategies according to different understandings in previous works. Kuncheva and Hadjitodorov [26] studied the diversity within a CE, and observed that a higher diversity leads to a better consensus result. Hadjitodorov *et al.* [27] observed two trends of the relationship between diversity and accuracy in their experiments. In order to achieve an overall better performance, they suggested choosing a moderate level of diversity. Fern and Lin [25] were the first to consider the tradeoff relationship between quality and diversity, and they proposed three CES strategies based on this relationship. Azimi and Fern [28] designed a threshold-based selection method, which compared the similarity between clustering members and the full ensemble result. Then a clustering member subset was chosen based on a preset similarity threshold. According to Hong *et al.* [29], multiple subset ensemble results obtained from resampling were used as indicators to measure the quality and diversity, and clustering members were selected through a threshold as well. Jia *et al.* [30] took advantage of spectral clustering (SC) to generate clustering members, and sorted them according to their similarity to the temporary ensemble results obtained through a bagging technique. Naldi *et al.* [14] used six relative clustering validity indices to evaluate the quality, and integrated them to create a final evaluation. Wang *et al.* [31] used rough set theory to find a better clustering member subset. Lu *et al.* [32] defined a new diversity measure based on covariance, and proposed a covariance-based selective CE algorithm. Akbari *et al.* [35] first computed a pair-wise diversity measure matrix, then applied hierarchical clustering algorithm to take both quality and diversity into consideration. Yang *et al.* [59] designed a unified framework that simultaneously considered the quality and diversity under the prior background knowledge of pairwise constraints. Zhao *et al.* [60] utilized multiple indices to measure the quality and diversity, then the clustering member subset is iteratively expanded based on the previous one to maximize these indices. In addition to the quality-diversity-based selection strategies, there are other methods. Alizadeh *et al.* [33], [71], Nazari *et al.* [70], and Parvin and Minaei-Bidgoli [68], [69] designed several methods to select clusters instead of clustering members. Yu *et al.* [34] viewed the clustering members as new features of the data points, and they applied four feature selection methods to obtain the final subset. Yu *et al.* [61] also designed a distribution-based method to measure the similarity between two clustering members, and further utilized it to select the subset of clustering members. To overcome the weakness of quality, Yousefnezhad *et al.* [62] proposed an independency index to be applied. Alizadeh *et al.* [67] combined the idea of wisdom of crowds into the selection process. Minaei-Bidgoli *et al.* [72] adapted the resampling schemes to select the clustering members. Instead of the hard selection strategies mentioned above, there are soft selection methods based on weighting the members. Zhou and Tang [23] reduced the weight of clustering members that were similar to other members. Li and Ding [63] proposed a consensus approach that integrated the weighted clustering members based on the NMF framework [64].

Previous CES methods are mostly based on quality or diversity, and we can summarize them as follows: 1) approaches that only use quality [14], [30]; 2) approaches that only use diversity [26], [27], [31], [32]; and 3) approaches that combine quality with diversity [25], [28], [29], [35], [59], [60]. We can see that only some of them consider both quality and diversity. Furthermore, they determine the tradeoff relationship between quality and diversity according to heuristic techniques, and specific parameter or threshold settings. These methods cannot be easily adapted to a variety of complex datasets because some datasets prefer higher quality while others need more diversity. As a result, how to search for the suitable tradeoff relationship between quality and diversity becomes a challenge.

To address this challenge, we take advantage of the idea of transfer learning. Traditional machine learning only utilizes data in a target domain, while transfer learning have access to data in similar domains [36]. Transfer learning in classification, feature extraction and regression has been extensively studied in a number of previous works, but there is a few research on transfer clustering [36]. Dai *et al.* [37] have proposed a self-taught clustering (STC) algorithm, which clusters data points in a target domain with the help of a large number of unlabeled data points in a source domain. STC assumes that both domains share the same feature clustering, and it minimizes the MI value between data points and features to achieve improved results on the target domain. Jiang and Chung [38] improved SC through a transfer learning formulation [transfer SC (TSC)]. TSC looks for potential data manifolds of both domains, and uses co-clustering to perform knowledge transfer. Yu *et al.* [39] designed a topic-constraint transfer clustering (TCTC) algorithm for text information transfer. TCTC attempts to extract must-link constraints from the potential topics in both domains, and a semisupervised clustering approach is adopted. Samanta *et al.* [40] constructed two iterative frameworks for knowledge transfer. Sun *et al.* [41] improved the maximum entropy clustering (MEC) algorithm through transfer learning, which is referred to as TL_MEC, which corrects clusters in the target domain by the historical cluster centers and clustering membership degree values in the source domain. Deng *et al.* [42] proposed transfer fuzzy *C*-means (TFCM) and transfer fuzzy subspace clustering (TFSC), which incorporates clustering information of the source domain into the original FCM and FSC objective functions to guide the prototypes of the target domain. Qian *et al.* [51] proposed a new framework which jointly leverages specific cluster prototypes and fuzzy membership values to improve the effectiveness and robustness of the classical MEC approach.

At present, the research of transfer learning is focused on the improvement of traditional single clustering algorithms, and the area of transfer CE has not been sufficiently studied. Compared with a single clustering algorithm, the ensemble method has better robustness and accuracy [47], [48]. Therefore, transfer CE is worthy of research.

In this paper, we propose a transfer CE framework based on a new ensemble selection strategy, which combines the idea of transfer learning with CE. TCES attempts to achieve the

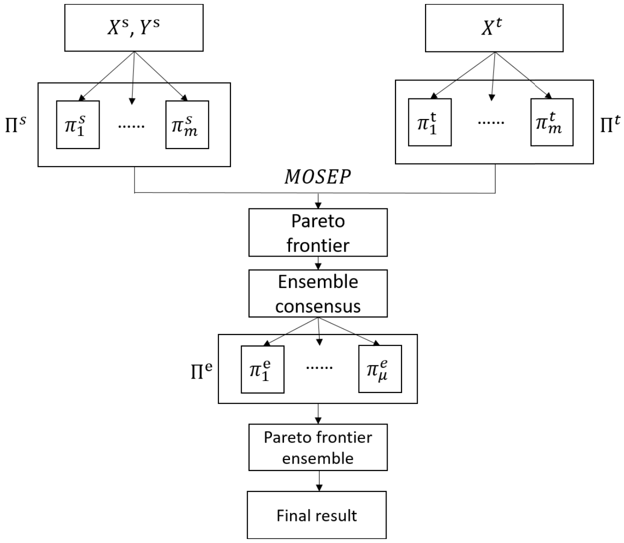


Fig. 1. Overview of the TCE-TCES.

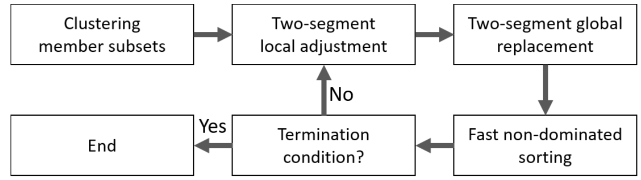


Fig. 2. Overview of the MOSEP for the clustering member subsets.

tradeoff between quality and diversity, and it can adaptively find out the suitable relationship instead of using an artificial setting. TCES not only improves the performance of CES but also explores the combination between transfer learning with CE. Besides, we design an MOSEP to optimize the selection of clustering members, which can find the suitable member subset of both domains simultaneously.

III. Transfer Clustering Ensemble Selection

Among the different CES strategies, quality and diversity are two important factors to be considered [14], [25]–[32], [35]. However, determining the suitable tradeoff relationship between these two factors according to human experience or parameter settings lacks adaptability when applied to different datasets. In addition, using a greedy algorithm cannot fully optimize the clustering member subset search process. With the rapid development of machine learning, a large number of related datasets have been accumulated in various fields. We consider taking advantage of the idea of transfer learning to utilize the data in a source domain to guide the clustering member subset selection in a target domain. Fig. 1 provides an overview of the TCE-TCES. TCE-TCES first generates clustering members in both domains, and then applies an MOSEP to optimize three objective functions for achieving the tradeoff relationship. After obtaining the Pareto frontier, each clustering member subset in the frontier is integrated. Finally, two Pareto frontier ensemble methods are proposed to obtain the final result.

We first describe the clustering task for TCE-TCES. A source domain dataset $X^s = \{x_1^s, \ldots, x_{n^s}^s\}$ and a target domain dataset $X^t = \{x_1^t, \ldots, x_{n^t}^t\}$ are given with n^s and n^t data points, respectively. The source feature space $\mathcal{F}^s$ contains n^{fs} features, while the target feature space $\mathcal{F}^t$ contains n^{ft} features. In other words, $X^s \in \mathcal{R}^{n^{fs} \times n^s}$ and $X^t \in \mathcal{R}^{n^{ft} \times n^t}$. Meanwhile, the label of X^s is Y^s and c is the number of clusters in both domains. TCE-TCES aims at obtaining a better ensemble clustering result in X^t with the help of X^s.

The first step of TCE-TCES is the generation of clustering member sets $\Pi^s = \{\pi_1^s, \ldots, \pi_m^s\}$ and $\Pi^t = \{\pi_1^t, \ldots, \pi_m^t\}$ (m is the number of clustering members) in the source and target domains, respectively. To obtain more comprehensive clustering members, K-means (KM) and SC [43] are used as base clustering algorithms. KM iteratively improves the cluster centers by directly utilizing the distance between data points, while SC is based on graph theory, and its purpose is to find the optimal subgraph partition that divides the graph into several parts. These two approaches partition the dataset from different perspectives and provide more comprehensive base clustering results. Besides, the strategies of using random initialization and random numbers of clusters in the range of $[c, 2c]$ are adopted for generating more diversified results.

The second step of TCE-TCES focuses on finding an optimal clustering member subset $\Pi_*^t = \{\pi_1^t, \ldots, \pi_{m^{t*}}^t\} \subseteq \Pi^t$. Considering the idea of transfer learning, we utilize the data in similar domains to guide the CES step in the target dataset. Although the distributions of the data in the source and target domains are different, we assume from a more abstract level that the quality and diversity distributions of the clustering members are similar because these data have similar clustering properties. Therefore, we may suppose that the tradeoff relationship of quality and diversity of a good clustering member subset in the source domain can be transferred to that in the target domain. Three objective functions are designed for the transfer task.

We intend to transfer the tradeoff relationship of a good clustering member subset in the source domain to the target domain, so the first objective function measures the performance of the selected subset $\Pi_*^s = \{\pi_1^s, \ldots, \pi_{m^{s*}}^s\} \subseteq \Pi^s$ in the source domain. TCE-TCES integrates Π_*^s using a consensus method in Algorithm 2, and a temporary ensemble result π^{se} is obtained. The performance of Π_*^s is indicated by the NMI value between π^{se} and Y^s, and the better the performance is, the higher the NMI value becomes. We subtract the NMI value from 1 and minimize the following objective function:

$$\min \quad F_1 = 1 - \mathrm{NMI}\left(\pi^{se}, Y^s\right). \tag{9}$$

The second objective function minimizes the difference between the quality indices of the selected subsets in the source and target domains. Considering that the number of clustering members selected in both domains may be distinct, and the quality measures defined in (6) is not normalized, it is difficult to compare the difference. Due to this limitation, we propose an adjusted quality index as follows:

$$\text{quality\_adj}(E) = \frac{\sum_{\pi \in E} \mathrm{SNMI}(\pi, \Pi)}{|E| \cdot Z} \tag{10}$$

$$Z = \max\left\{\mathrm{SNMI}(\pi_1, \Pi), \ldots, \mathrm{SNMI}\left(\pi_{|E|}, \Pi\right)\right\} \tag{11}$$

where E is the clustering member set, $|\cdot|$ stands for the size of the set, and Z is the normalization factor to ensure quality_adj$(E) \in [0, 1]$. Next, the difference between the quality indices of the selected subsets in both domains can be defined based on the adjusted quality index as follows:

$$\begin{aligned}\min F_2 &= \left|\mathrm{quality\_adj}\left(E^s\right) - \mathrm{quality\_adj}\left(E^t\right)\right| \\ &= \left|\frac{\sum_{\pi \in E^s} \mathrm{SNMI}(\pi, \Pi^s)}{|E^s| \cdot |Z^s|} - \frac{\sum_{\pi \in E^t} \mathrm{SNMI}\left(\pi, \Pi^t\right)}{|E^t| \cdot |Z^t|}\right|\end{aligned} \tag{12}$$

where E^s and E^t are the selected clustering member subsets in the source and target domains, respectively. When the qualities of E^s and E^t are similar, F_2 is close to 0.

The third objective function minimizes the difference between the diversity indices of the selected subsets in the source and target domains. For the same reason of incompatible numbers of clustering members and normalization problem, we redefine the diversity as follows:

$$\mathrm{diversity\_adj}(E) = \frac{\sum_{i \neq j \& i,j \leq |E|} \mathrm{NMI}\left(\pi_i, \pi_j\right)}{0.5 \times |E| \times (|E| - 1)}. \tag{13}$$

We can see that diversity_adj(E) also ranges in [0, 1], and we define the final objective function as follows:

$$\begin{aligned}\min\ F_3 &= \left|\mathrm{diversity\_adj}\left(E^s\right) - \mathrm{diversity\_adj}\left(E^t\right)\right| \\ &= \left|\frac{\sum_{i \neq j \& i,j \leq |E^s|} \mathrm{NMI}\left(\pi_i, \pi_j\right)}{0.5 \times |E^s| \times (|E^s| - 1)} - \frac{\sum_{i \neq j \& i,j \leq |E^t|} \mathrm{NMI}\left(\pi_i, \pi_j\right)}{0.5 \times |E^t| \times (|E^t| - 1)}\right|.\end{aligned} \tag{14}$$

Obviously, the smaller F_3 is, the more similar the diversity of E^s and E^t are.

These three objective functions transfer the quality and diversity relationship and guide the choice of the subset in the target domain, while selecting a subset in the source domain with good performance. However, it is difficult to directly optimize these objective functions at the same time. Instead of finding a single optimal subset, we attempt to obtain the Pareto frontier where none of the subsets performs better than others on all three objective functions. Therefore, we design an MOSEP which iteratively optimizes the three objective functions, while selecting member subsets of both domains. Fig. 2 provides an overview of the MOSEP. It consists of five steps to optimize the member subsets with the three objective functions.

For representing member subsets, MOSEP uses one-hot coding for both domains, then concatenates them as follows:

$$\Phi = \left\{\phi_1^s, \ldots, \phi_m^s, \phi_1^t, \ldots, \phi_m^t\right\} \tag{15}$$

$$\phi_i^s = \begin{cases} 1, & \text{if } \pi_i^s \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$\phi_j^t = \begin{cases} 1, & \text{if } \pi_j^t \text{ is selected} \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The size of Φ is 2 m, where the first half of the components represents a member subset in the source domain, while the rest of the components represents a member subset in the target domain. MOSEP first generates O different member subsets $\Psi = \{\Phi_1, \ldots, \Phi_O\}$ with a random initialization strategy. Next, two operations, which are referred to as two-segment local adjustment and two-segment global replacement, are applied.

For the two-segment local adjustment operation, a subset Φ is randomly selected. Then two random index sets denoted as $\Upsilon^s = \{r_1^s, \ldots, r_{\kappa^s}^s\}$ and $\Upsilon^t = \{r_1^t, \ldots, r_{\kappa^t}^t\}$ are generated with replacement, where r_i^s and r_j^t are random integers in $[1, m]$, and κ^s and κ^t are random integers in $[1, (m/2)]$. A new subset Φ' is generated according to the following operations:

$$\begin{aligned}{\phi'}_i^s &= \begin{cases} 1 - \phi_i^s, & \text{if } i \in \Upsilon^s \\ \phi_i^s, & \text{otherwise} \end{cases} \\ {\phi'}_j^t &= \begin{cases} 1 - \phi_j^t, & \text{if } i \in \Upsilon^t \\ \phi_j^t, & \text{otherwise.} \end{cases}\end{aligned} \tag{18}$$

The two-segment local adjustment operation is iteratively applied until $(O/2)$ new subsets are generated.

For the two-segment global replacement operation, two subsets Φ_a and Φ_b are randomly selected as follows:

$$\begin{aligned}\Phi_a &= \left\{\phi_{a1}^s, \ldots, \phi_{ai}^s, \ldots, \phi_{am}^s, \phi_{a1}^t, \ldots, \phi_{aj}^t, \ldots, \phi_{am}^t\right\} \\ \Phi_b &= \left\{\phi_{b1}^s, \ldots, \phi_{bi}^s, \ldots, \phi_{bm}^s, \phi_{b1}^t, \ldots, \phi_{bj}^t, \ldots, \phi_{bm}^t\right\}.\end{aligned} \tag{19}$$

Then two replacement indices r_i^s and r_j^t are randomly chosen in the interval $[1, m]$. The global replacement operations in the index intervals $[r_i^s, m]$ and $[r_j^t, m]$ result in the following two new subsets:

$$\begin{aligned}{\Phi'}_a &= \left\{\phi_{a1}^s, \ldots, \phi_{bi}^s, \ldots, \phi_{bm}^s, \phi_{a1}^t, \ldots, \phi_{bj}^t, \ldots, \phi_{bm}^t\right\} \\ {\Phi'}_b &= \left\{\phi_{b1}^s, \ldots, \phi_{ai}^s, \ldots, \phi_{am}^s, \phi_{b1}^t, \ldots, \phi_{aj}^t, \ldots, \phi_{am}^t\right\}.\end{aligned} \tag{20}$$

The two-segment global replacement operation is iteratively applied until $(O/2)$ new subsets are generated.

After the two-segment local adjustment and two-segment global replacement operations, there are totally $2O$ clustering member subsets combining the original subsets with the newly generated subsets, which can be denoted as $\Psi_{\mathrm{candidate}} = \{\Phi_1, \ldots, \Phi_O, {\Phi'}_1, \ldots, {\Phi'}_O\}$. Next, MOSEP applies the fast nondominated sorting algorithm [45] to stratify $\Psi_{\mathrm{candidate}}$ according to the dominance relationship, then crowded comparison and elitist strategy are used to select O subsets to form a new Ψ with overall better performance. The subsets assigned to the first layer during the fast nondominated sorting process means that none of them is dominated by others. The subsets in the second layer mean that each of them is dominated by only one subset in the first layer, and the rest of the layers are determined in a similar way. The fast nondominated sorting algorithm sorts the subsets according to the crowding distance defined in [45] layer by layer, then the first O subsets are chosen to form Ψ.

MOSEP repeats the above three operations until the preset maximum number of iterations is reached. In the last iteration, we choose the subsets in the first layer after fast nondominated sorting to form the Pareto frontier $P = \{\Phi_1, \ldots, \Phi_\mu\}$, where

Algorithm 1 MOSEP

Require:

Input: the clustering members in the source domain $\Pi^s = \{\pi_1^s, ..., \pi_m^s\}$;
the clustering members in the target domain $\Pi^t = \{\pi_1^t, ..., \pi_m^t\}$;
the ground truth clustering labels in the source domain Y^s;
the maximum number of iterations β

Ensure:

1: Randomly generate O clustering member subsets $\Psi = \{\Phi_1, ..., \Phi_O\}$ according to (15);
2: **For** $t = 1$ to β
3: Set newly generated subsets $\Psi' = \emptyset$;
4: **Repeat**
5: Generate two random index sets Υ^s and Υ^t;
6: Produce a new subset Φ' according to (18);
7: **Until** $\frac{O}{2}$ new subsets are added to Ψ';
8: **Repeat**
9: Randomly select two subsets Φ_a and Φ_b from Ψ;
10: Randomly generate two replacement indices r_i^s and r_j^t;
11: Produce two new subsets Φ'_a and Φ'_b according to (20);
12: **Until** $\frac{O}{2}$ new subsets are added to Ψ';
13: Obtain candidate clustering member subsets $\Psi_{candidate} = \Psi \cup \Psi'$;
14: **If** $t \neq \beta$
15: Apply the fast non-dominated sorting algorithm on $\Psi_{candidate}$, choose the first O subsets to form Ψ;
16: **Else**
17: Apply the fast non-dominated sorting algorithm on $\Psi_{candidate}$ to obtain the Pareto frontier $P = \{\Phi_1, ..., \Phi_\mu\}$;

Output: Pareto frontier $P = \{\Phi_1, ..., \Phi_\mu\}$.

each subset is one of the optimal solutions under the proposed three objective functions. Algorithm 1 shows the complete process of MOSEP.

After the Pareto frontier is obtained, each Φ_i contains a selected clustering member subset in the target domain. A consensus ensemble method based on a hypergraph is applied to the Pareto frontier to obtain μ ensemble results $\Pi^e = \{\pi_1^e, \ldots, \pi_\mu^e\}$. Given a clustering member subset $\Pi = \{\pi_1, \ldots, \pi_\iota\}$, TCE-TCES uses the method proposed by Strehl and Ghosh [18] to obtain the similarity matrix. TCE-TCES constructs the binary membership indicator matrix $M_i \in \mathbb{N}^{n\times k_i}$ (k_i is the number of clusters) for each π_i whose elements m_{ab} are defined as follows:

$$m_{ab} = \begin{cases} 1, & \text{if } x_a \in C_b \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

where C_b is the bth cluster in π_i. The adjacency matrix of a hypergraph with n vertices and $\sum_{t=1}^{\iota} k_t$ hyperedges is defined as $M = (M_1 \cdots M_\iota)$. Then the similarity matrix can be calculated as follows:

$$W = \frac{1}{\iota} M M^T \tag{22}$$

Algorithm 2 Consensus Ensemble Method Based on a Hypergraph

Require:

Input: the clustering member subset $\Pi = \{\pi_1, ..., \pi_\iota\}$;
the dataset in the target domain X^t;
number of clusters c;

Ensure:

1: **For** each π_i in Π
2: Construct a binary membership indicator matrix M_i according to (21);
3: Construct adjacency matrices $M = (M_1 ... M_\iota)$;
4: Calculate a similarity matrix W according to (22);
5: Construct a hypergraph $G = (X^t, W)$;
6: Obtain the graph cuts for G using the Ncut algorithm;

Output: consensus ensemble result π^e.

where the element w_{ij} in W is the weight between x_i and x_j. Based on X^t and W, a hypergraph is constructed as $G = (X^t, W)$. Specifically, the vertices of G represent data points in X^t, while the edges of G specify the correlation between two data points. The relaxation of normalized cut (Ncut) algorithm [44], [56], [57] can be applied to partition the graph G into c parts, which solves the first k eigenvectors of $Lu = \lambda Du$ (where L is the unnormalized Laplacian and D is the degree matrix) to obtain new data representation, and applies KM to obtain the clustering result.

After the consensus ensemble step, the ensemble results $\Pi^e = \{\pi_1^e, \ldots, \pi_\mu^e\}$ are obtained. We propose two Pareto frontier ensemble strategies to obtain the final clustering result from Π^e as follows.

1) Considering that the ensemble results in Π^e are reliable, we choose the one with the highest agreement as the final clustering result, which is denoted as

$$\pi^* = \arg\max_{\pi \in \Pi^e} \text{SNMI}\big(\pi, \Pi^e\big). \tag{23}$$

2) We can view the ensemble results in Π^e as new base clustering members, then perform a double consensus ensemble step on Π^e by using Algorithm 2. The final clustering result is denoted as

$$\pi^* = \text{ConsensusEnsemble}(\Pi^e). \tag{24}$$

IV. Theoretical Analysis

We also perform a theoretical analysis of TCE-TCES with respect to its computational cost. Assume that the numbers of data points in the source and target domains are both n, and the numbers of features are both f for concise description of complexity analysis. We divide the time complexity into three components: T_{GCM}, T_{MOSEP}, and T_{PFE}. They denote the step of generating clustering members, the step of MOSEP, and the step of Pareto frontier ensemble, respectively.

T_{GCM} includes the time complexity of KM and SC, which can be expressed as

$$T_{\text{GCM}} = M T_{\text{KM}} + M T_{\text{SC}} \tag{25}$$

where M is the number of clustering members generated in each domain. The time complexity of KM is $O(G_1 C n f)$, where

G_1 stands for the number of iterations and C denotes the number of clusters. Because M, G_1 and C are significantly smaller than n and f, MT_{KM} can be approximated as $O(nf)$. The time complexity of SC is $O(n^3)$. As a result, the computational cost for the step of generating clustering members is $O(n^3 + nf)$.

Before analyzing T_{MOSEP} and T_{PFE}, we consider the time complexity of ensemble consensus with Ncut T_{ES} which is $O(n^3)$. To reduce the computational cost, it can be realized by transforming into weighted KM clustering according to the SEC algorithm proposed by Liu *et al.* [58]. Based on the technique of SEC, the time complexity of ensemble consensus can be reduced to $T_{\mathrm{ES}} = O(n)$.

T_{MOSEP} includes the step of initial generation T_{IG}, two-segment local adjustment T_{TLA}, two-segment global replacement T_{TGR}, calculation of objective functions T_{COF}, and fast nondominated sorting T_{FNS}, which can be expressed as

$$T_{\mathrm{MOSEP}} = T_{\mathrm{IG}} + G_2(T_{\mathrm{IG}} + T_{\mathrm{TLA}} + T_{\mathrm{TGR}} + T_{\mathrm{COF}} + T_{\mathrm{FNS}}). \tag{26}$$

G_2 is the maximum number of iterations in MOSEP. T_{IG}, T_{TLA}, T_{TGR}, and T_{FNS} are all $O(1)$ because they are only related to M, S (the number of clustering member subsets) and B (the number of objective functions) which are significantly smaller than n and f. T_{COF} contains the cost of computing three objective functions, which can be expressed as

$$T_{\mathrm{COF}} = S(T_{F1} + T_{F2} + T_{F3}). \tag{27}$$

T_{F1} consists of two steps: 1) integrating the selected source clustering members and 2) computing the NMI value, which are both $O(n)$, so $T_{F1} = O(n)$. For T_{F2} and T_{F3}, the NMI value between each pair of clustering members can be precomputed to improve the efficiency to $O(S^2)$. Therefore, $T_{\mathrm{COF}} = O(S(n + S^2)) = o(n)$ and the time complexity of MOSEP is $O(n)$.

T_{PFE} is related to the time complexity of ensemble consensus T_{EC} and the Pareto frontier ensemble strategy T_{PFES}, which is expressed as

$$T_{\mathrm{PFE}} = UT_{\mathrm{EC}} + T_{\mathrm{PFES}} \tag{28}$$

where U is the number of clustering member subsets in the Pareto frontier, which is also significantly smaller than n and f. The time complexity of the Pareto frontier ensemble strategy 1 is $O(Un) = O(n)$ for computing the NMI values, while the complexity of strategy 2 is $O(n)$ as well for the ensemble consensus. As a result, the computational cost $T_{\mathrm{PFE}} = O(Un + n) = O(n)$.

In summary, the time complexity of TCE-TCES can be estimated as follows:

$$\begin{aligned} T_{\mathrm{TCE-TCES}} &= T_{\mathrm{GCM}} + T_{\mathrm{MOSEP}} + T_{\mathrm{PFE}} \\ &= O\left(n^3 + nf\right) + O(n) + O(n) = O\left(n^3 + nf\right). \end{aligned} \tag{29}$$

It can be noticed that the main calculation is focused on the generation of the base clustering members, which can be easily parallelized to reduce the computational time.

TABLE I
OVERVIEW OF THE 20NG DATASET

Top-level classes	Secondary-level classes
comp	comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x
rec	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey
sci	sci.crypt sci.electronics sci.med sci.space
talk	talk.politics.guns talk.politics.mideast talk.politics.misc talk.religion.misc
alt	alt.atheism
misc	misc.forsale
soc	soc.religion.christian

V. EXPERIMENT

In this section, the performance of the proposed TCE-TCES is evaluated using 12 transfer clustering tasks obtained from 20NG[1] which is preprocessed by Cardoso-Cachopo [50]. As shown in Table I, 20NG divides all the news texts into seven top-level classes: alt, comp, misc, rec, sci, talk, and soc. Furthermore, classes comp, rec, sci, and talk are subdivided into several secondary-level classes. We use the method similar to Jiang and Chung [38] to construct the 12 transfer clustering tasks as shown in Table II. Tasks C2-1 to C2-6 have two clusters, tasks C3-1 to C3-4 have three clusters, while tasks C4-1 and C4-2 have four clusters. Four top-level classes (comp, rec, sci, and talk) are used, then the secondary-level classes can be divided into two similar domains. Taking task C2-1 as an example, it aims at clustering data into two top-level classes (comp and sci) in the target domain with the help of data in the source domain. Besides, the secondary-level classes in the source domain (comp.graphics + comp.os.ms-windows.misc and sci.crypt + sci.electronics) belong to the same top-level classes as the secondary-level classes in the target domain (comp.sys.ibm.pc.hardware + comp.windows.x and sci.med + sci.space), so their domains are similar to each other.

20NG contains a total of 18 821 texts, 60% of them are included in the training set, while the rest are included in the testing set. We only use the training set in our experiments. The vector space model is applied, in which each term represents a feature in the text vector. The feature weights are computed using TF-IDF and l2-normalization is performed. In order to preprocess the data for filtering infrequent terms, we only retain the terms with df $\geq$ 5 (df is the document frequency). The data composition for each task is shown in Table III, where n is the number of data points and n^f is the number of features.

The clustering performance in our experiments is measured by the NMI and ARI values between the predicted clustering

[1] http://ana.cachopo.org/datasets-for-single-label-text-categorization

TABLE II
OVERVIEW OF THE 12 TRANSFER CLUSTERING TASKS

Task	Cluster	Source domain	Target domain
C2-1	1	comp.graphics comp.os.ms-windows.misc	comp.sys.ibm.pc.hardware comp.windows.x
	2	sci.crypt sci.electronics	sci.med sci.space
C2-2	1	comp.os.ms-windows.misc comp.sys.mac.hardware	comp.sys.ibm.pc.hardware comp.windows.x
	2	talk.religion.misc talk.politics.mideast	talk.politics.misc talk.politics.guns
C2-3	1	comp.graphics comp.sys.ibm.pc.hardware	comp.os.ms-windows.misc comp.sys.mac.hardware
	2	rec.motorcycles rec.sport.hockey	rec.autos rec.sport.baseball
C2-4	1	rec.motorcycles rec.autos	rec.sport.hockey rec.sport.baseball
	2	sci.crypt sci.electronics	sci.med sci.space
C2-5	1	rec.sport.baseball rec.sport.hockey	rec.autos rec.motorcycles
	2	talk.politics.misc talk.politics.mideast	talk.religion.misc talk.politics.guns
C2-6	1	sci.electronics sci.crypt	sci.med sci.space
	2	talk.politics.misc talk.politics.mideast	talk.religion.misc talk.politics.guns
C3-1	1	comp.sys.mac.hardware	comp.graphics
	2	sci.electronics	sci.crypt
	3	rec.sport.hockey	rec.autos
C3-2	1	comp.sys.mac.hardware	comp.graphics
	2	sci.med	sci.electronics
	3	talk.politics.guns	talk.religion.misc
C3-3	1	rec.motorcycles	rec.autos
	2	sci.med	sci.crypt
	3	talk.politics.misc	talk.religion.misc
C3-4	1	rec.sport.hockey	rec.autos
	2	sci.space	sci.electronics
	3	comp.os.ms-windows.misc	comp.graphics
C4-1	1	comp.os.ms-windows.misc	comp.windows.x
	2	rec.autos	rec.motorcycles
	3	sci.crypt	sci.med
	4	talk.religion.misc	talk.politics.misc
C4-2	1	rec.autos	rec.motorcycles
	2	sci.electronics	sci.med
	3	talk.religion.misc	talk.politics.guns
	4	comp.sys.ibm.pc.hardware	comp.sys.mac.hardware

TABLE III
SUMMARY OF THE 12 TRANSFER CLUSTERING TASKS FROM THE 20NG DATASET

Task	Domain	n	n^f
C2-1	Source Target	2342 2370	7966
C2-2	Source Target	2091 2193	8098
C2-3	Source Target	2377 2345	7620
C2-4	Source Target	2377 2385	9055
C2-5	Source Target	2138 2202	9113
C2-6	Source Target	2215 2109	9262
C3-1	Source Target	1769 1773	6746
C3-2	Source Target	1729 1552	6637
C3-3	Source Target	1657 1566	7395
C3-4	Source Target	1765 1769	6777
C4-1	Source Target	2138 2250	8534
C4-2	Source Target	2152 2315	7920

result and the ground truth clustering result. NMI and ARI are defined in (1) and (4), respectively. We run every algorithm ten times, then the average value and standard deviation of NMI and ARI are calculated as the final performance measures. The number of clustering members of both domains generated in the first step is set to 50. In MOSEP, the size of Ψ is set to $O = 20$ and the maximum number of iterations is set to $\beta = 50$. Besides, we provide the detailed parameter descriptions of the other related algorithms in Table XI in the supplementary material.

In the following experiments, we first compare TCES with other CES algorithms. Then TCE-TCES is compared with a number of transfer clustering algorithms. Next, the effect of the size of clustering member subset is studied. The effect of the base clustering algorithm and the ensemble consensus method in TCE-TCES is studied as well. In addition, we adopt a number of nonparametric tests to compare multiple algorithms. Finally, the robustness of TCES is analyzed.

A. Comparison of Clustering Ensemble Selection Strategies

To evaluate the capability of TCES in selecting clustering member subsets adaptively, we use the clustering member generation strategy in TCE-TCES to produce the clustering members pool for all selecting methods and keep the method of ensemble consensus in TCE-TCES unchanged, we then compare TCES with other CES strategies, including selecting subset with highest quality (HQ) [25], selecting subset with moderate diversity (MD) [27], selecting subset with highest diversity (HD) [25], [26], combining quality and diversity by parameter setting (JS) [25], combining quality and diversity by threshold setting (ACES) [28], and the weighted method (WCES) [23]. Specifically, HQ and HD are applied based on a greedy strategy according to [25], the joint parameter in JS and the threshold in ACES are set to 0.5, and the size of subset is set to 30. Tables IV and V provide the comparison result of TCES with different CES strategies with respect to NMI and ARI. TCES-1 and TCES-2 are TCES with two Pareto frontier ensemble strategies as expressed in (23) and (24), respectively. The best result in each task is highlighted in bold.

It can be seen in Tables IV and V that:

1) good clustering member subsets in different tasks have their own tendencies toward quality or diversity. For example, C2-5 and C3-3 prefer subsets with higher quality (HQ is better than MD and HD); C2-1, C2-4, and C2-6 prefer subsets with moderate diversity (MD is better than HQ and HD); the rest of the tasks prefer subsets with higher diversity (HD is better than HQ and MD). If only quality or diversity is considered, the algorithm may perform well on certain tasks, but not on others;
2) jointly considering quality and diversity (TCES, JS, and ACES) results in a better overall clustering performance than the CES strategies, in which only quality or diversity is considered (HQ, MD, and HD). This indicates that quality and diversity are both important metrics in CES which should be considered comprehensively;

TABLE IV
COMPARISON OF TCES WITH DIFFERENT CES STRATEGIES WITH RESPECT TO NMI

Task	TCES-1	TCES-2	HQ	MD	HD	JS	ACES	WCES
C2-1	**0.8566**±0.0033	0.8563±0.0042	0.6454±0.2199	0.8505±0.0049	0.8467±0.0190	0.8022±0.0260	0.8514±0.0044	0.8481±0.0077
C2-2	0.9049±0.0039	0.9052±0.0044	0.9033±0.0028	0.8541±0.0088	**0.9060**±0.0043	0.9041±0.0036	0.9032±0.0026	0.9031±0.0025
C2-3	0.8958±0.0047	**0.8964**±0.0040	0.3226±0.0005	0.8698±0.0070	0.8793±0.0148	0.6894±0.2399	0.8896±0.0067	0.3690±0.1562
C2-4	0.9046±0.0074	**0.9049**±0.0040	0.3852±0.1543	0.8861±0.0213	0.8848±0.0158	0.8577±0.0044	0.8960±0.0111	0.8927±0.0142
C2-5	0.8298±0.0036	0.8308±0.0034	0.8290±0.0050	0.7812±0.0088	0.8195±0.0087	**0.8328**±0.0055	0.8251±0.0059	0.7813±0.0111
C2-6	**0.7764**±0.0060	0.7741±0.0082	0.3601±0.1524	0.7337±0.0776	0.6678±0.1937	0.4679±0.2109	0.3465±0.1613	0.7705±0.0180
C3-1	0.8314±0.0021	**0.8316** ±0.0020	0.8267±0.0017	0.8226±0.0062	0.8296±0.0014	0.8270±0.0011	0.8298±0.0017	0.8298±0.0014
C3-2	**0.7560**±0.0060	0.7553±0.0048	0.7282±0.0545	0.7172±0.0297	0.7489±0.0062	0.7505±0.0076	0.7518±0.0039	0.7466±0.0037
C3-3	0.8693±0.0051	**0.8698**±0.0040	0.8542±0.0093	0.8515±0.0123	0.8530±0.0066	0.8537±0.0073	0.8588±0.0034	0.8570±0.0041
C3-4	0.6907±0.0037	**0.6922**±0.0031	0.5509±0.0019	0.5549±0.0338	0.6912±0.0051	0.6872±0.0072	0.6823±0.0089	0.6803±0.0065
C4-1	0.8677±0.0055	0.8675±0.0054	0.8063±0.0605	0.8407±0.0342	0.8576±0.0115	0.8119±0.0603	**0.8689**±0.0060	0.8622±0.0058
C4-2	0.8815±0.0063	0.8802±0.0068	0.8266±0.0678	0.8771±0.0073	0.8781±0.0066	0.8747±0.0189	0.8827±0.0032	**0.8839**±0.0031

TABLE V
COMPARISON OF TCES WITH DIFFERENT CES STRATEGIES WITH RESPECT TO ARI

Task	TCES-1	TCES-2	HQ	MD	HD	JS	ACES	WCES
C2-1	**0.9168**±0.0022	0.9163±0.0035	0.6668±0.2856	0.9151±0.0033	0.9078±0.0188	0.8645±0.0261	0.9122±0.0045	0.9088±0.0077
C2-2	0.9520±0.0044	0.9520±0.0028	0.9510±0.0017	0.9121±0.0079	**0.9524**±0.0025	0.9513±0.0022	0.9508±0.0016	0.9508±0.0016
C2-3	0.9454±0.0035	**0.9459**±0.0029	0.2194±0.0007	0.9288±0.0038	0.9321±0.0125	0.6990±0.3137	0.9402±0.0060	0.3690±0.2862
C2-4	0.9471±0.0053	**0.9475**±0.0028	0.3060±0.1981	0.9365±0.0148	0.9320±0.0124	0.9100±0.0039	0.9404±0.0084	0.9389±0.0115
C2-5	0.9018±0.0026	0.9025±0.0024	0.9012±0.0036	0.8635±0.0072	0.8943±0.0064	**0.9039**±0.0039	0.8984±0.0043	0.8429±0.0281
C2-6	**0.8623**±0.0046	0.8606±0.0061	0.2671±0.2249	0.8095±0.1061	0.7070±0.2794	0.4003±0.3017	0.2612±0.2472	0.8563±0.0182
C3-1	0.8822±0.0016	**0.8824**±0.0015	0.8796±0.0011	0.8738±0.0054	0.8805±0.0012	0.8790±0.0005	0.8810±0.0012	0.8810±0.0010
C3-2	**0.7940**±0.0064	0.7935±0.0066	0.7272±0.1167	0.7490±0.0405	0.7817±0.0138	0.7790±0.0049	0.7904±0.0071	0.7887±0.0042
C3-3	0.9150±0.0045	**0.9155**±0.0035	0.9016±0.0083	0.8960±0.0112	0.9002±0.0057	0.9016±0.0065	0.9057±0.0030	0.9042±0.0034
C3-4	0.7598±0.0038	**0.7612**±0.0031	0.4724±0.0017	0.5513±0.0628	0.7592±0.0052	0.7548±0.0076	0.7509±0.0089	0.7487±0.0065
C4-1	0.9095±0.0053	0.9094±0.0056	0.8046±0.1161	0.8717±0.0425	0.8995±0.0118	0.8162±0.1082	**0.9102**±0.0061	0.9041±0.0061
C4-2	0.9143±0.0082	0.9128±0.0087	0.8187±0.1242	0.9046±0.0109	0.9126±0.0068	0.9134±0.0186	0.9187±0.0034	**0.9194**±0.0034

3) WCES performs similarly to ACES except for the unsatisfactory performance in C2-3 and C2-5. The possible reason could be that WCES can consider all the clustering members with different confidence;
4) TCES obtains the best clustering result in 8 out of 12 tasks, and it provides comparable, or even better performance comparing with other quality and diversity-based CES strategies. The possible reasons could be that TCES adaptively finds the suitable tradeoff relationship between quality and diversity from the source domain to guide subset selection in the target domain for different tasks, while other CES strategies only rely on human experience or specific parameter settings to determine the tendencies between quality and diversity, which lacks adaptability;
5) the performances of the two Pareto frontier ensemble strategies are similar, and TCES-1 is preferable to TCES-2 on about half of the tasks and vice versa.

B. Comparison of Transfer Clustering Algorithms

In this experiment, we compare TCE-TCES with four transfer clustering algorithms, including STC [37], TSC [38], TFCM [42], and TFSC [42]. These transfer clustering algorithms are the improvements of single clustering algorithm by transfer learning, while TCE-TCES is the ensemble version. Tables VI and VII provide the comparison results of TCE-TCES with other transfer clustering algorithms with respect to NMI and ARI.

We can observe the following.

1) TCE-TCES-1 and TCE-TCES-2 obtain the best clustering results in 10 out of 12 tasks. TCE-TCES improves the performance by about 1%–5% comparing with other transfer clustering algorithms. We believe that the ensemble method combines multiple clustering members and adds relevant knowledge of the source domain to facilitate the selection of clustering member subset, which can significantly improve the clustering performance.
2) TFCM and TFSC have unsatisfactory performance, and we consider that they may not be effective on high-dimensional datasets.

C. Effect of the Clustering Member Subset Size

From the generation of subsets in MOSEP, we can notice that there is no need to specify the size of the clustering member subset. With the alternation of generation and elimination, subsets with suitable size will remain. To study the effect of the size of clustering member subset, we compare TCES with the CES strategies (HQ, MD, HD, JS, and ACES) as described in Section V-A, and set the subset size to 10, 20, 30, 40, and 50. The result is shown in Fig. 3. The horizontal axis corresponds to the subset size, while the vertical axis is the NMI value. The horizontal solid line stands for TCES. We can see that the performance of other CES strategies fluctuates with the subset size, and it is not easy to determine the maximum point, while TCES maintains a high level performance on the average. Generally speaking, TCES can adaptively choose the size of subsets, and achieves a comparable or even better performance than the other approaches.

D. Effect of the Base Clustering Algorithm

In the first step of TCE-TCES, we generate clustering members by KM as well as SC. In order to study the effect of

TABLE VI
COMPARISON OF TCE-TCES WITH DIFFERENT TRANSFER CLUSTERING ALGORITHMS WITH RESPECT TO NMI

Task	TCES-1	TCES-2	STC	TSC	TFCM	TFSC
C2-1	**0.8566**±0.0033	0.8563±0.0042	0.5000±0.1593	0.8358±0.0042	0.2222±0.0000	0.2835±0.0022
C2-2	0.9049±0.0039	**0.9052**±0.0044	0.8366±0.0006	0.8989±0.0034	0.6618±0.0000	0.5433±0.0227
C2-3	0.8958±0.0047	**0.8964**±0.0040	0.7718±0.0042	0.8799±0.0049	0.6754±0.0000	0.5190±0.0194
C2-4	0.9046±0.0074	**0.9049**±0.0040	0.2213±0.1071	0.8929±0.0036	0.0498±0.0000	0.0647±0.0172
C2-5	0.8298±0.0036	**0.8308**±0.0034	0.3630±0.0113	0.8073±0.0042	0.1495±0.0000	0.1744±0.0114
C2-6	**0.7764**±0.0060	0.7741±0.0082	0.3261±0.0190	0.7202±0.0100	0.1544±0.0000	0.0160±0.0039
C3-1	0.8314±0.0021	0.8316±0.0020	0.2404±0.0157	**0.8336**±0.0049	0.0808±0.0000	0.0676±0.0050
C3-2	0.7560±0.0060	0.7553±0.0048	0.3136±0.0125	**0.7696**±0.0028	0.1851±0.0000	0.3139±0.0110
C3-3	0.8693±0.0051	**0.8698**±0.0040	0.4253±0.0279	0.8667±0.0092	0.2452±0.0000	0.2057±0.0204
C3-4	0.6907±0.0037	**0.6922**±0.0031	0.2578±0.0075	0.5556±0.0045	0.1533±0.0000	0.1540±0.0063
C4-1	**0.8677**±0.0055	0.8675±0.0054	0.4997±0.0633	0.8380±0.0291	0.3495±0.0000	0.2693±0.0036
C4-2	**0.8815**±0.0063	0.8802±0.0068	0.4823±0.0425	0.8785±0.0093	0.2902±0.0000	0.1924±0.0060

TABLE VII
COMPARISON OF TCE-TCES WITH DIFFERENT TRANSFER CLUSTERING ALGORITHMS WITH RESPECT TO ARI

Task	TCES-1	TCES-2	STC	TSC	TFCM	TFSC
C2-1	**0.9168**±0.0022	0.9163±0.0035	0.5958±0.1665	0.9055±0.0030	0.2761±0.0000	0.2683±0.0024
C2-2	**0.9520**±0.0044	**0.9520**±0.0028	0.8971±0.0005	0.9453±0.0024	0.7323±0.0000	0.6369±0.0243
C2-3	0.9454±0.0035	**0.9459**±0.0029	0.8562±0.0031	0.9355±0.0032	0.7738±0.0000	0.5857±0.0199
C2-4	0.9471±0.0053	**0.9475**±0.0028	0.2675±0.1146	0.9439±0.0022	0.0118±0.0000	0.0573±0.0169
C2-5	0.9018±0.0026	**0.9025**±0.0024	0.4612±0.0129	0.8824±0.0034	0.1864±0.0000	0.1927±0.0154
C2-6	**0.8623**±0.0046	0.8606±0.0061	0.3955±0.0203	0.8149±0.0090	0.1836±0.0000	0.0051±0.0006
C3-1	0.8822±0.0016	0.8824±0.0015	0.2502±0.0174	**0.8856**±0.0033	0.0162±0.0000	0.0074±0.0012
C3-2	0.7940±0.0064	0.7935±0.0066	0.2716±0.0075	**0.8138**±0.0018	0.0980±0.0000	0.2772±0.0072
C3-3	0.9150±0.0045	**0.9155**±0.0035	0.4372±0.0486	0.9135±0.0071	0.2688±0.0000	0.2150±0.0179
C3-4	0.7598±0.0038	**0.7612**±0.0031	0.2762±0.0082	0.6010±0.0036	0.1206±0.0000	0.0928±0.0044
C4-1	**0.9095**±0.0053	0.0053±0.0056	0.4865±0.0674	0.8715±0.0399	0.2581±0.0000	0.0794±0.0020
C4-2	0.9143±0.0082	0.9128±0.0087	0.4570±0.0493	**0.9162**±0.0083	0.1480±0.0000	0.1148±0.0052

these base clustering algorithms, we keep TCE-TCES with the first Pareto frontier ensemble strategy unchanged except for the clustering member generation method: 1) using only KM to generate clustering members (TCE-KM); 2) using only SC to generate clustering members only (TCE-SC); and 3) using both KM and SC to generate clustering members (TCE-KMSC). The result in terms of NMI is shown in Fig. 11 in the supplementary material. It can be seen that TCE-KM has better results than TCE-SC in some tasks (C2-1, C2-4, and C4-2), while TCE-SC outperforms TCE-KM in the remaining tasks. However, by taking KM and SC into consideration, TCE-KMSC obtains the best performance in most of the tasks. The possible reason could be that KM and SC can view the dataset from different perspectives, based on which diverse and complementary clustering members are produced.

E. Effect of Ensemble Consensus Method

TCE-TCES integrates clustering member subsets through a hypergraph. To explore the effect of ensemble consensus methods, we compare our method, which uses Ncut as the consensus function (TCE-Ncut), with a number of alternative consensus functions: 1) using KM (TCE-KM); 2) using hierarchical clustering (TCE-HC); and 3) using the METIS algorithm (TCE-METIS) [18], [49]. The result is shown in Fig. 5 in the supplementary material. TCE-Ncut outperforms the other approaches on 10 out of 12 tasks. TCE-KM, TCE-HC, and TCE-METIS have their own advantages on certain tasks but they perform unsatisfactorily on other tasks. For example, TCE-HC is comparable to TCE-Ncut on tasks C2-4, C2-5, C3-1, and C4-2, but its performance is not as good on tasks C2-1, C2-3, C2-6, C3-4, and C4-1. Generally speaking, TCE-Ncut is able to integrate the subsets in a more stable and accurate way by balancing between the degree of compactness within clusters with the degree of scatter between clusters.

F. Nonparametric Tests

We further adopt a number of nonparametric tests [65], [66] to compare multiple algorithms, including TCES-1, TCES-2, HQ, MD, HD, JS, ACES, WCES, STC, TSC, TFCM, and TFSC over the 12 clustering tasks. Tables VIII–X in the supplementary material show the significant difference among various algorithms mentioned above by using different nonparametric statistical procedures, such as the Bonferroni–Dunn test, the Holm test, the Hochberg test, and the Hommel test. It can be seen in Table VIII in the supplementary material that the average ranking of TCES-1 and TCES-2 are 2.25 and 1.99, respectively, which are higher than other approaches, and indicates that TCES obtains an overall better performance on these clustering tasks. Tables IX and X in the supplementary material list the p-values in control tests and multiple tests, which indicate that TCES-2 has very close performance with TCES-1 and TCES outperforms other algorithms to different degrees of significance.

G. Robustness Analysis

We further analyze the robustness of TCE-TCES from the two aspects of tasks and noise. Fig. 6 in the supplementary material shows the performance bias of the top 6 algorithms in Table VIII in the supplementary material (TCES-1, TCES-2, HS, ACES, WCES, and TSC) over all clustering tasks.

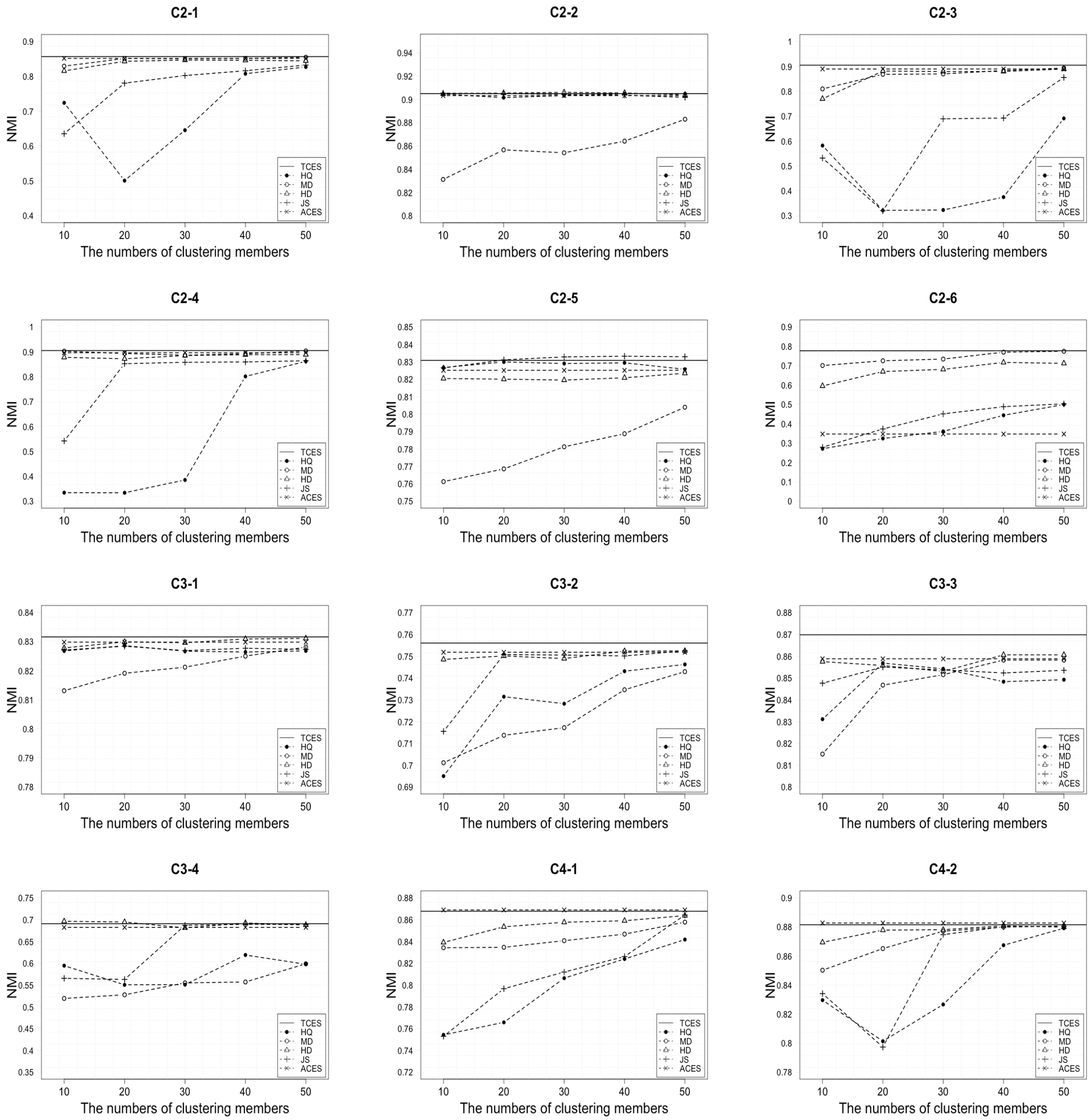


Fig. 3. Effect of the clustering member subset size with respect to NMI.

The vertical axis indicates the NMI value bias from the best results, and 0 means the best result among these six algorithms. It can be seen that TCES-1 and TCES-2 have overall lower bias and more robust performance for all tasks, while other algorithms fluctuate significantly on some tasks. The possible reason could be that the competing methods cannot adaptively adjust themselves to fit all the tasks, therefore they perform poorly on some tasks. In addition, we evaluate the performance of TCES with noisy clustering members and noisy data. First, we generate different numbers of noisy clustering members ranging from 5% to 10%, each of which partitions the target dataset randomly, and these members are used to replace the same number of normal clustering members in the pool. Fig. 7 in the supplementary material shows the performance of TCES with different numbers of noisy clustering members. We can observe that TCES has high robustness to these types of noise. Next, different levels of Gaussian noise (σ ranges from 0.1 to 0.5) are added to selected data points (1%–3%) in the six tasks to test the robustness of TCES-1. Fig. 9 in the supplementary material shows the results, where the horizontal axis corresponds to the standard deviation of the Gaussian noise and the vertical axis corresponds to the NMI values. It can be seen that an increase of noise level affects the clustering performance to a certain extent due to the inclusion of fewer high quality clustering members.

H. Comparison of Cluster Selection Algorithms

Instead of selecting clustering members, some CES algorithms select useful clusters among all these members. In this section, we compare the performance of TCES with three state-of-the-art cluster selection methods, including EEAC [71], CLWEAC [70], and AAPMM [33] (the detailed parameter settings are shown in Table XI in the supplementary material). We use the same strategy as before to generate the clustering members, then run these algorithms 30 times and take the average NMI values as the final results. We test their performance over all the 12 tasks which are shown in Table XII in the supplementary material. TCES outperforms EEAC, CLWEAC, and AAPMM on most of the tasks. The possible reason could be that these three cluster selection methods choose 33% or 50% of the clusters in an empirical way that may lead to poor performance on some tasks, while TCES can adaptively select a suitable clustering member subset to balance between quality and diversity. In addition, the computation time of the selection process in TCES is also studied. The running time and performance of TCES-1 is closely related to the number of iterations of MOSEP, so we study the performance trend with respect to the number of iterations over tasks C2-1, C3-1, and C4-1, as shown in Fig. 10 in the supplementary material. We can observe that the performance becomes relatively stable when the number of iterations is around 10. Table XIII in the supplementary material lists the average execution time of the selection processes in TCES-1, EEAC, CLWEAC, and AAPMM on task C2-1. The required time of TCES-1 with 5, 10, and 20 iterations of MOSEP is given, which is longer than the other three methods. The possible reason for the increased time could be due to the additional step of transferring knowledge from the source domain to adaptively balance between quality and diversity, and TCES improves clustering performance at the cost of additional execution time.

VI. Conclusion

In this paper, we focus on the problem of CES. Specifically, we propose a TCES algorithm, which combines the idea of transfer learning and CES to adaptively select clustering members based on a tradeoff between quality and diversity, the MOSEP which selects the clustering members while optimizing the proposed three objectives, and the transfer CE framework (TCE-TCES). We perform a thorough study of TCE-TCES on 12 clustering tasks constructed from the 20NG dataset, and obtain a number of conclusions. First, the tradeoff relationship between quality and diversity, as well as the size of the clustering member subset, affect the performance. Second, TCES can adaptively determine the tradeoff relationship and the subset size, which leads to better performance on most of the tasks when compared with other CES strategies. Third, TCE-TCES combines transfer learning with CE, and outperforms other traditional transfer clustering algorithms. Fourth, the selection of a suitable basic clustering algorithm and consensus method can improve the performance as well. Fifth, the performance of TCES based on nonparametric tests is studied, and its robustness is analyzed. Finally, we compare TCES with a number of state-of-the-art cluster selection methods. In the future, we shall further deploy TCE-TCES in a distributed environment to enhance its efficiency, and test it with different types of datasets. Besides, we will explore other combination strategies between transfer learning and CE.

References

[1] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[2] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[3] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Doc. Anal. Recognit.*, vol. 1, 1995, pp. 278–282. doi: 10.1109/ICDAR.1995.598994.

[4] Z. Yu, H. Chen, J. You, G. Han, and L. Li, "Hybrid fuzzy cluster ensemble framework for tumor clustering from biomolecular data," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 10, no. 3, pp. 657–670, May 2013.

[5] B. Hanczar and M. Nadif, "Ensemble methods for biclustering tasks," *Pattern Recognit.*, vol. 45, no. 11, pp. 3938–3949, 2012.

[6] Z. Yu, H. S. Wong, J. You, Q. Yang, and H. Liao, "Knowledge based cluster ensemble for cancer discovery from biomolecular data," *IEEE Trans. Nanobiosci.*, vol. 10, no. 2, pp. 76–85, Jun. 2011.

[7] G. Rafiee, S. S. Dlay, and W. L. Woo, "Region-of-interest extraction in low depth of field images using ensemble clustering and difference of Gaussian approaches," *Pattern Recognit.*, vol. 46, no.10, pp. 2685–2699, 2013.

[8] S. Zhang, H.-S. Wong, and Y. Shen, "Generalized adjusted rand indices for cluster ensembles," *Pattern Recognit.*, vol. 45, no. 6, pp. 2214–2226, 2012.

[9] M. Yousefnezhad, S. J. Huang, and D. Zhang, "WoCE: A framework for clustering ensemble by exploiting the wisdom of crowds theory," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 486–499, Feb. 2018.

[10] D. Huang, C.-D. Wang, and J.-H. Lai, "Locally weighted ensemble clustering," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1460–1473, May 2018, doi: 10.1109/TCYB.2017.2702343.

[11] Z. Yu *et al.*, "Semi-supervised ensemble clustering based on selected constraint projection," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2394–2407, Dec. 2018.

[12] Y. Jiang *et al.*, "Collaborative fuzzy clustering from multiple weighted views," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 688–701, Apr. 2015.

[13] Z. Yu, J. You, H.-S. Wong, and G. Han, "From cluster ensemble to structure ensemble," *Inf. Sci.*, vol. 198, pp. 81–99, Sep. 2012.

[14] M. C. Naldi, A. C. P. L. F. Carvalho, and R. J. G. B. Campello, "Cluster ensemble selection based on relative validity indexes," *Data Min. Knowl. Disc.*, vol. 27, no. 2, pp. 259–289, 2013.

[15] L. Franek and X. Jiang, "Ensemble clustering by means of clustering embedding in vector spaces," *Pattern Recognit.*, vol. 47, no.2, pp. 833–842, 2014.

[16] Y. Hong, S. Kwong, Y. Chang, and Q. Ren, "Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm," *Pattern Recognit.*, vol. 41, no. 9, pp. 2742–2756, 2008.

[17] J. Kleinberg, "An impossibility theorem for clustering," in *Proc. Neural Inf. Process. Syst.*, 2002, pp. 463–470.

[18] A. L. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Jan. 2003.

[19] B. Minaei-Bidgoli, A. Topchy, and W. F. Punch, "A comparison of resampling methods for clustering ensembles," in *Proc. Int. Conf. Mach. Learn. Models Technol. Appl.*, 2004, pp. 939–945.

[20] B. Minaei-Bidgoli, A. Topchy, and W. F. Punch, "Ensembles of partitions via data resampling," in *Proc. Int. Conf. Inf. Technol. Coding Comput.*, 2004, pp. 188–192.

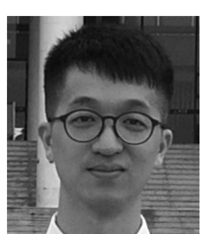
[21] A. Topchy, B. Minaeibidgoli, A. K. Jain, and W. F. Punch, "Adaptive clustering ensembles," in *Proc. Int. Conf. Pattern Recognit.*, vol. 1, 2004, pp. 272–275.

[22] Z. Yu, H. S. Wong, and H. Wang, "Graph-based consensus clustering for class discovery from gene expression data," *Bioinformatics*, vol. 23, no. 21, pp. 2888–2896, 2007.

[23] Z.-H. Zhou and W. Tang, "Clusterer ensemble," *Knowl. Based Syst.*, vol. 19, no. 1, pp. 77–83, 2006.

[24] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.

[25] X. Z. Fern and W. Lin, "Cluster ensemble selection," *Stat. Anal. Data Min.*, vol. 1, no. 3, pp. 128–141, 2008.
[26] L. I. Kuncheva and S. T. Hadjitodorov, "Using diversity in cluster ensembles," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 2, 2004, pp. 1214–1219.
[27] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova, "Moderate diversity for better cluster ensembles," *Inf. Fusion*, vol. 7, no. 3, pp. 264–275, 2006.
[28] J. Azimi and X. Z. Fern, "Adaptive cluster ensemble selection," in *Proc. Int. Joint Conf. Artif. Intell.*, 2009, pp. 992–997.
[29] Y. Hong, S. Kwong, H. Wang, and Q. Ren, "Resampling-based selective clustering ensembles," *Pattern Recognit. Lett.*, vol. 30, no. 3, pp. 298–305, 2009.
[30] J. Jia, X. Xiao, B. Liu, and L. Jiao, "Bagging-based spectral clustering ensemble selection," *Pattern Recognit. Lett.*, vol. 32, no. 10, pp. 1456–1467, 2011.
[31] X. Wang, D. Han, and C. Han, "Rough set based cluster ensemble selection," in *Proc. 16th Int. Conf. Inf. Fusion*, 2013, pp. 438–444.
[32] X. Lu, Y. Yang, and H. Wang, "Selective clustering ensemble based on covariance," in *Proc. Int. Workshop Multiple Classifier Syst.*, 2013, pp. 179–189.
[33] H. Alizadeh, B. Minaeibidgoli, and H. Parvin, "Cluster ensemble selection based on a new cluster stability measure," *Intell. Data Anal.*, vol. 18, no. 3, pp. 389–408, 2014.
[34] Z. Yu *et al.*, "Hybrid clustering solution selection strategy," *Pattern Recognit.*, vol. 47, no. 10, pp. 3362–3375, 2014.
[35] E. Akbari, H. M. Dahlan, R. Ibrahim, and H. Alizadeh, "Hierarchical cluster ensemble selection," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 146–156, Mar. 2015.
[36] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[37] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Self-taught clustering," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 200–207.
[38] W. Jiang and F.-L. Chung, "Transfer spectral clustering," in *Proc. Eur. Conf. Princ. Data Min. Knowl. Disc.*, 2012, pp. 789–803.
[39] L. Yu, Y. Dang, and G. Yang, "Transfer clustering via constraints generated from topics," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2012, pp. 3203–3208.
[40] S. Samanta, A. T. Selvan, and S. Das, "Cross-domain clustering performed by transfer of knowledge across domains," in *Proc. Comput. Vis. Pattern Recognit. Image Process. Graph.*, 2013, pp. 1–4.
[41] S. Sun, Y. Jiang, and P. Qian, "Transfer learning based maximum entropy clustering," in *Proc. IEEE Int. Conf. Inf. Sci. Technol.*, 2014, pp. 829–832.
[42] Z. Deng *et al.*, "Transfer prototype-based fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 5, pp. 1210–1232, Oct. 2016.
[43] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Neural Inf. Process. Syst.*, 2001, pp. 849–856.
[44] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[45] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2000, pp. 849–858.
[46] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, 1985.
[47] R. Ghaemi, M. N. Sulaiman, H. D. Ibrahim, and N. Mustapha, "A survey: Clustering ensembles techniques," in *Proc. World Acad. Sci. Eng. Technol.*, vol. 50, 2009, p. 636.
[48] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 25, no. 3, pp. 337–372, 2011.
[49] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, 1998.
[50] A. Cardoso-Cachopo, "Improving methods for single-label text categorization," Ph.D. dissertation, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, Lisbon, Portugal, 2007.
[51] P. Qian *et al.*, "Cluster prototypes and fuzzy memberships jointly leveraged cross-domain maximum entropy clustering," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 181–193, Jan. 2016.
[52] Z. Yu, L. Li, J. Liu, and G. Han, "Hybrid adaptive classifier ensemble," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 177–190, Feb. 2015.
[53] Z. Yu *et al.*, "Hybrid *K*-nearest neighbor classifier," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1263–1275, Jun. 2016.
[54] Z. Yu *et al.*, "Progressive semisupervised learning of multiple classifiers," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 689–702, Feb. 2018.
[55] V. Soto, S. García-Moratilla, G. Martínez-Muñoz, D. Hernández-Lobato, A. Suárez, "A double pruning scheme for boosting ensembles," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2682–2695, Dec. 2014.
[56] S. Yu and J. Shi, "Multiclass spectral clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, 2003, pp. 11–17.
[57] U. V. Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[58] H. F. Liu, T. L. Liu, J. J. Wu, D. C. Tao, and Y. Fu, "Spectral ensemble clustering," in *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2015, pp. 715–724.
[59] F. Yang, T. Li, Q. Zhou, and H. Xiao, "Cluster ensemble selection with constraints," *Neurocomputing*, vol. 235, pp. 59–70, Apr. 2017.
[60] X. Zhao, J. Jiang, and C. Dang, "Clustering ensemble selection for categorical data based on internal validity indices," *Pattern Recognit.*, vol. 69, pp. 150–168, Sep. 2017.
[61] Z. W. Yu *et al.*, "Distribution-based cluster structure selection," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3554–3567, Nov. 2017.
[62] M. Yousefnezhad, A. Reihanian, D. Zhang, and B. Minaei-Bidgoli, "A new selection strategy for selective cluster ensemble based on diversity and independency," *Eng. Appl. Artif. Intell.*, vol. 56, pp. 260–272, Nov. 2016.
[63] T. Li and C. Ding, "Weighted consensus clustering," in *Proc. SIAM Int. Conf. Data Min.*, 2008, pp. 798–809.
[64] T. Li, C. Ding, and M. I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," in *Proc. Int. Conf. Data Min.*, 2007, pp. 577–582.
[65] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.
[66] Z. W. Yu *et al.*, "A new kind of nonparametric test for statistical comparison of multiple classifiers over multiple datasets," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4418–4431, Dec. 2017.
[67] H. Alizadeh, M. Yousefnezhad, and B. M. Bidgoli, "Wisdom of crowds cluster ensemble," *Intell. Data Anal.*, vol. 19, no. 3, pp. 485–503, 2016.
[68] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm," *Pattern Anal. Appl.*, vol. 18, no. 1, pp. 87–112, 2015.
[69] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on elite selection of weighted clusters," *Adv. Data Anal. Classification*, vol. 7, no. 2, pp. 181–208, 2013.
[70] A. Nazari, A. Dehghan, S. Nejatian, V. Rezaie, and H. Parvin, "A comprehensive study of clustering ensemble weighting based on cluster quality and diversity," in *Pattern Analysis and Applications*, vol. 3. London, U.K.: Springer, 2017, pp. 1–13.
[71] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin, "To improve the quality of cluster ensembles by selecting a subset of base clusters," *J. Exp. Theor. Artif. Intell.*, vol. 26, no. 1, pp. 127–150, 2014.
[72] B. Minaei-Bidgoli, H. Parvin, H. Alinejad-Rokny, H. Alizadeh, and W. F. Punch, "Effects of resampling method and adaptation on clustering ensemble efficacy," *Artif. Intell. Rev.*, vol. 41, no. 1, pp. 27–48, 2014.
[73] Z. Yu *et al.*, "Multiobjective semisupervised classifier ensemble," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2018.2824299.



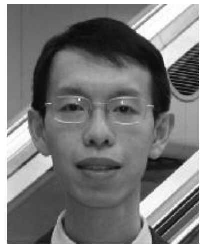
Yifan Shi is currently pursuing the master's degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His current research interests include machine learning and data mining.

Zhiwen Yu (S'06–M'08–SM'14) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2008.

He is a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, and an Adjunct Professor with Sun Yat-sen University, Guangzhou. He has published over 100 referred journal papers and international conference papers. His current research interests include data mining, machine learning, and pattern recognition.

Dr. Yu is a Council Member of China Computer Federation (CCF), a Distinguished Member of CCF, and a Senior Member of ACM and CAAI.



Hau-San Wong received the B.Sc. and M.Phil. degrees in electronic engineering from the Chinese University of Hong Kong, Hong Kong, and the Ph.D. degree in electrical and information engineering from the University of Sydney, Sydney, NSW, Australia.

He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has also held research positions with the University of Sydney and the Hong Kong Baptist University, Hong Kong. His current research interests include multimedia information processing, multimodal human–computer interaction, and machine learning.

C. L. Philip Chen (S'88–M'88–SM'94–F'07) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1985 and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is currently a Chair Professor with the Department of Computer and Information Science and the Dean of the Faculty of Science and Technology, University of Macau, Macau, China. His current research interests include computational intelligence, systems, and cybernetics.

Yide Wang received the B.S. degree in electrical engineering from the Beijing University of Post and Telecommunication, Beijing, China, in 1984 and the M.S. and Ph.D. degrees in signal processing and telecommunications from the University of Rennes, Rennes, France, in 1986 and 1989, respectively.

He is currently a full-time Professor with the Ecole Polytechnique de l'Universite de Nantes, Nantes, France. He is a member of the Communication Systems Team, IETR (UMR CNRS 6164) Laboratory, Rennes. His current research interests include array signal processing, spectral analysis, and mobile wireless communication systems.

Jane You received the Ph.D. degree from La Trobe University, Melbourne, VIC, Australia, in 1992.

She joined Hong Kong Polytechnic University, Hong Kong, in 1998, where she is currently a Professor with the Department of Computing and the Chair of Department Research Committee. She has over 190 research papers published with over 1000 nonself citations. She has been a Principal Investigator for one ITF project, three GRF projects, and many other joint grants. She is also a Team Member for two successful patents (one HK patent and one U.S. patent). Her current research interests include image processing, medical imaging, computer-aided diagnosis, and pattern recognition.

Ms. You was a recipient of three Awards, including the Hong Kong Government Industrial Awards, the Special Prize and Gold Medal with Jury's Commendation at the 39th International Exhibition of Inventions of Geneva in 2011 for her research on retinal imaging, and the Second Place in an International Competition (SPIE Medical Imaging'2009 Retinopathy Online Challenge) in 2009. She is also an Associate Editor of *Pattern Recognition* and other journals.

Jun Zhang (M'02–SM'08–F'16) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2002.

Since 2016, he has been with the South China University of Technology, Guangzhou, China, where he is currently a Cheung Kong Professor. His current research interests include computational intelligence, cloud computing, big data, and wireless sensor networks. He has authored seven research books and book chapters, and over 100 technical papers in the above areas.

Dr. Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON CYBERNETICS. He is the Founding and Current Chair of the IEEE Guangzhou Section and ACM Guangzhou Chapter.