

fMBN-E: Efficient Unsupervised Network Structure Ensemble and Selection for Clustering

Xiao-Lei Zhang

Abstract—It is known that unsupervised nonlinear dimensionality reduction and clustering is sensitive to the selection of hyperparameters, particularly for deep learning based methods, which hinders its practical use. How to select a proper network structure that may be dramatically different in different applications is a hard issue for deep models, given little prior knowledge of data. In this paper, we aim to automatically determine the optimal network structure of a deep model, named multilayer bootstrap networks (MBN), via extremely simple ensemble learning and selection techniques. Specifically, we first propose an MBN ensemble (MBN-E) algorithm which concatenates the sparse outputs of a set of MBN base models with different network structures into a new representation. Then, we take the new representation produced by MBN-E as a reference for selecting the optimal MBN base models. Moreover, we propose a fast version of MBN-E (fMBN-E), which is not only theoretically even faster than a single standard MBN but also does not increase the estimation error of MBN-E. Importantly, MBN-E and its ensemble selection techniques maintain the simple formulation of MBN that is based on one-nearest-neighbor learning. Empirically, comparing to a number of advanced deep clustering methods and as many as 20 representative unsupervised ensemble learning and selection methods, the proposed methods reach the state-of-the-art performance without manual hyperparameter tuning. fMBN-E is empirically even hundreds of times faster than MBN-E without suffering performance degradation. The source code is available at <http://www.xiaolei-zhang.net/mbn-e.htm>.

Index Terms—Ensemble selection, cluster ensemble, multilayer bootstrap networks, unsupervised learning

I. INTRODUCTION

UNSUPERVISED learning and clustering is a fundamental task of machine learning. It finds wide applications in data mining, community detection, human-machine interaction, biology, etc. One of its long term headache problem is hyperparameter tuning. Since the early works on principal component analysis (PCA) and k-means clustering, a vast number of methods have been developed. Some algorithms conduct clustering in the original data space directly without parameter tuning, such as agglomerative clustering. However, their performance is usually unsatisfied, since the data in the original space is usually linearly-inseparable and noisy. Later on, research turned to projecting data in the original space into a probability space where

the data is supposed to be uniformly distributed and linearly separable, such as manifold learning, kernel methods, and probabilistic models. However, a proper probability space is usually found by tuning parameters manually, e.g. kernel widths or regularization parameters. Although some work has tried to find the optimal parameters automatically, the learned representation, which is produced from a single layer nonlinear transform, is not abstract enough to describe the semantic classes of data.

To learn highly abstract representations, deep neural network based data clustering has received much attention recently. The first work [1] extracts abstract representations from the bottleneck layer of a deep belief network. However, the output of the deep belief network aims to reconstruct its input data without considering the clustering task. To make the deep representations suitable for clustering, some work adds additional terms, such as constraints [2], clustering-like loss functions and models [3], [4], or novel network structures [5], to the network training. Some work learns deep representations and refines cluster assignments iteratively [6]–[9]. Recently, a new kind of deep learning based clustering, named self-supervised clustering optimizes cleverly designed objective functions of some pretext tasks, such as image completion, image colorization, or clustering, in which supervised pseudo labels are automatically obtained from the input data without manual annotations. It can be generally categorized into predictive self-supervised clustering [10]–[12], generative self-supervised clustering [13]–[17], and contrastive self-supervised clustering [18]–[22], respectively [23], [24]. Although the methods achieve superior performance over conventional clustering methods, many of them apply handcrafted priors to the benchmark data case by case, such as strong prior knowledge of data, (e.g. multi-modal information [25]), data augmentation with clear intrinsic data structures (e.g. [26]), or hyperparameter tuning with the ground-truth labels (e.g. [27]).

As we know, a long term goal of unsupervised learning and clustering is to design algorithms that are tuning-free and with little human labor, like k-means clustering. However, the aforementioned methods need to be tuned more or less. If the hyperparameters were not properly set, then the performance may drop significantly. Although auto machine learning tries to find the optimal hyperparameters without human labors, it is mainly designed for supervised learning. As for unsupervised deep learning and clustering, the topic seems far from explored yet.

This paper aims to find the optimal hyperparameter

Xiao-Lei Zhang is with the Research & Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen, China, and the School of Marine Science and Technology and the Center for Intelligent Acoustics and Immersive Communications, Northwestern Polytechnical University, Xi'an, China. E-mail: xiaolei.zhang@nwpu.edu.cn.

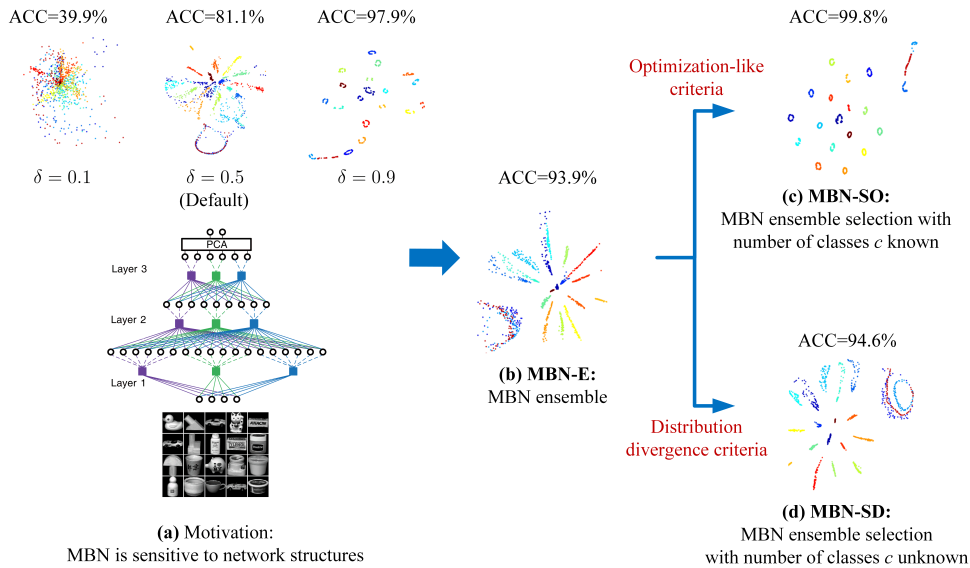


Fig. 1. On the network structure selection problem of MBN. Each square of MBN in figure (a) represents a base clustering, while the black circles connected to the square represent the input/output of the base clustering. The hyperparameter “ δ ” controls the network structure of MBN. The words in red color are two ensemble selection criteria for MBN-SO and MBN-SD respectively. The word “ACC” is short for clustering accuracy. The demo data is the COIL20 dataset [43].

setting of unsupervised deep models automatically. Because the search space of auto machine learning is large for a common deep neural network, we pick *multilayer bootstrap network* (MBN) [28] as a research object. MBN is an ideal object, since that it is a deep model sensitive to only a single hyperparameter which is used to control the network structure of MBN.

We address the network structure selection problem of MBN by unsupervised ensemble learning and ensemble selection. See Section I-A for an overview of the state-of-the-art works on unsupervised ensemble learning and selection. Although many ensemble selection methods may be applied successfully, we aim to exploit a simple and efficient way under the rule of Occam’s Razor. We find empirically that, when applied to MBN, even a very simple ensemble selection method is able to achieve comparable top performance with the advanced ones, where 20 unsupervised ensemble learning and selection methods are used for comparison in the Supplement Material. Eventually, this finding derives a simple and efficient tuning-free unsupervised deep learning algorithm for practical use.

To summarize, as shown in Fig. 1, the contribution of this paper is listed as follows:

- MBN ensemble (MBN-E) is proposed. It groups the sparse outputs of a number of MBN base models with different hyperparameter settings into a new representation.
- A fast version of MBN-E (fMBN-E) is proposed. It first discards the random feature selection step of MBN, and then changes the step of random data resampling into the random resampling of similarity scores. It accelerates MBN-E by over hundreds of times both theoretically and empirically. We have proved that the acceleration does not sacrifice accu-

racy.

- MBN ensemble selection with optimization-like criteria (MBN-SO) is proposed. It first predicts the labels of data by conducting clustering on the output representation of MBN-E, and then measures the discriminant ability of the output representation of each base model by the optimization-like criteria given the predicted labels. Finally, it selects the base models with highly discriminant outputs as a new ensemble.
- MBN ensemble selection with distribution divergence criteria (MBN-SD) is proposed. It measures the distribution divergence between the outputs of MBN-E and its base models by maximum mean discrepancy (MMD), and then selects the base models whose outputs are similar to the MBN-E output. To our knowledge, this is the first time that unsupervised ensemble selection is conducted on data distributions directly without clustering labels.

Note that, because the optimization-like criteria require predicted labels to evaluate the discriminant ability of a data distribution, we consider using MBN-SO for the scenario where the number of classes is known as a prior. Because the distribution divergence criteria evaluate divergence of data distributions directly, we use MBN-SD mainly for the scenario where the number of classes is unknown.

We have run experiments on a number of benchmark datasets where the optimal hyperparameter of MBN appears at fundamentally different ranges. Experimental results show that MBN-E significantly outperforms MBN with the default setting and approaches to MBN with the optimal setting. fMBN-E achieves similar performance with MBN-E, and is over dozens of times faster than MBN-E. MBN-SO and MBN-SD further improves the performance of MBN-E.

The rest of the paper is organized as follows. In Section I-A, we present the related work that contributes to the novelty of the proposed methods, including cluster ensemble, ensemble selection, and unsupervised domain adaptation. In Section II, we introduce MBN as a preliminary. In Sections III to IV, we present MBN-E, fMBN-E, MBN-SO, and MBN-SD, respectively. In Section V, we present an extensive experiments. Finally, in Section VI, we conclude the paper.

A. Related work

1) *Clustering ensemble*: Ensemble learning, such as *bagging*, *boosting*, and their variations, has demonstrated its effectiveness on many learning problems. Unsupervised ensemble learning inherits the fundamental theories and methods of classifier ensemble. The mostly studied unsupervised ensemble learning is *clustering ensemble*. It aims to combine multiple *base clusterings* with a so-called *meta-clustering function*, a.k.a *consensus function*, for enhancing the stability and accuracy of the base clusterings [29], [30]. Meta-clustering functions can be categorized generally to two classes [30]. The first class analyzes the co-occurrence of objects: how many times an object belongs to one cluster or how many times two objects belong to the same cluster. The second class, called the median partition, pursues the maximal similarity with all partitions in the ensemble [31]. Recently, some unsupervised deep ensemble learning methods have been proposed. For example, [32] takes deep neural networks act like a meta-clustering function. [33] decomposes each layer of a deep neural network into an ensemble of encoders or decoders and mask operations. To our knowledge, unsupervised deep ensemble learning is not prevalent, due to maybe that neural networks need supervised signals to maximize their discriminant ability. See [30], [34], [35] for the reviews of clustering ensemble.

2) *Clustering ensemble reweighting and selection*: Because not all base clusterings contribute equivalently to a cluster ensemble, it is needed to conduct ensemble reweighting and selection, which mainly focuses on three respects: (i) different types of weights, (ii) algorithms for determining the weights, and (iii) cluster validation criteria for measuring the diversity and quality of the base models.

The most common type of weights is to assign a weight to each base clustering according to its quality or/and diversity in the ensemble, e.g. [36]. A special case of this type is to constrain the weights of some weak base clusterings to zero, named *clustering selection* [37], [38]. However, weak base clusterings may also contain some high quality clusters, and vice versa. With this perspective, many reweighting strategies at levels of clusters [39], [40], data structures [41], and data points [42] were proposed.

The algorithms for determining the weights can be categorized into two types [44]. The first type calculates weights by measuring the similarity between the predicted labels of the clustering ensemble and its base clusterings [36], [37]. The second type treats the weights as variables of consensus functions which are obtained by advanced optimization algorithms, e.g. [45].

The criteria for measuring the diversity and quality of the base models can be categorized into two classes. The first class of measurements calculates the normalized mutual information [36], [37], adjusted rand index [46], clustering accuracies [47], and their variants [48] between the sets of the predicted labels. The second class of validation criteria is based on data distributions [49], [50]. They usually calculate some kinds of statistics of data [41], [51]. Some systematical studies on cluster validation indices [49], [50] have been carried out as well.

3) *Unsupervised domain adaptation*: Domain adaptation is the ability of applying an algorithm trained in one or more “source domains” to a different but related “target domain”. Unsupervised domain adaptation is a subtask of domain adaptation where the target domain does not have labels. The algorithms can be categorized into three branches [52], which are sample-based, feature-based, and inference-based approaches. No matter how the approaches vary, the distribution divergence measurement between the source domains and the target domain always lies in the core of unsupervised domain adaptation. The most popular measurement is MMD [53]. Other measurements include Kullback-Leibler divergence, total variation distance, second-order (covariance) statistics, and Hellinger distance. Although the distribution divergence measurement has been extensively studied in unsupervised domain adaptation, it seems far from explored in unsupervised ensemble selection.

II. PRELIMINARY

A. An introduction to MBN

This paper takes MBN as a research object. It is a simple deep model. As shown in Fig. 1a, suppose we are to build an M -layer MBN from bottom-up, it can be described as follows:

- Step 1, for each layer, MBN trains V mutually-independent k -centroids base clustering. For each base clustering, it takes the following three operators successively to generate a new representation of data:
 - **Random selection of features**: It first randomly selects some features of the input data, which yields a new representation of the data.
 - **Random sampling of data**: It randomly samples k data points from the data with the new representation as the k centroids.
 - **One nearest neighbor optimization**: It assigns each input data to one of the k clusters, and outputs a k -dimensional one-hot code, indicating which cluster the input data belongs to.

The one-hot representations from all base clusterings are concatenated as the input of the upper layer.

- Step 2, MBN stacks the cluster ensemble described in Step 1 for M times. The parameter k at two adjacent layers have the following connection:

$$k_m = \delta k_{m-1} \quad (1)$$

where k_m and k_{m-1} are the parameter k at the m -th and $(m-1)$ -th adjacent layers respectively, and $\delta \in (0, 1)$ is a hyperparameter controlling the network structure of MBN.

The principle for the success of MBN is as follows [28]:

Theorem 1. *MBN builds as many as $O(k_o 2^V)$ agglomerative hierarchical trees on the original data space, where k_o is the parameter k at the top layer. The child nodes of a tree are gradually merged into father nodes from bottom up, which equals to the process of discarding small local variances and noise gradually. The root nodes of the trees construct the abstract representation of data.*

To understand Theorem 1, we first imagine that a single k -centroids base clustering partitions the input space to k disconnected fractions. Thereafter, V base clusterings partition the input space to $O(k 2^V)$ fractions at the maximum. Given parameters $k_1 > k_2 > \dots > k_o$, it is easy to see that $O(k_1 2^V) > O(k_2 2^V) > \dots > O(k_o 2^V)$. As a result, between any two adjacent layers, there must be $O(k_{m-1} 2^V) - O(k_m 2^V)$ nodes at the $(m-1)$ -th layer absorbed into other nodes, which builds tree structures. The effectiveness of the trees is guaranteed by that each base clustering is a weak learner that discards noise and small variances of the input as if k is large enough. See [28] for the detailed explanation.

B. On the network structure selection problem of MBN

Like many deep models, MBN is sensitive to its network structure. Specifically, given the parameters at the bottom layer k_1 and at the top layer k_o fixed, how fast the agglomerative trees grow up from k_1 to k_o , which is determined by δ in (1), should match the nonlinearity and noise level of data. For example, as shown in Fig. 1a, when δ approaches to 0, MBN builds a shallow network with a single nonlinear layer, which is suitable for linearly separable data and hence performs poorly on the COIL20 data. When δ is enlarged towards 1, MBN becomes deeper and deeper, which is suitable for nonlinear and non-Gaussian data. Some contrary examples to COIL20 can also be observed in [28, Fig. 10]. Because it is difficult to evaluate the properties of data in unsupervised learning, MBN has to make a compromise by setting $\delta = 0.5$. This may lead to far inferior performance from the optimal one.

III. MULTILAYER BOOTSTRAP NETWORK ENSEMBLE

In this section, we first introduce MBN-E in Section III-A, then present an efficient algorithm for MBN-E, named fMBN-E, in Section III-B and finally discuss why fMBN-E can accelerate MBN-E without sacrificing estimation accuracy in Section III-C.

A. MBN-E

MBN-E is an ensemble of MBN base models who have different δ . We present MBN-E in detail as follows:

- **Step 1: Train an ensemble of MBN base learners.**

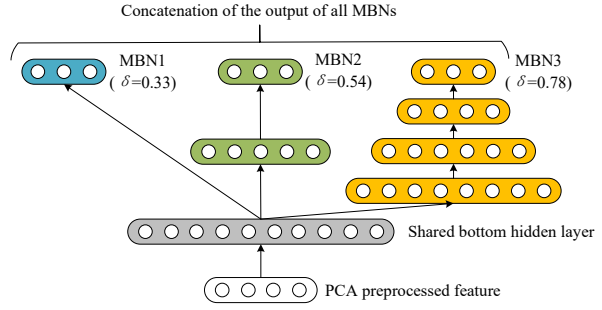


Fig. 2. Architecture of fMBN-E. Different color represents different MBN base models with random δ values.

MBN-E trains Z MBN base models ($Z \gg 1$). For each MBN base model, we randomly sample its hyperparameter δ from the range $(0, 1)$. Then, we train the MBN model layer by layer from bottom up, with the parameter $k_m = \delta k_{m-1}$. The entire training process stops when k_m reaches a predefined value k_o ($k_o \ll k_1$).

- **Step 2: Construct an output layer.**

After training an ensemble of MBNs, MBN-E concatenates the sparse outputs of the MBN base models as a new representation of the data. If we denote the output of the z -th MBN base model as $\{\mathbf{x}_{z,i}\}_{i=1}^n$, $\forall z = 1, \dots, Z$, then the output representation of MBN-E is $\bar{\mathbf{x}}_i = [\mathbf{x}_{1,i}^T, \dots, \mathbf{x}_{Z,i}^T]^T$, $\forall i = 1, \dots, n$, where n is the number of data points.

Because $\{\bar{\mathbf{x}}_i\}_{i=1}^n$ is very high dimensional, we sometimes need to reduce $\{\bar{\mathbf{x}}_i\}_{i=1}^n$ to a low-dimensional representation $\{\bar{\mathbf{y}}_i\}_{i=1}^n$ in an Euclidian space by, e.g. PCA, for applications. Likewise, we denote the low-dimensional representation of $\{\mathbf{x}_{z,i}\}_{i=1}^n$ as $\{\mathbf{y}_{z,i}\}_{i=1}^n$. We usually conduct PCA preprocessing before MBN-E, which not only reduces the computational complexity of the bottom layers of the MBN base models but also de-correlates the input features.

From [28], we can derive the following theorem:

Theorem 2. *The computational complexity of MBN-E approximates to $Z(\mathcal{O}(dskVn) + \mathcal{O}(kVn))$ empirically, where $\mathcal{O}(dskVn)$ and $\mathcal{O}(kVn)$ are the complexity of a single MBN model at the bottom layer and the other layers respectively, d is the dimension of the original input data, and s is the sparsity of the data.*

The complexity is too high when $Z \gg 1$ and $k \propto n$, which is mainly caused by the calculation of the distance between the input data and the k centroids.

B. fMBN-E

To reduce the computational complexity of MBN-E, fMBN-E has a new architecture shown in Fig. 2. It is described as follows:

- **Step 1: Share the bottom layer.**

fMBN-E trains a single bottom layer as Section II-A does.

- **Step 2: Train an ensemble of MBN base learners by random resampling of similarity scores.**

fMBN-E builds Z MBN base models that use the output from Step 1 as their inputs. For each layer of a MBN base model, suppose its input data is $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_n]$ where \mathbf{u}_i is the i -th input data point of the layer. fMBN-E first calculates the similarity matrix of the input data by $\mathbf{S} = \mathbf{U}^T \mathbf{U}$. Then, for each k -centroids clustering, fMBN-E randomly selects k columns of \mathbf{S} into a new matrix \mathbf{S}' , which is the similarity scores between the input data and the randomly sampled centroids of the k -centroids clustering. Finally, the one-hot representation of the input data \mathbf{u}_i is obtained by activating the position that corresponds to the largest value of the i -th row of \mathbf{S}' .

• **Step 3: Construct an output layer.**

This step is the same as MBN-E.

From the above algorithm, we can easily obtain that:

Theorem 3. *The computational complexity of fMBN-E is $\mathcal{O}(dskVn) + \mathcal{O}(Zn^2)$.*

Comparing Theorems 2 and 3, we see that the computational complexities of the bottom layer and the other layers are reduced by Z and kV/n times respectively. For example, in a typical setting where $k = n/2$, $Z = 40$, and $V = 400$, the computational complexity of MBN-E is as high as $(\mathcal{O}(8000dsn^2) + \mathcal{O}(8000n^2))$, while the complexity of fMBN-E is $\mathcal{O}(200dsn^2) + \mathcal{O}(40n^2)$ which may be hundreds of times faster than MBN-E. Particularly, because the complexity of the original MBN model is $(\mathcal{O}(dskVn) + \mathcal{O}(kVn))$ [28], we can see that fMBN-E may be even faster than the original MBN since that V is larger than Z in practice.

C. Analysis

Here we study theoretically how fMBN-E reduces the computational complexity of MBN-E without suffering significant performance degradation. First of all, we review the following theorem:

Theorem 4. *The estimation error of a single layer of MBN $\mathcal{E}_{\text{ensemble}}$ and the estimation error of a single k -centroids clustering $\mathcal{E}_{\text{single}}$ in the layer have the following relationship:*

$$\mathcal{E}_{\text{ensemble}} = \left(\frac{1}{V} + \left(1 - \frac{1}{V} \right) \rho \right) \mathcal{E}_{\text{single}} \quad (2)$$

where ρ is the pairwise positive correlation coefficient between the k -centroids clusterings, $0 \leq \rho \leq 1$ [28].

According to the theorem, we can draw the connections between $\mathcal{E}_{\text{ensemble}}/\mathcal{E}_{\text{single}}$, ρ , and V in Fig. 3, and further derive the following corollary.

Corollary 1. *The estimation errors of the bottom layers of fMBN-E $\mathcal{E}_{\text{fMBN-E}}$ and MBN $\mathcal{E}_{\text{MBN-E}}$ have the following connection:*

$$\frac{\mathcal{E}_{\text{fMBN-E}}}{\mathcal{E}_{\text{MBN-E}}} = \frac{\left(\frac{1}{V} + \left(1 - \frac{1}{V} \right) \rho \right) \mathcal{E}_{\text{single}}}{\left(\frac{1}{ZV} + \left(1 - \frac{1}{ZV} \right) \rho \right) \mathcal{E}_{\text{single}}} = \frac{Z + (ZV - Z)\rho}{1 + (ZV - 1)\rho} \quad (3)$$

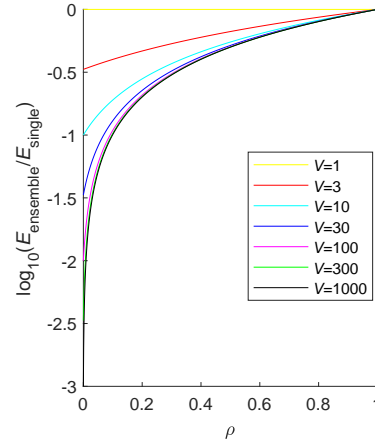


Fig. 3. Relationship between the estimation error $\mathcal{E}_{\text{ensemble}}/\mathcal{E}_{\text{single}}$, correlation coefficient ρ , and number of k -centroids clusterings per layer V .

Remark 1. *When V is large enough, the estimation error of the bottom layer of fMBN-E is similar to that of Z independent bottom layers of MBN-E:*

$$\mathcal{E}_{\text{fMBN-E}} \approx \mathcal{E}_{\text{MBN-E}} \quad (4)$$

Proof: According to Corollary 1, we see that, when V and N are both large enough, $\mathcal{E}_{\text{fMBN-E}}/\mathcal{E}_{\text{MBN-E}}$ is determined by ρ . For the first case when $\rho \rightarrow 0$, $\mathcal{E}_{\text{fMBN-E}} \approx Z\mathcal{E}_{\text{MBN-E}}$; for the second case when $\rho \gg 0$, $\mathcal{E}_{\text{fMBN-E}} \approx \mathcal{E}_{\text{MBN-E}}$. In the following, we show that the second case is true.

From Theorem 4, we see that the estimation error $\mathcal{E}_{\text{ensemble}}$ is proportional to two contradict factors $\mathcal{E}_{\text{single}}$ and ρ . On one side, when enlarging k towards n , $\mathcal{E}_{\text{single}}$ becomes small, so as to $\mathcal{E}_{\text{ensemble}}$. On the other side, any two k -centroids clusterings may share a number of common centroids in probability of $(k/n)^2$. It is easy to imagine that $\rho \propto (k/n)^2$. Therefore, when enlarging k , ρ is enlarged, so as to $\mathcal{E}_{\text{ensemble}}$.

For the bottom layer of a single MBN, empirically, setting k to a large number balances $\mathcal{E}_{\text{single}}$ and ρ , which produces the minimum $\mathcal{E}_{\text{ensemble}}$. Here we take the common setting $k = n/2$ as an example. In this setting, we may have $\rho \approx 0.25$ for instance, which supports that $\mathcal{E}_{\text{fMBN-E}} \approx \mathcal{E}_{\text{MBN-E}}$. Remark 1 is proved. ■

Remark 1 motivates us to merge the bottom layers of MBN-E to a single bottom layer.

Remark 2. *The random feature selection step reduces the estimation error of the bottom layer of fMBN-E significantly.*

Proof: From Theorem 4 and Fig. 3, we see clearly that, when ρ is fixed and V is large enough, the estimation error of the bottom layer of fMBN-E, i.e. $\mathcal{E}_{\text{fMBN-E}}$, is upper-bounded by $\rho\mathcal{E}_{\text{single}}$.

Following the proof of Remark 1, we see that ρ at the bottom layer of fMBN-E is far larger than 0, e.g. $\rho \approx 0.25$. From Fig. 3, we see that $\mathcal{E}_{\text{ensemble}}$ does not reduce the estimation error much over $\mathcal{E}_{\text{single}}$ when $\rho \approx 0.25$.

Therefore, we need to further reduce ρ by decorrelating the k -centroids clusterings via the random feature selection step, which should be able to reduce $\mathcal{E}_{\text{fMBN-E}}$ significantly. ■

Remark 2 motivates us to retain the random feature selection step at the bottom layer of fMBN-E.

Remark 3. *The random feature selection step has limited effect on the upper layers of the MBN base models of fMBN-E.*

Proof: For the upper layers of fMBN-E, following the proof of Remark 1, because the nonlinearity and noise has been gradually reduced by the lower layers, setting k to a small number, e.g. $k = n/2^3$, is able to achieve a good $\mathcal{E}_{\text{single}}$. In this setting, we may have $\rho \approx 1/2^6$ for instance. From Fig. 3, we see that $\mathcal{E}_{\text{ensemble}}$ is far smaller than $\mathcal{E}_{\text{single}}$ when $\rho \approx 1/2^6$. Therefore, we do not need the random feature selection step to further pursue a marginal reduction of $\mathcal{E}_{\text{ensemble}}$. ■

Remark 3 motivates us to remove the random feature selection step at the upper layers of fMBN-E, which provides the opportunity to reduce the computational complexity significantly.

IV. UNSUPERVISED NETWORK STRUCTURE SELECTION

In this section, we first present an unsupervised ensemble selection framework for MBN-E in Section IV-A, and then present MBN-SO and MBN-SD in Sections IV-B and IV-C respectively.

A. Framework

Algorithm 1 presents the unsupervised ensemble selection framework for MBN-E. If the number of classes c is given, it first conducts clustering on $\{\bar{\mathbf{y}}_i\}_{i=1}^n$, which generates a set of predicted labels $\{l_i\}_{i=1}^n$. Then, it calculates a weight w_z for the z -th MBN base model by an optimization-like criterion $f_{\text{MBN-SO}}(\{l_i\}_{i=1}^n, \{\mathbf{y}_{z,i}\}_{i=1}^n)$. If c is not given, it calculates the weight w_z by evaluating the difference of the distributions $\{\bar{\mathbf{x}}_i\}_{i=1}^n$ and $\{\mathbf{x}_{z,i}\}_{i=1}^n$ directly via a distribution divergence criterion $f_{\text{MBN-SD}}(\cdot)$. After obtaining $\{w_z\}_{z=1}^Z$, it concatenates the sparse output of the B ($B \ll Z$) MBN base models whose weights are the B largest ones among $\{w_z\}_{z=1}^Z$ into a new sparse representation of data $\{\bar{\bar{\mathbf{x}}}_i\}_{i=1}^n$.

Note that there are a vast number of ensemble selection algorithms manipulating on $\{w_z\}_{z=1}^Z$. Because this is not the focus of this paper, here we prefer the simple yet effective one.

B. MBN-SO: Ensemble selection with optimization-like criteria

When the number of classes c is given, we use the optimization-like criteria to generate the weights of the base models. We follow the comparison conclusion on the optimization-like criteria [49], and pick the 4 best criteria,

Algorithm 1 Unsupervised ensemble selection for MBN-E.

Input: Sparse output of MBN-E $\{\bar{\mathbf{x}}_i\}_{i=1}^n$ and its low-dimensional representation $\{\bar{\mathbf{y}}_i\}_{i=1}^n$;
 Sparse outputs of the MBN base models $\{\{\mathbf{x}_{z,i}\}_{i=1}^n\}_{z=1}^Z$ and their low-dimensional representations $\{\{\mathbf{y}_{z,i}\}_{i=1}^n\}_{z=1}^Z$;
 Number of selected base models B
 Number of classes c (optional).
Output: $\{\bar{\bar{\mathbf{x}}}_i\}_{i=1}^n, \{\bar{\bar{\mathbf{y}}}_i\}_{i=1}^n$.

```

1: if  $c$  is given then
2:    $\{l_i\}_{i=1}^n \leftarrow \text{clustering}(\{\bar{\mathbf{y}}_i\}_{i=1}^n, c)$ 
3:   for  $z = 1$  to  $Z$  do
4:      $w_z \leftarrow f_{\text{MBN-SO}}(\{l_i\}_{i=1}^n, \{\mathbf{y}_{z,i}\}_{i=1}^n)$ 
      (or  $w_z \leftarrow f_{\text{MBN-SO}}(\{l_i\}_{i=1}^n, \{\mathbf{x}_{z,i}\}_{i=1}^n)$ )
5:   end for
6: else
7:   for  $z = 1$  to  $Z$  do
8:      $w_z \leftarrow f_{\text{MBN-SD}}(\{\bar{\mathbf{x}}_i\}_{i=1}^n, \{\mathbf{x}_{z,i}\}_{i=1}^n)$ 
      (or  $w_z \leftarrow f_{\text{MBN-SD}}(\{\bar{\mathbf{y}}_i\}_{i=1}^n, \{\mathbf{y}_{z,i}\}_{i=1}^n)$ )
9:   end for
10: end if
11: Pick  $B$  sparse representations that correspond to the  $B$ 
    largest weights, supposed to be  $\{\{\mathbf{x}_{b,i}\}_{i=1}^n\}_{b=1}^B$  without
    loss of generality
12:  $\bar{\bar{\mathbf{x}}}_i \leftarrow [\mathbf{x}_{1,i}^T, \dots, \mathbf{x}_{B,i}^T]^T$ 
13:  $\bar{\bar{\mathbf{y}}}_i \leftarrow \text{PCA}(\bar{\bar{\mathbf{x}}}_i)$ 

```

which are the silhouette width criterion (SWC), point-biserial (PB), PBM, and variance ratio criterion (VRC), respectively. Because they are defined in Euclidian spaces, we take the low-dimensional representations $\{\mathbf{y}_{z,i}\}_{z=1}^Z$ of the MBN base models for evaluation. We omit the subscript z for simplicity in this subsection. The criteria are described as follows:

1) *Silhouette width criterion:* SWC calculates the ratio of the geometric compactness and separation of clusters. Suppose the i -th data point \mathbf{y}_i belongs to a cluster $p \in \{1, \dots, c\}$. Let the average distance of \mathbf{y}_i to all other data points in cluster p be denoted by a_i . Let the average distance of \mathbf{y}_i to all data points in another cluster q ($q \neq p$) be denoted as $g_{q,i}$. Let b_i be the minimum $g_{q,i}$ over all $q = 1, \dots, c, q \neq p$. Then, the silhouette of \mathbf{y}_i is defined as:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \quad (5)$$

In case that cluster p consists of only \mathbf{y}_i , then $s_i = 0$.

The SWC score is the average of s_i over all data points:

$$w^{\text{SWC}} = \frac{1}{n} \sum_{i=1}^n s_i \quad (6)$$

The higher the SWC score is, the better the discriminant ability of a representation is.

2) *Point-biserial:* PB calculates correlation between a distance matrix and a binary matrix that encodes the pairwise memberships of data points to clusters. It first calculates the average within-class distance d_w and the

average between-class distance d_b , which can be formulated as:

$$d_w = \frac{1}{n} \sum_{i=1}^n a_i \quad (7)$$

$$d_b = \frac{1}{n} \sum_{i=1}^n \sum_{\{q|q=1,\dots,c,q \neq p\}} \frac{n_q}{n - n_p} g_{q,i} \quad (8)$$

where n_p is the number of data points of cluster p where \mathbf{y}_i belongs to, and n_q is the number of data points in cluster q where $q = 1, \dots, c$ and $q \neq p$. Then, it is defined as:

$$w^{\text{PB}} = \frac{(d_b - d_w) \sqrt{w_d b_d / t^2}}{s_d} \quad (9)$$

where s_d is the standard deviation of the pairwise distances of all data points, $w_d = \sum_{p=1}^c n_p(n_p - 1)/2$ is the number of within-class distances, $b_d = \sum_{p=1}^c n_p(n - n_p)/2$ is the number of between-class distances, and $t = n(n - 1)/2$ is the total number of pairwise distances. The higher the PB score is, the better the discriminant ability of a representation is.

3) *PBM*: PBM is defined over between-class distances and within-class distances:

$$w^{\text{PBM}} = \left(\frac{1}{k} \frac{E_1}{E_K} D_K \right)^2 \quad (10)$$

where E_1 denotes the average distance between the data points and the grand mean of the data, E_K denotes the average within-class distances, and D_K denotes the maximum distance between cluster centroids:

$$E_1 = \frac{1}{n} \sum_{i=1}^n \|\mathbf{y}_i - \bar{\boldsymbol{\mu}}\| \quad (11)$$

$$E_K = \frac{1}{n} \sum_{p=1}^c \sum_{\{\mathbf{y}_i | l_i = p\}} \|\mathbf{y}_i - \boldsymbol{\mu}_p\| \quad (12)$$

$$D_K = \max_{p,q=1,\dots,c} \|\boldsymbol{\mu}_p - \boldsymbol{\mu}_q\| \quad (13)$$

where $\bar{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i$ is the grand mean of the data, $\boldsymbol{\mu}_p = \frac{1}{n_p} \sum_{\{\mathbf{y}_i | l_i = p\}} \mathbf{y}_i$ is the center of the p -th cluster centroid. A large PBM score implies a good separation ability of the representation.

4) *Variance ratio criterion*: VRC calculates the ratio of the between-class variance over within-class variance:

$$w^{\text{VRC}} = \frac{1}{h} \frac{n - c}{c - 1} \frac{\text{tr}(\mathbf{B})}{\text{tr}(\mathbf{W})} \quad (14)$$

where $\text{tr}(\cdot)$ denotes the trace operator, h is the dimension of the feature, and \mathbf{B} and \mathbf{W} are the between-class variance and within-class variance respectively, defined as:

$$\mathbf{W} = \sum_{p=1}^c \mathbf{W}_p \quad (15)$$

$$\mathbf{W}_p = \sum_{\{\mathbf{y}_i | l_i = p\}} (\mathbf{y}_i - \boldsymbol{\mu}_p)(\mathbf{y}_i - \boldsymbol{\mu}_p)^T \quad (16)$$

$$\mathbf{B} = \sum_{p=1}^c n_p (\boldsymbol{\mu}_p - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_p - \bar{\boldsymbol{\mu}})^T \quad (17)$$

The normalization terms $1/h$ and $(n - c)/(c - 1)$ make the VRC score irrelevant to h and c . A large VRC score implies a good separation ability of the representation.

C. MBN-SD: Ensemble selection with distribution divergence criteria

When the number of classes c is unknown, we prefer MMD, which is a common distribution divergence criterion in unsupervised domain adaptation, for evaluating the distribution divergence between the outputs of MBN-E and its MBN base models.

We have also studied many probability distribution divergence criteria in literature, including the Kullback-Leibler (KL) divergence, total variance distance, L2-norm distance, Hellinger distance, Wasserstein distance, Bhattacharyya distance, etc. Unfortunately, they do not work for MBN-SD.

1) *Maximum mean discrepancy*: MMD is originally defined in kernel-induced feature spaces, where multiple kernels are usually adopted to reach an accurate estimation. Here we simply use the linear kernel based MMD to evaluate the distribution divergence between $\{\bar{\mathbf{x}}_i\}_{i=1}^n$ and $\{\mathbf{x}_{z,i}\}_{i=1}^n$. Since $\bar{\mathbf{x}}_i = [\mathbf{x}_{1,i}^T, \dots, \mathbf{x}_{Z,i}^T]^T$, here we define MMD as follows:

$$\begin{aligned} v^{\text{MMD}} &= \frac{1}{Z} \frac{1}{n(n-1)} \sum_{i \neq j} \bar{\mathbf{x}}_i^T \bar{\mathbf{x}}_j \\ &+ \frac{1}{n(n-1)} \sum_{i \neq j} \mathbf{x}_{z,i}^T \mathbf{x}_{z,j} - \frac{2}{Z} \frac{1}{n^2} \sum_{u=1}^Z \sum_{i,j} \mathbf{x}_{u,i}^T \mathbf{x}_{z,j} \end{aligned} \quad (18)$$

Because the first term of MMD is the same for all MBN base models, we only calculate the last two terms in practice. The smaller the MMD score is, the more similar the distributions $\{\bar{\mathbf{x}}_i\}_{i=1}^n$ and $\{\mathbf{x}_{z,i}\}_{i=1}^n$ are. To make MMD satisfy Algorithm 1, we transform v^{MMD} by:

$$w^{\text{MMD}} = 1 - \frac{v^{\text{MMD}} - v_{\min}}{v_{\max} - v_{\min}} \quad (19)$$

where v_{\max} and v_{\min} are the largest and smallest values of all MMD scores respectively.

V. EXPERIMENTS

In this section, we first compare the proposed methods with a number of representative methods on several benchmark datasets in Section V-D, then demonstrate how fMBN-E accelerates MBN-E without sacrificing accuracy in Section V-E, and finally study the effect of the number of the selected base models in Section V-F, as well as how the generation method of the referenced labels affect MBN-SO in Section V-G.

A. Datasets

We selected 8 benchmark datasets as summarized in Table I. For Extended-Yale B, because the luminance of the images dominates the similarity measurement instead of the faces themselves, we preprocessed Extended-Yale B by

TABLE I
DESCRIPTION OF DATA SETS. THE TERM “OPTIMAL δ ” DENOTES
WHERE THE OPTIMAL PERFORMANCE OF MBN APPEARS BY
SEARCHING δ FROM A RANGE OF $(0, 1)$.

Name	# samples	# dimensions	# classes	Attribute	Optimal δ
Dermatology	366	34	6	Biomedical	(0, 0.2)
New-Thyroid	255	5	3	Biomedical	(0, 0.35)
UMIST	575	1024	20	Faces	(0.75, 0.85)
Extended-Yale B	2414	32256	38	Faces	(0.6, 0.75)
COIL20	1440	4096	20	Images	(0.8, 0.9)
COIL100	7200	1024	100	Images	(0.8, 0.9)
20-Newsgroups	18846	26214	20	Text	(0.4, 0.5)
MNIST	70000	768	10	Images	(0.35, 0.75)

the dense scale invariant feature transform as in [54]. For 20-Newsgroups, we extracted the term frequency-inverse document frequency text feature. PCA preprocessing was applied to the image datasets, which reduced the original features to 100 dimensions. Cosine similarity measurement was used to measure the similarity between the documents of 20-Newsgroups. All other datasets used Euclidean distance as the similarity measurement. Clustering accuracy (ACC) was used as the evaluation metric.

From the table, we see that the operating range of the optimal δ of MBN appears at dramatically different positions, which are sufficient to demonstrate how the proposed methods address the network structure selection problem, as well as how the proposed methods behave when comparing with the state-of-the-art referenced methods.

B. Parameter settings

The parameter settings of MBN and the proposed methods are summarized as follows:

- **MBN (default) [28]:** We used its default setting as in [28].
- **MBN-E:** It used 40 MBN base models. The base models of MBN-E used the same parameter setting as MBN except that δ was randomly selected from $[0.05, 0.95]$.
- **fMBN-E:** It is the fast version of MBN-E without performance degradation. It discards the random feature selection step in the upper layers of the MBN base models.
- **fMBN-Ev2:** It is a *variant of fMBN-E* that discards the random feature selection step at the bottom layer, and uses the random resampling of similarity scores instead of the random data resampling to train the bottom layer as its upper layers. It accelerates the training time of the bottom layer of fMBN-E, with a risk of performance degradation.
- **MBN-SO:** The number of selected base models B was set to 3. The MBO-SO with the four optimization-like criteria are denoted as “MBN-SO (SWC)”, “MBN-SO (PB)”, “MBN-SO (PBM)”, and “MBN-SO (VRC)”, respectively.
- **MBN-SD:** The parameter B was set to 10.

Agglomerative hierarchical clustering (AHC) was used for partitioning data into clusters. Although the MMD criterion

in MBN-SD is designed to handle the case where the number of classes is unknown, we still give AHC the number of classes during the clustering stage, for a comparable study on how the distribution divergence criterion differs from the optimization-like criteria in MBN-SO. All reported results are average ones over 5 independent runs. The time efficiency was evaluated on an Intel(R) Xeon(R) Platinum 8160 CPU server with 512 GB memory, where the CPU has 48 physical cores. All experiments were run with 48 parallel workers of MATLAB.

C. Comparison methods

The comparison strategy is described as follows. For the image datasets, we copied the ranking lists of the image clustering methods from <https://paperswithcode.com/>, which reflects the state-of-the-art performance on the datasets. Note that because self-supervised deep learning based methods actually explore strong handcrafted features from augmented data, we omit them from the experiments to maintain the fairness of the comparison. For the small-scale Dermatology and New-Thyroid datasets that deep learning methods usually do not handle with, we compared with 12 representative clustering ensemble methods, see Supplementary Material for the referenced methods. All these clustering ensemble methods are meta-clustering functions, which can be used jointly with any base clusterings, such as k-means or spectral clustering. Here we took 40 k-means clusterings as the base clusterings for each meta-clustering function. Like many clustering ensemble methods, e.g. [71], we selected the number of clusters of each k-means base clustering randomly from a range of $[2c, 10c]$. For the 20-Newsgroups text corpus, we compared with 9 text clustering methods, see [72] for the referenced methods. Besides, k-means clustering are also provided as a baseline. Because k-means clustering suffers from bad local minima, we ran k-means clustering on each dataset for 100 times, and pick one that has the minimum objective value. All reported results are average ones over 5 independent runs.

D. General results

Table II lists the results of the aforementioned comparison methods and the proposed methods. Because it is too lengthy to list all results, here we only list the results of the top 5 referenced methods; for the proposed MBN-SO variants, we only provide “MBN-SO (VRC)” as a representative. See Supplementary Material for the results of the other three variants of MBN-SO. We also list the performance of the MBN with the optimal δ , denoted as MBN^\dagger . Note that because it is unlikely to select the optimal δ manually in real-world applications, MBN^\dagger only provides an upperbound of the proposed methods.

From the table, we see that the proposed methods outperform “MBN (default)” in general, as what we have targeted to in this paper. Specifically, MBN-E outperforms “MBN (default)” on UMIST, Extended Yale B, COIL20, and COIL100 significantly where the optimal operating

TABLE II

ACC COMPARISON BETWEEN THE PROPOSED METHODS AND THE STATE-OF-THE-ART REFERENCED METHODS. THE RESULTS OF THE REFERENCED METHODS ON THE DATASETS MARKED WITH “*” ARE COPIED FROM THEIR ORIGINAL PUBLICATIONS OR THE “PAPERS WITH CODE” WEBSITE. THE NUMBER IN BOLD DENOTES THE BEST PERFORMANCE.

	Dermatology	New-Thyroid	UMIST*	Extended-Yale B*
kmeans	0.261	0.860	0.408	0.311
Rank1	0.313 (DREC [55])	0.863 (Borda [56])	0.769 (DASC [57])	0.992 (DMSC [25])
Rank2	0.307 (LinkClueE [58])	0.859 (LinkClueE [58])	0.750 (DSC-Net-L2 [5])	0.973 (DSC-Net-L2 [5])
Rank3	0.306 (HGPA [29])	0.853 (ECPCS_MC [59])	0.732 (J-DSSC [27])	0.924 (J-DSSC [27])
Rank4	0.299 (CSPA [29])	0.851 (MCLA [29])	0.728 (DSC-Net-L1 [5])	0.917 (A-DSSC [27])
Rank5	0.297 (ECPCS_HC [59])	0.845 (Vote [60])	0.725 (A-DSSC [27])	0.776 (SSC-OMP [61])
MBN (default)	0.855	0.881	0.544	0.934
MBN-E	0.866	0.860	0.670	0.973
MBN-SO (VRC)	0.714	0.771	0.767	0.941
MBN-SD	0.947	0.941	0.547	0.909
MBN [†]	0.971	0.964	0.770	0.969

	COIL20*	COIL100*	20-Newsgroups	MNIST*
kmeans	0.679	0.511	0.416	0.527
Rank1	1.000 (JULE [9])	0.911 (JULE [9])	0.600 (LTM [62])	0.979 (N2D [63])
Rank2	0.858 (AGDL [64])	0.824 (A-DSSC [27])	0.523 (DFPA [65])	0.969 (DDC-DA [26])
Rank3	0.858 (GDL [64])	0.796 (J-DSSC [27])	0.490 (LDA [66])	0.965 (PSSC [67])
Rank4	0.793 (DBC [8])	0.775 (DBC [8])	0.447 (AnchorFree [68])	0.964 (GDL [64])
Rank5	N/A	0.731 (GDL [64])	0.435 (LapPLSI [69])	0.939 (SR-K-means [70])
MBN (default)	0.795	0.683	0.623	0.964
MBN-E	0.929	0.832	0.584	0.964
MBN-SO (VRC)	0.995	0.908	0.623	0.964
MBN-SD	0.973	0.803	0.611	0.963
MBN [†]	0.994	0.901	0.623	0.965

range of δ of MBN is far from the default value 0.5. It is also comparable to “MBN (default)” on Dermatology and New-Thyroid. As for MNIST and 20-Newsgroups, even if the default δ happens to be in the optimal operating range, MBN-E can still be competitive to “MBN (default)” if the optimal range is wide enough, such as that on MNIST. MBN-SO further improves the performance of MBN-E, and outperforms “MBN (default)” significantly on most datasets, except the small-scale Dermatology and New-Thyroid. Finally, MBN-SD outperforms “MBN (default)” on Dermatology and New-Thyroid, COIL20, and COIL100 significantly, and is comparable to the latter in the remaining four datasets.

The proposed MBN-SO also approaches to the top performance of the referenced methods on most datasets. Although it behaves worse than DMSC on Extended Yale B, it still ranks among the top 5 comparison methods. Here we need to emphasize one merit of MBN-SO: it is implemented in a simple mathematical form and behaves robustly across datasets without carefully selected architectures or hyperparameters, which fascinates its practical use. Note that it is interesting to observe that the clustering ensemble methods do not show significant performance improvement over k-means on the small scale Dermatology and New-Thyroid data. Note also that although deep learning has dominated image clustering, it is not very prevalent in text clustering. From the table as well as the summary on text clustering in <https://paperswithcode.com/>, we see that

the deep model DFPA [65] is inferior to the conventional probabilistic method LTM [62].

Focusing on our three algorithms, we see that MBN-SO is at least comparable to MBN-E and MBN-SD on most of the challenging data, except the two small-scale data where a shallow network of MBN is able to produce a highly accurate result. Comparing MBN-E and MBN-SD, we see that MBN-SD outperforms MBN-E on the two small-scale data, COIL20 and 20-Newsgroups, and is inferior to the latter on UMIST, Extended Yale B, and COIL100. Although the result of MBN-SD is not very impressive, it introduces a new class of ensemble selection criteria—distribution divergence criteria—into clustering ensemble, which may motivate new criteria beyond MMD for further improving the performance of MBN-SD.

E. Comparison between MBN-E and fMBN-E

Table III lists the clustering accuracies of MBN-E, fMBN-E, and fMBN-Ev2. From the table, we see that MBN-E and fMBN-E achieve similar performance. This phenomenon supports the correctness of Remarks 1 and 3. Moreover, fMBN-E behaves better than fMBN-Ev2, particularly on Dermatology, New-Thyroid, and Extended Yale-B, which supports the correctness of Remark 2.

Tables IV and V summarize the running time of the comparison methods. From the tables, we see that fMBN-E is dozens of times faster than MBN-E on training the

TABLE III
ACC COMPARISON BETWEEN MBN-E, FMBN-E, AND FMBN-EV2.

	Dermatology	New-Thyroid	UMIST	Extended-Yale B	COIL20	COIL100	20-Newsgroups	MNIST
MBN-E	0.866	0.860	0.670	0.973	0.929	0.832	0.584	0.964
fMBN-E	0.868	0.907	0.659	0.964	0.938	0.837	0.582	0.964
fMBN-Ev2	0.528	0.576	0.653	0.896	0.902	0.828	0.595	0.963

TABLE IV
RUNNING TIME (IN SECONDS) OF THE BOTTOM LAYERS OF MBN-E, FMBN-E, AND FMBN-EV2.

	Dermatology	New-Thyroid	UMIST	Extended-Yale B	COIL20	COIL100	20-Newsgroups	MNIST
MBN-E	225.08	14.96	118.00	2190.72	834.64	22148.48	59997.16	979832.20
fMBN-E	0.63	0.36	3.44	70.96	24.99	679.75	1356.35	5525.12
fMBN-Ev2	0.84	0.74	0.82	2.74	1.17	20.58	278.06	1216.84

TABLE V
RUNNING TIME (IN SECONDS) OF THE UPPER LAYERS OF MBN-E, FMBN-E, AND FMBN-EV2.

	Dermatology	New-Thyroid	UMIST	Extended-Yale B	COIL20	COIL100	20-Newsgroups	MNIST
MBN-E	293.85	165.15	508.75	1829.94	1413.17	5617.11	26002.17	63939.58
fMBN-E	3.02	1.63	3.38	31.85	20.05	206.46	2085.35	9108.11
fMBN-Ev2	1.95	1.34	2.37	21.52	10.17	103.35	1141.76	8638.58

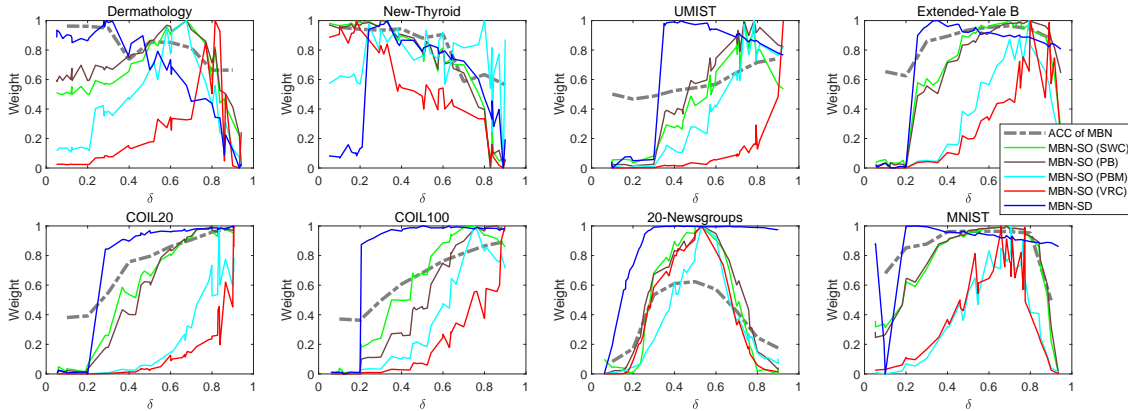


Fig. 4. Weights of the MBN base models of MBN-SO and MBN-SD.

bottom layers. Moreover, fMBN-E and fMBN-Ev2 are even hundreds of times faster than MBN-E on training the upper layers. The phenomenon supports the theoretical analysis of Theorem 3.

F. Effect of the ensemble selection algorithms on performance

This subsection studies how the proposed MBN-SO and MBN-SD affect the performance. Fig. 4 show the weights of the MBN base models of all ensemble selection methods in a single run. From the figure, we see that the weights produced by all variants of MBN-SO can cleverly reflect the quality of the base models on most datasets except Dermatology. Particularly, the weights produced by “MBN-SO (VRC)” seem to be the most accurate among the variants of MBN-SO. Although the weights produced by MBN-SD seem not as accurate as MBN-SO, if we pick a number of MBN base models, then the optimal MBN base models may be selected as well.

Based on the above observation, we study how many MBN base models, i.e. the hyperparameter B , should be selected. Specifically, we search B through $\{1, 2, 3, 5, 10\}$ respectively. From the result in Fig. 5, we see that the MBN-SO variants are not sensitive to the number of the base models on most datasets except Dermatology and New-Thyroid. Therefore, we can set the hyperparameter B of MBN-SO to a small number for saving the computing resource. On the other side, the performance of MBN-SD is generally improved when B is increased, which suggests that we should set B to a large number in order to achieve the optimal performance of MBN-SD.

G. Effect of the referenced labels of MBN-SO on performance

The optimization-like criteria of MBN-SO need referenced labels to calculate the weights of the MBN base models, where we adopt the predicted labels from MBN-E as the reference. Here we study whether MBN-SO is

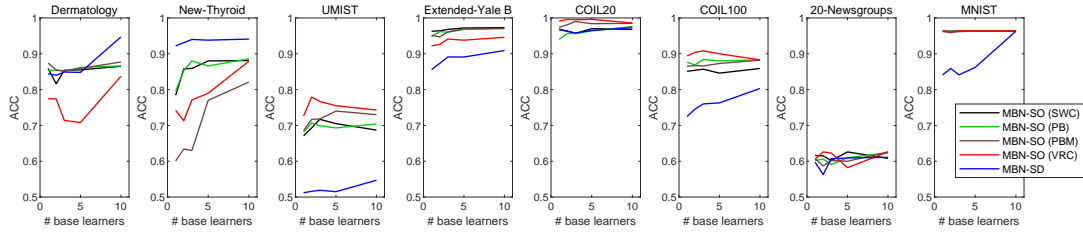


Fig. 5. Effect of the number of the selected base models of MBN-SO and MBN-SD on performance.

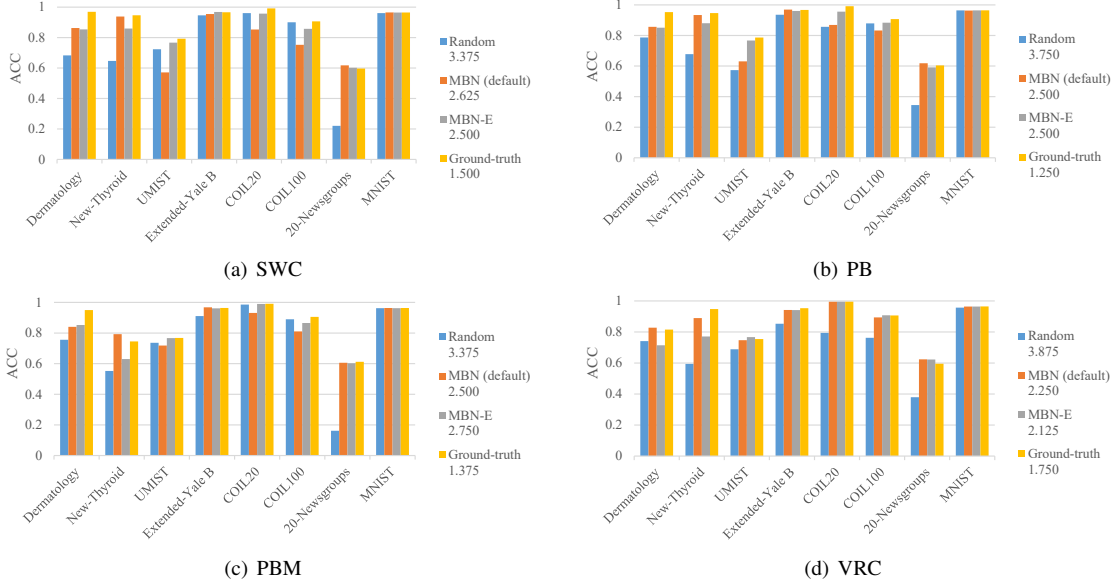


Fig. 6. Effect of the referenced labels on the performance of the MBN-SO variants. The four sub-figures show the results with the selection criteria of (a) SWC, (b) PB, (c) PBM, and (d) VRC, respectively. The numbers in the caption of each sub-figure are the ranks of the comparison methods.

sensitive to the referenced labels by generating the labels in different ways, which are (i) randomly generated labels, (ii) predicted labels from “MBN (default)”, (iii) predicted labels from MBN-E, and (iv) ground-truth labels.

Fig. 6 shows the comparison results of different referenced-label generation methods. From the figure, we observe the following interesting phenomena. First, using the predicted labels from either “MBN (default)” and MBN-E is equivalently good in terms of the ranking list. Moreover, the methods of using the predicted labels from both MBN-E and “MBN (default)” perform generally very close to the method with the ground-truth labels in terms of ACC, even though the predicted labels themselves do not have a high accuracy, e.g. on UMIST and 20-Newsgroups. In other words, MBN-SO is insensitive to the accuracy of the referenced labels.

Do the above phenomena mean that the referenced labels are unimportant? Of course no! A higher accuracy of the predicted labels do lead to better performance. If we take a look at the absolute ACC on each dataset in detail, we find that using the predicted labels from MBN-E seems a better choice than using the predicted labels from “MBN (default)”. Moreover, the method of using the ground-truth labels ranks No. 1 in all four ensemble selection criteria, while the method of using the randomly generated labels always performs the poorest.

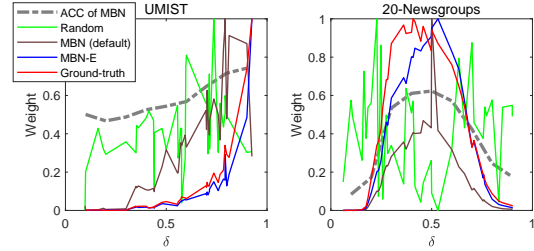


Fig. 7. Effect of different referenced-label generation methods on the weights of the base models of “MBN-SO (VRC)”.

Fig. 7 further draws the effect of the referenced labels on the weight calculation of the MBN base models on UMIST and 20-Newsgroups, where the predicted labels from MBN-E and “MBN (default)” are far less accurate than the ground-truth labels. It further manifests the correctness of the aforementioned conclusion. Specifically, from the figure, we see that, although the predicted labels are inaccurate, the weight curves of MBN-E are quite close to those produced by the ground-truth labels, which supports the empirical correctness of using MBN-E to generate the referenced labels for MBN-SO. Although the weight curves of “MBN (default)” are slightly different from those produced by the ground-truth labels, it is still able to select the top MBN base models. At last, we see that

the weight curves produced by the randomly generated labels are irregular. Comparing Fig. 7 with Fig. 6, we can further explain the phenomena why the performance with the randomly generated labels seems not so bad is because that a number of randomly selected MBN base models are able to produce a reasonable result.

H. Discussions

1) On candidate meta-clustering functions of MBN-E:

It is known that combining the base clusterings via a meta-clustering function is important for clustering ensemble technologies. In this paper, we combine the MBN base models by simply concatenating their sparse output without referring to an advanced meta-clustering function. In the Supplementary Material, we have tried 12 representative meta-clustering functions to fuse the output of the MBN base models, empirical results show that simply concatenating the outputs of the MBN base models yields similar performance to the best meta-clustering functions.

2) On candidate ensemble selection methods of MBN-SO: MBN-SO simply selects the MBN base models with the highest weights. In literature, there are many studies on how to select the base models given the weights, which may lead to higher performance and lower computational power than the proposed method. In the Supplementary Material, we have compared with 8 representative ensemble selection methods as well as their 5 variants. Empirical results show that simply picking the top MBN base models is enough to reach the highest performance, while further exploring the diversity between the base models via complicated ensemble selection algorithms is unnecessary.

VI. CONCLUSIONS

In this paper, we aim to derive a simple and tuning-free deep clustering tool that is able to yield comparable performance to the state-of-the-art deep clustering methods, for the sake of towards solving the heavy parameter-tuning problem in clustering. To achieve this goal, we propose to automatically determine the network structure of the deep clustering algorithm—MBN—by ensemble learning and selection. The proposed MBN-E simply concatenates the sparse output of a number of MBN base models with different δ to a meta-representation. The proposed MBN-SO and MBN-SD use the output of MBN-E to select the base models whose output distributions have the highest discriminability, without further exploring the diversity between the base models as conventional ensemble selection methods did. Because training an ensemble of MBN is expensive, we proposed fMBN-E, which first discards the random feature selection step of MBN and then replaces the step of random data resampling by the random resampling of similarity scores. We proved theoretically that this simplification does not degrade the estimation accuracy of MBN-E. Finally, the above methods contribute an efficient off-the-shelf deep clustering tool.

Experimental comparison results on a wide variety of benchmark datasets show that the proposed methods significantly outperform the MBN model with the default network

structure; fMBN-E is empirically hundreds of times faster than MBN-E without suffering performance degradation; MBN-SO is able to detect the optimal MBN base model, and reaches comparable performance to the state-of-the-art clustering methods; although MBN-SD is less effective than MBN-SO, it is the first work of unsupervised ensemble selection based on the distribution divergence criteria. Further studies also show that the proposed methods reach top performance via only a simple mathematical formulation, comparing to as many as 20 candidate meta-clustering functions and clustering ensemble selection functions.

REFERENCES

- [1] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [2] P. Huang, Y. Huang, W. Wang, and L. Wang, "Deep embedding network for clustering," in *2014 22nd International conference on pattern recognition*. IEEE, 2014, pp. 1532–1537.
- [3] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3861–3870.
- [4] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5736–5745.
- [5] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 24–33.
- [6] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.
- [7] C.-C. Hsu and C.-W. Lin, "Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 421–429, 2017.
- [8] F. Li, H. Qiao, and B. Zhang, "Discriminatively boosted image clustering with fully convolutional auto-encoders," *Pattern Recognition*, vol. 83, pp. 161–173, 2018.
- [9] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5147–5156.
- [10] J. Chang, G. Meng, L. Wang, S. Xiang, and C. Pan, "Deep self-evolution clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 4, pp. 809–823, 2018.
- [11] J. Wang, Z. Ma, F. Nie, and X. Li, "Progressive self-supervised clustering with novel category discovery," *IEEE Transactions on Cybernetics*, 2021.
- [12] —, "Fast self-supervised clustering with anchor graph," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [13] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39 501–39 514, 2018.
- [14] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: A generative approach to clustering," *CoRR*, 2016.
- [15] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5879–5887.
- [16] X. Guo, E. Zhu, X. Liu, and J. Yin, "Deep embedded clustering with data augmentation," in *Asian conference on machine learning*. PMLR, 2018, pp. 550–565.
- [17] W. Xia, X. Zhang, Q. Gao, and X. Gao, "Adversarial self-supervised clustering with cluster-specificity distribution," *Neurocomputing*, vol. 449, pp. 38–47, 2021.
- [18] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9865–9874.

- [19] J. Wu, K. Long, F. Wang, C. Qian, C. Li, Z. Lin, and H. Zha, "Deep comprehensive correlation mining for image clustering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8150–8159.
- [20] Z. Dang, C. Deng, X. Yang, and H. Huang, "Doubly contrastive deep clustering," *arXiv preprint arXiv:2103.05484*, 2021.
- [21] Z. Dang, C. Deng, X. Yang, K. Wei, and H. Huang, "Nearest neighbor matching for deep clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 693–13 702.
- [22] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *2021 AAAI Conference on Artificial Intelligence (AAAI)*, 2021.
- [23] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [24] L. Wu, H. Lin, Z. Gao, C. Tan, S. Li *et al.*, "Self-supervised on graphs: Contrastive, generative, or predictive," *arXiv preprint arXiv:2105.07342*, 2021.
- [25] M. Abavisani and V. M. Patel, "Deep multimodal subspace clustering networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1601–1614, 2018.
- [26] Y. Ren, N. Wang, M. Li, and Z. Xu, "Deep density-based image clustering," *Knowledge-Based Systems*, vol. 197, p. 105841, 2020.
- [27] D. Lim, R. Vidal, and B. D. Haeffele, "Doubly stochastic subspace clustering," *arXiv preprint arXiv:2011.14859*, 2020.
- [28] X.-L. Zhang, "Multilayer bootstrap networks," *Neural Networks*, vol. 103, pp. 29–43, 2018.
- [29] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.
- [30] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 03, pp. 337–372, 2011.
- [31] T. Li, C. Ding, and M. I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. IEEE, 2007, pp. 577–582.
- [32] H. Liu, M. Shao, S. Li, and Y. Fu, "Infinite ensemble for image clustering," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1745–1754.
- [33] M. Koohzadi, N. M. Charkari, and F. Ghaderi, "Unsupervised representation learning based on the deep multi-view ensemble learning," *Applied Intelligence*, vol. 50, no. 2, pp. 562–581, 2020.
- [34] T. Boongoen and N. Iam-On, "Cluster ensembles: A survey of approaches with recent extensions and applications," *Computer Science Review*, vol. 28, pp. 1–25, 2018.
- [35] M. Ganaie, M. Hu *et al.*, "Ensemble deep learning: A review," *arXiv preprint arXiv:2104.02395*, 2021.
- [36] Z.-H. Zhou and W. Tang, "Clusterer ensemble," *Knowledge-Based Systems*, vol. 19, no. 1, pp. 77–83, 2006.
- [37] X. Z. Fern and W. Lin, "Cluster ensemble selection," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 1, no. 3, pp. 128–141, 2008.
- [38] J. Azimi and X. Z. Fern, "Adaptive cluster ensemble selection," in *Ijcai*, vol. 9, 2009, pp. 992–997.
- [39] Y. Yang and K. Chen, "Temporal data clustering via weighted clustering ensemble with different representations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 2, pp. 307–320, 2010.
- [40] D. Huang, C.-D. Wang, and J.-H. Lai, "Locally weighted ensemble clustering," *IEEE transactions on cybernetics*, vol. 48, no. 5, pp. 1460–1473, 2017.
- [41] Z. Yu, X. Zhu, H.-S. Wong, J. You, J. Zhang, and G. Han, "Distribution-based cluster structure selection," *IEEE transactions on cybernetics*, vol. 47, no. 11, pp. 3554–3567, 2016.
- [42] F. Li, Y. Qian, J. Wang, C. Dang, and L. Jing, "Clustering ensemble based on sample's stability," *Artificial Intelligence*, vol. 273, pp. 37–55, 2019.
- [43] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," *Technical Report CUCS-005-96*, 1996.
- [44] M. Zhang, "Weighted clustering ensemble: A review," *arXiv preprint arXiv:1910.02433*, 2019.
- [45] T. Li and C. Ding, "Weighted consensus clustering," in *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, 2008, pp. 798–809.
- [46] J. Jia, X. Xiao, B. Liu, and L. Jiao, "Bagging-based spectral clustering ensemble selection," *Pattern Recognition Letters*, vol. 32, no. 10, pp. 1456–1467, 2011.
- [47] Y. Hong, S. Kwong, H. Wang, and Q. Ren, "Resampling-based selective clustering ensembles," *Pattern recognition letters*, vol. 30, no. 3, pp. 298–305, 2009.
- [48] F. J. F. Duarte, A. L. Fred, F. Rodrigues, J. M. Duarte, and A. Lourenco, "Weighted evidence accumulation clustering using subsampling," in *PRIS*, 2006, pp. 104–116.
- [49] L. Vendramin, R. J. Campello, and E. R. Hruschka, "Relative clustering validity criteria: A comparative overview," *Statistical analysis and data mining: the ASA data science journal*, vol. 3, no. 4, pp. 209–235, 2010.
- [50] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of intelligent information systems*, vol. 17, no. 2, pp. 107–145, 2001.
- [51] M. C. Naldi, A. Carvalho, and R. J. Campello, "Cluster ensemble selection based on relative validity indexes," *Data Mining and Knowledge Discovery*, vol. 27, no. 2, pp. 259–289, 2013.
- [52] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [53] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.
- [54] J. Maggu, A. Majumdar, E. Chouzenoux, and G. Chierchia, "Deeply transformed subspace clustering," *Signal Processing*, vol. 174, p. 107628, 2020.
- [55] J. Zhou, H. Zheng, and L. Pan, "Ensemble clustering based on dense representation," *Neurocomputing*, vol. 357, pp. 66–76, 2019.
- [56] X. Sevillano, F. Alfás, and J. C. Socoró, "Bordaconsensus: a new consensus function for soft cluster ensembles," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 743–744.
- [57] P. Zhou, Y. Hou, and J. Feng, "Deep adversarial subspace clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1596–1604.
- [58] N. Iam-On, T. Boongoen, S. Garrett, and C. Price, "A link-based approach to the cluster ensemble problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2396–2409, 2011.
- [59] D. Huang, C.-D. Wang, H. Peng, J. Lai, and C.-K. Kwok, "Enhanced ensemble clustering via fast propagation of cluster-wise similarities," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
- [60] E. Dimitriadou, A. Weingessel, and K. Hornik, "A combination scheme for fuzzy clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 07, pp. 901–912, 2002.
- [61] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3918–3927.
- [62] D. Cai, X. Wang, and X. He, "Probabilistic dyadic data analysis with local and global consistency," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 105–112.
- [63] R. McConville, R. Santos-Rodriguez, R. J. Piechocki, and I. Craddock, "N2d:(not too) deep clustering via clustering the local manifold of an autoencoded embedding," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5145–5152.
- [64] W. Zhang, X. Wang, D. Zhao, and X. Tang, "Graph degree linkage: Agglomerative clustering on a directed graph," in *European Conference on Computer Vision*. Springer, 2012, pp. 428–441.
- [65] R. Henao, Z. Gan, J. Lu, and L. Carin, "Deep poisson factor modeling," *Advances in Neural Information Processing Systems*, vol. 28, pp. 2800–2808, 2015.
- [66] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
- [67] A. Villar-Corralles and V. I. Morgenshtern, "Scattering transform based image clustering using projection onto orthogonal complement," *arXiv preprint arXiv:2011.11586*, 2020.
- [68] X. Fu, K. Huang, N. D. Sidiropoulos, Q. Shi, and M. Hong, "Anchor-free correlated topic modeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 5, pp. 1056–1071, 2018.
- [69] D. Cai, Q. Mei, J. Han, and C. Zhai, "Modeling hidden topics on document manifold," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 911–920.

- [70] M. Jabi, M. Pedersoli, A. Mitiche, and I. B. Ayed, "Deep clustering: On the link between discriminative models and k-means," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [71] A. L. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, 2005.
- [72] J. Wang and X.-L. Zhang, "Deep nmf topic modeling," *arXiv preprint arXiv:2102.12998*, 2021.