

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 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 applications to image segmentation and graph data mining further demonstrate the advantage of the proposed methods.

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, text analysis, etc. Early works, like principal component analysis (PCA) and k-means clustering, conduct clustering in the original data space. Because 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 kernel methods, probabilistic models, and manifold and subspace learning [1]. However, a proper probability space is usually found by tuning parameters manually, e.g. kernel widths [2] or regularization parameters, which is a long term headache problem. Although some work has tried to find the optimal parameters automatically, e.g. [3], 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 [4] extracts abstract representations from the bottleneck layer of a deep belief network. To make the deep representations suitable for clustering, some work adds additional terms, such as constraints [5], clustering-like loss functions and models [6], or novel network structures [7], to the network training; while some work learns deep representations and refines cluster assignments iteratively [8]. 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 [9]–[11], generative self-supervised clustering [12], [13], and contrastive self-supervised clustering [14], [15], respectively. See [16] for an recent overview. 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, data augmentation with clear intrinsic data structures, or hyperparameter tuning with the ground-truth labels. If prior knowledge is insufficient, then some methods have to make a compromise with default hyperparameter settings, e.g. [17], which may degrade performance apparently.

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, from the above literature review, it seems that this topic is far from explored yet. Because the topic is a rather broad research area, this paper focuses on the network structure selection problem of a special deep model, named *multilayer bootstrap network* (MBN) [17], given little prior knowledge of data. It seems a difficult problem, since that the network structure of MBN, which is controlled by hyperparameters, is strongly related to the unknown intrinsic property of the input data. See Section III for the details of the problem.

We address the network structure selection problem of MBN by unsupervised ensemble learning and ensemble selection. See Section II 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

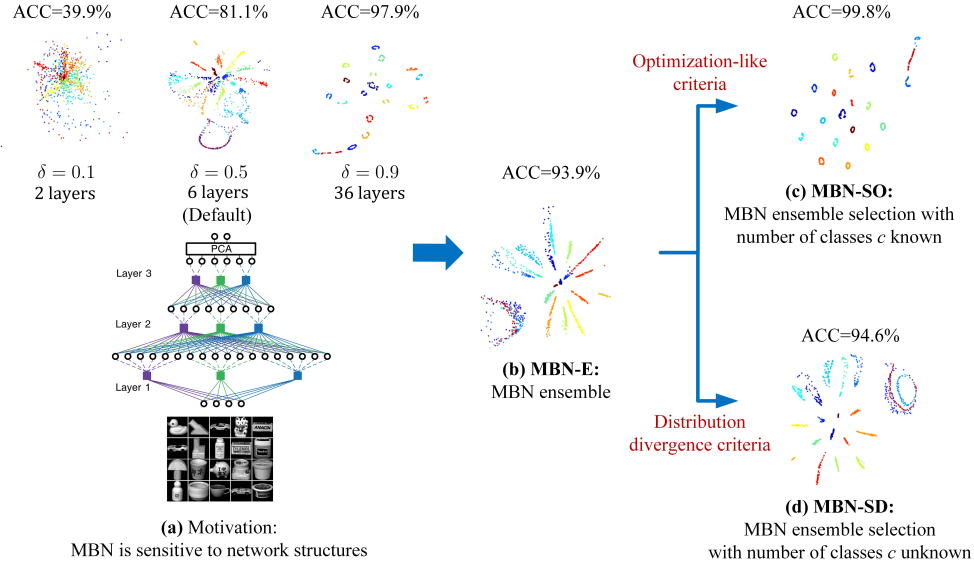


Fig. 1. On the network structure selection problem of MBN. Each square of MBN in figure (a) represents a base clustering, while the black circles connected to the square represent the input/output of the base clustering. The hyperparameter “ δ ” controls the network structure of MBN. The words in red 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 [18].

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 Appendix D of 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:

- We theoretically prove that increasing the depth of MBN does not always improve the performance, which induces the network structure selection problem of MBN.
- To address the aforementioned problem, we propose a simple MBN ensemble (MBN-E) algorithm. It groups the sparse outputs of a number of MBN base models with different network structures into a new representation.
- To reduce the high computational complexity problem of MBN-E, we propose the fast MBN-E (fMBN-E) by a simple modification of MBN-E. It accelerates MBN-E by over hundreds of times both theoretically and empirically. We have proved that the acceleration does not degrade the performance.
- To further improve the performance of MBN-E, we propose (i) the MBN ensemble selection with optimization-like criteria (MBN-SO) for the case when the number of classes is known, and (ii) the MBN ensemble selection with distribution divergence criteria (MBN-SD) when the number of classes is unknown. Both of them select a number of highly-effective MBN base models from MBN-E to group into a new MBN-E. The difference between them lies in the selection criteria of the base models.
- We have run experiments on a number of benchmark datasets where the optimal network structure of MBN

appears in fundamentally different ranges. Experimental results show that MBN-E significantly outperforms the MBN with the default setting and approaches to the 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.

- Because the proposed algorithms intend to solve the difficulty of real-world applications of MBN, we further applied the proposed methods to image segmentation and graph data mining. Experimental results verified the effectiveness of the proposed methods.

The rest of the paper is organized as follows. In Section II, we present related work. In Section III, we review MBN. In Section IV, we analyze the structure selection problem of MBN both theoretically and empirically. In Sections V and VI, we present MBN-E, fMBN-E, MBN-SO, and MBN-SD, respectively. In Section VII, we present an extensive experiment. In Section VIII, we apply the proposed methods to image segmentation and graph data mining. Finally, in Section IX, we conclude the paper.

II. RELATED WORK

The proposed MBN-E essentially is rooted in clustering ensemble. The proposed MBN-SO and MBN-SD are essentially rooted in ensemble selection and reweighting. The selection criteria of the base models of MBN-SD, which measures the divergence between data distributions, are rooted in unsupervised domain adaptation. We present the three aspects as follows.

A. Clustering ensemble

Ensemble learning, such as *bagging*, *boosting*, and their variations, has demonstrated its effectiveness on many learning

problems [19]. 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 [20], [21]. Meta-clustering functions can be categorized generally to two classes [21]. 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 [22]–[24]. Recently, some unsupervised deep ensemble learning methods have been proposed [25]–[27]. For example, [25] takes deep neural networks act like a meta-clustering function. [26] 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 [21], [28] for the reviews of clustering ensemble.

B. 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 calculating 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. [29]. A special case of this type is to constrain the weights of some weak base clusterings to zero, named *clustering selection* [30], [31]. However, weak base clusterings may also contain some high quality clusters, and vice versa. With this perspective, many reweighting strategies at levels of clusters [32], [33], data structures [34], and data points [35] were proposed.

The algorithms for calculating the weights can be categorized into two types [36]. The first type calculates weights by measuring the similarity between the predicted labels of the clustering ensemble and its base clusterings [29], [30]. The second type treats the weights as variables of consensus functions which are obtained by advanced optimization algorithms, e.g. [37].

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 [29], [30], adjusted rand index [38], clustering accuracies [39], and their variants [40] or aggregations [41] between the sets of the predicted labels. The second class of validation criteria is based on data distributions [42], [43]. They usually calculate some kinds of statistics of data [34], [44]. Some systematical studies on cluster validation indices [42], [43] have been carried out as well.

To summarize, when the number of classes is given, we evaluate the quality of the base models by *optimization-like*

criteria [42], for MBN-SO. When the number of classes is not given, we propose to evaluate the quality of the base models by so-called *distribution divergence criteria* for MBN-SD, which measure the learned representations of data directly without predicted labels.

C. 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 [45], 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 maximum mean discrepancy (MMD) [46]. 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.

In this paper, we name this kind of measurements as *distribution divergence criteria*, and apply them to MBN-SD. Because MMD performs generally well among the measurements and is applicable to all data types, from high-dimensional vectors to strings and graphs, we focus on using MMD.

III. PRELIMINARIES

This section presents MBN and its theoretical foundation briefly. See Appendices A and B of the Supplementary Material for the summary of important notations and detailed description of MBN as well as its geometric and theoretical foundations.

A. Multilayer bootstrap networks

This paper takes MBN [17] 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 clusterings, where the parameter k of all clusterings at the same layer is the same. 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. Because $\delta \in (0, 1)$, we must have

$$k_1 > k_2 > \dots > k_m > \dots > k_o \quad (2)$$

where k_o is the parameter k at the top layer. Note that, the total number of layers of MBN is usually determined automatically by k_1 , k_o , and δ .

B. Estimation error of a single layer of MBN

[17] analyzed the estimation error of a single layer of MBN, which explains the empirical success of MBN. We summarize the analysis here.

Given an input \mathbf{x} of MBN at a layer, it is easy to image that each k -centroids clustering contributes a nearest neighbor \mathbf{w}_v to \mathbf{x} , $\forall v = 1, \dots, V$, then, the new location of \mathbf{x} in the input data space, denoted as $\hat{\mathbf{x}}$, is given by the V nearest neighbors as:

$$\hat{\mathbf{x}} = \frac{1}{V} \sum_{v=1}^V \mathbf{w}_v \quad (3)$$

If $\hat{\mathbf{x}}$ is an effective estimation of \mathbf{x} , then the *locally linear assumption* between $\{\mathbf{w}_v\}_{v=1}^V$ and \mathbf{x} must hold; otherwise, $\hat{\mathbf{x}}$ is not an accurate estimation.

Under the locally linear assumption, the estimation error $\mathbb{E}(x - \hat{x})$ can be decomposed into the following form using the famous *bias-variance decomposition of expectation risk* [47]:

$$\begin{aligned} \mathbb{E}((\mathbf{x} - \hat{\mathbf{x}})^2) &= (\mathbf{x} - \mathbb{E}(\hat{\mathbf{x}}))^2 + \mathbb{E}((\mathbf{x} - \mathbb{E}(\hat{\mathbf{x}}))^2) \\ &= \text{Bias}^2(\hat{\mathbf{x}}) + \text{Var}(\hat{\mathbf{x}}) \end{aligned} \quad (4)$$

Given (4), we can derive the following theorem for the estimation error of a single layer of MBN:

Theorem 1. *The estimation error of a single layer of MBN $\mathbb{E}_{\text{ensemble}}$ and the estimation error of a single k -centroids clustering $\mathbb{E}_{\text{single}}$ in the layer have the following relationship:*

$$\mathbb{E}_{\text{ensemble}} = \left(\frac{1}{V} + \left(1 - \frac{1}{V} \right) \rho \right) \mathbb{E}_{\text{single}} \quad (5)$$

where ρ is the pairwise positive correlation coefficient between the k -centroids clusterings, $0 \leq \rho \leq 1$ [17].

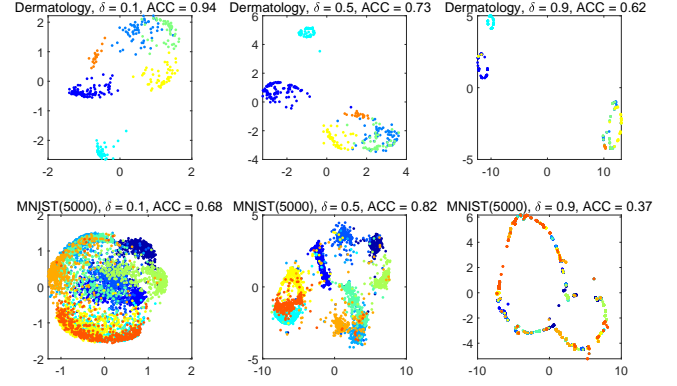


Fig. 2. Visualization of features produced by MBN with different δ on the Dermatology and MNIST(5000) datasets, where Dermatology is a dataset from UCI, and MNIST(5000) is a subset of MNIST dataset that consists of 5000 randomly selected data points.

IV. ANALYSIS OF THE NETWORK STRUCTURE PROBLEM OF MBN

It is expected that adding more layers to a deep network could improve the representation learning ability of the network. However, this is not always the case empirically, so as to MBN.

In this section, we first give an empirical demo on how different network structures affect the performance in Section IV-A, and then derive the estimation error of the entire MBN in Section IV-B by extending Theorem 1 to the multilayer scenario, which explains the empirical phenomenon theoretically and motivates the novel algorithms of this paper.

A. Empirical justification

A core problem of MBN is that its effectiveness is strongly related to the network structure which is controlled by parameter δ . Given parameters k_1 and k_o in (2) fixed, how fast k drops from k_1 to k_o layer by layer according to (2), which is determined by δ , should match the nonlinearity and noise level of data. When δ approaches to 0, MBN builds a shallow network with a single nonlinear layer, which is suitable for linearly separable data. When δ is enlarged towards 1, MBN becomes deeper and deeper, which is suitable for highly nonlinear and non-Gaussian data. If the above regularity is violated, the performance of MBN may drop sharply.

In Fig. 1a, we can see that, increasing δ from 0.1 to 0.9 yields gradually improved performance on COIL20. The gap between the best performance and poorest performance is as high as 58%. However, in Fig. 2, we see that (i) the best performance of MBN on the Dermatology dataset appears at $\delta = 0.1$, and the performance degrades gradually along with the increase of δ , which is contrary to the trend on COIL20; (ii) the best performance on MNIST(5000) appears at $\delta = 0.5$, which significantly outperforms the performance when $\delta = 0.1$ and $\delta = 0.9$. Moreover, as will be shown in Table I and Fig. 7 in the experiment, the best δ for different datasets appears at dramatically different ranges.

Because it is difficult to evaluate the properties of data in unsupervised learning, MBN has to make a compromise by

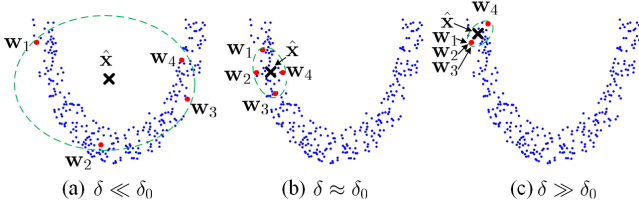


Fig. 3. Diagram of the density estimation process of MBN with different δ . The notation δ_0 denotes the optimal δ . The black cross $\hat{\mathbf{x}}$ denotes the coordinate of the learned representation of an input data \mathbf{x} . The four red points, which are \mathbf{w}_1 , \mathbf{w}_2 , \mathbf{w}_3 and \mathbf{w}_4 respectively, are the nearest centroids of four k -centroids clusterings to an input data point \mathbf{x} . The blue dotted oval is the area of the locally linear assumption.

setting $\delta = 0.5$. This may lead to far inferior performance from the optimal one, though $\delta = 0.5$ happens to be the best choice on some data like MNIST. In this paper, we aim to address this issue by detecting the optimal δ automatically.

B. Theoretical explanation

A fundamental element of MBN is the locally linear assumption defined in (3). The correctness of the assumption is strongly related to the choice of δ . Suppose the optimal performance of MBN appears at $\delta = \delta_0$. Then, a diagram in Fig. 3 explains the empirical phenomenon in Section IV-A.

When we set $\delta \ll \delta_0$, the locally linear assumption (3) may be violated, which makes MBN fail to learn correct representations. For example, in Fig. 3a, given an input data point \mathbf{x} that is sampled from the nonlinear data distribution, its representation $\hat{\mathbf{x}}$ learned by the nearest centroids \mathbf{w}_1 , \mathbf{w}_2 , \mathbf{w}_3 , and \mathbf{w}_4 is even out of the data distribution, which is clearly wrong. This explains the empirical phenomenon that MBN does not reach the top performance on COIL 20 when $\delta \ll 0.9$, and on MNIST(5000) when $\delta \ll 0.5$.

To explain the failure of MBN at $\delta \gg \delta_0$, we first give the following theorem:

Theorem 2. When $\delta > \delta_0$, the estimation error of MBN is:

$$\mathbb{E}_{\text{MBN}} \geq \sum_{m=1}^M \left(\frac{1}{V} + \left(1 - \frac{1}{V}\right) \left(\frac{ak_1}{n}\right)^2 \delta^{2(m-1)} \right) \mathbb{E}_{(\text{single},1)} \quad (6)$$

where $a \in (0, 1]$ is the ratio of the number of randomly selected features over the number of all features in Step 1 of MBN, $\mathbb{E}_{\text{single},1}$ is the estimation error of a single k -centroids clustering at the bottom layer, and M is the number of nonlinear layers of MBN.

Proof. First of all, we should emphasize that, when $\delta < \delta_0$, the locally linear assumption for (3) does not hold, which makes Theorem 1 do not hold as well. Because the following proof is built on Theorem 1, Theorem 2 is effective only when $\delta > \delta_0$.

Because the probability that any two k -centroids clusterings select the same element of the same input data point as one of their centroids is $(ak/n)^2$, then we can imagine easily that the correlation is

$$\rho = (ak/n)^2 \quad (7)$$

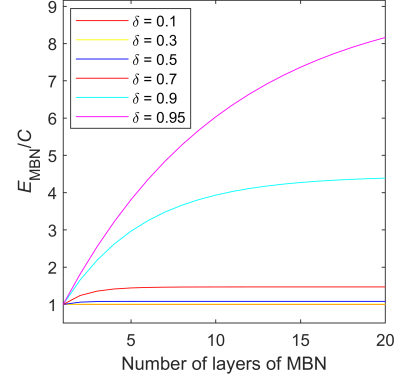


Fig. 4. Connection between the estimation error of MBN and δ when $\delta > \delta_0$, where $C = (ak_1/n)^2 \mathbb{E}_{(\text{single},1)}$ is a constant.

We denote the correlation at the m th layer as ρ_m . Substituting (1) into (7) derives

$$\rho_m = (ak_{m-1}/n)^2 \delta^2 = \dots = (ak_1/n)^2 \delta^{2(m-1)} \quad (8)$$

We denote the estimation error of a single k -centroids clustering and an ensemble of clusterings at the m th layer as $\mathbb{E}_{(\text{single},m)}$ and $\mathbb{E}_{(\text{ensemble},m)}$ respectively. Because reducing k makes $\mathbb{E}_{\text{single}}$ enlarged, we may assume that $\mathbb{E}_{(\text{single},m)}$ is lower-bounded by $\mathbb{E}_{(\text{single},1)}$. Substituting (8) into (5) derives:

$$\mathbb{E}_{(\text{ensemble},m)} \geq \left(\frac{1}{V} + \left(1 - \frac{1}{V}\right) \left(\frac{ak_1}{n}\right)^2 \delta^{2(m-1)} \right) \mathbb{E}_{(\text{single},1)} \quad (9)$$

Because \mathbb{E}_{MBN} accumulates $\mathbb{E}_{(\text{ensemble},m)}$ of all layers from bottom-up, we can derive the overall estimation error of MBN as (6). \square

We further derive the following corollary from Theorem 2:

Corollary 1. When $\delta > \delta_0$ and $V \rightarrow \infty$, the estimation error of MBN is:

$$\mathbb{E}_{\text{MBN}} \geq C \sum_{m=1}^M \delta^{2(m-1)} \quad (10)$$

where $C = (ak_1/n)^2 \mathbb{E}_{(\text{single},1)}$ is a constant.

Corollary 1 can be visualized in Fig. 4. From the figure, we see that, when δ approaches to 1, \mathbb{E}_{MBN} is increased exponentially.

Fig. 3c gives an example on how the large estimation error occurs when $\delta \gg \delta_0$. In this figure, we see that, because the four k -centroids clusterings have strong correlation, three out of four nearest centroids to \mathbf{x} , i.e. \mathbf{w}_1 , \mathbf{w}_2 , and \mathbf{w}_3 , share the same location, which makes MBN difficult to learn a good representation. The above analysis explains the phenomenon why the performance of MBN on Dermatology and MNIST(5000) drops sharply when $\delta = 0.9$.

As shown in Fig. 3b, only when $\delta \approx \delta_0$, not only the locally linear assumption holds, but also the k -centroids clusterings have weak correlation, which makes MBN learn the best representation for \mathbf{x} . However, avoiding the sensitivity of

Algorithm 1 MBN-E.

Input: A h -dimensional unlabeled dataset $\{\mathbf{x}_i\}_{i=1}^n$, parameter k_o , and number of MBN base models Z

Output: $\{\bar{\mathbf{y}}_i\}_{i=1}^n$

```

1: for  $z = 1, \dots, Z$  do
2:   Randomly generate  $\delta$  from the range  $[0.05, 0.95]$ ;
3:    $\{\mathbf{y}_{z,i}\}_{i=1}^n \leftarrow \text{MBN}(\{\mathbf{x}_i\}_{i=1}^n, k_o, \delta)$ 
4: end for
5: for  $i = 1, \dots, n$  do
6:    $\bar{\mathbf{y}}_i \leftarrow [\mathbf{y}_{1,i}^T, \mathbf{y}_{2,i}^T, \dots, \mathbf{y}_{Z,i}^T]^T$ 
7: end for

```

MBN to δ is not straightforward, which motivates the proposed methods in the following of this paper.

V. MULTILAYER BOOTSTRAP NETWORK ENSEMBLE

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

A. MBN-E

Because MBN is sensitive to δ , a straightforward thought is to integrate a number of MBN base models with different δ into MBN-E. We present MBN-E in Algorithm 1.

In Algorithm 1, we usually conduct PCA preprocessing to $\{\mathbf{x}_i\}_{i=1}^n$ 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. After getting the output $\{\bar{\mathbf{y}}_i\}_{i=1}^n$, we sometimes need to reduce $\{\bar{\mathbf{y}}_i\}_{i=1}^n$ to a low-dimensional representation $\{\bar{\mathbf{u}}_i\}_{i=1}^n$ in an Euclidian space by, e.g. PCA, for applications, since that $\{\bar{\mathbf{y}}_i\}_{i=1}^n$ is very high dimensional. Likewise, we denote the low-dimensional representation of the base models $\{\mathbf{y}_{z,i}\}_{i=1}^n$ as $\{\mathbf{u}_{z,i}\}_{i=1}^n$.

The computational complexity of MBN-E, which is Z times higher than MBN, is too high to be intolerable in practice when $Z \gg 1$:

Theorem 3. *The computational complexity of MBN-E approximates to $Z(\mathcal{O}(\alpha k V n) + \mathcal{O}(k V n))$ empirically, where $\mathcal{O}(\alpha k V n)$ and $\mathcal{O}(k V n)$ are the complexity of a single MBN at the bottom layer and the other layers respectively, and α is a constant related to the sparse property of the input data.*

B. fMBN-E

To reduce the computational complexity of MBN-E, we design a new algorithm fMBN-E in Algorithm 2. Its architecture is shown in Fig. 5. Specifically, fMBN-E and MBN-E differs in the following two aspects.

- **The first novel aspect:** fMBN-E trains a single bottom layer, instead of training Z independent bottom layers as that in MBN-E.
- **The second novel aspect:** For training each MBN base model, fMBN-E removes the random feature selection

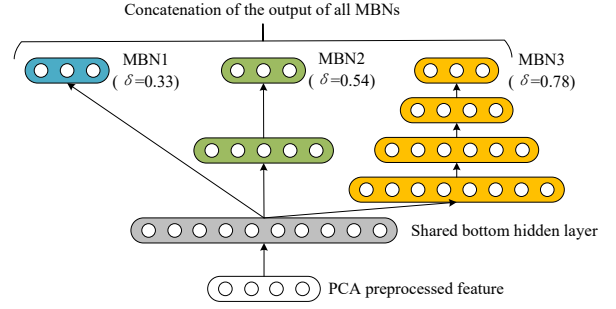


Fig. 5. Architecture of fMBN-E. Different color represents different MBN base models with random δ values.

step from MBN. This modification makes us able to train the MBN base learners by random resampling of similarity scores, instead of random resampling of data.

From the above algorithm, we can easily obtain that:

Theorem 4. *The computational complexity of fMBN-E is $\mathcal{O}(\alpha k V n) + \mathcal{O}(Z n^2)$.*

Comparing Theorems 3 and 4, 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 $(\mathcal{O}(8000\alpha n^2) + \mathcal{O}(8000n^2))$, while the complexity of fMBN-E is $\mathcal{O}(200\alpha n^2) + \mathcal{O}(40n^2)$ which may be hundreds of times faster than MBN-E. Particularly, because the complexity of the original MBN model is $(\mathcal{O}(\alpha k V n) + \mathcal{O}(k V n))$ [17], we can see that fMBN-E may be even faster than a single MBN described in [17] since that V is larger than Z in practice.

C. Analysis

Here we explain theoretically how the two novel aspects of fMBN-E reduce the computational complexity of MBN-E without suffering significant performance degradation.

1) *On the first novel aspect of fMBN-E:* Based on Theorem 1, we can draw the connections between $\mathbb{E}_{\text{ensemble}}/\mathbb{E}_{\text{single}}$, ρ , and V in Fig. 6, and further derive the following corollary from (5).

Corollary 2. *The estimation errors of the bottom layers of fMBN-E $\mathbb{E}_{\text{fMBN-E}}$ and MBN $\mathbb{E}_{\text{MBN-E}}$ have the following connection:*

$$\frac{\mathbb{E}_{\text{fMBN-E}}}{\mathbb{E}_{\text{MBN-E}}} = \frac{(\frac{1}{V} + (1 - \frac{1}{V})\rho) \mathbb{E}_{\text{single}}}{(\frac{1}{ZV} + (1 - \frac{1}{ZV})\rho) \mathbb{E}_{\text{single}}} = \frac{Z + (ZV - Z)\rho}{1 + (ZV - 1)\rho} \quad (11)$$

From Corollary 2, we can further derive the following corollary:

Corollary 3. *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:*

$$\mathbb{E}_{\text{fMBN-E}} \approx \mathbb{E}_{\text{MBN-E}} \quad (12)$$

Proof. According to Corollary 2, we see that, when V and Z are both large enough, $\mathbb{E}_{\text{fMBN-E}}/\mathbb{E}_{\text{MBN-E}}$ is determined by

Algorithm 2 fMBN-E.

Input: A h -dimensional unlabeled dataset $\{\mathbf{x}_i\}_{i=1}^n$, parameter k_o , and number of MBN base models Z

Initialization: $k_1 = \lfloor n/2 \rfloor$, number of base clusterings per layer $V = 400$

Output: $\{\bar{\mathbf{y}}_i\}_{i=1}^n$

```

1: /* train a shared bottom layer */
2:  $\{\mathbf{y}_i\}_{i=1}^n \leftarrow \text{MBN}(\{\mathbf{x}_i\}_{i=1}^n, k_1 - 1, \delta = 0)$ 
3: /* train an ensemble of fast MBN */
4: for  $z = 1, \dots, Z$  do
5:    $\mathbf{x}_{z,i} \leftarrow \mathbf{y}_i, \forall i = 1, \dots, n$ 
6:    $m \leftarrow 2$ 
7:   Randomly generate  $\delta$  from the range  $[0.05, 0.95]$ 
8:   while  $k_m \geq k_o$  do
9:     for  $v = 1, \dots, V$  do
10:      Calculate pairwise similarity matrix  $\mathbf{B} = \mathbf{X}_z^T \mathbf{X}_z$ 
        where  $\mathbf{X}_z = [\mathbf{x}_{z,1}, \dots, \mathbf{x}_{z,n}]$ 
11:      Randomly select  $k_m$  columns of  $\mathbf{B}$  to form a
        new matrix  $\mathbf{B}'$ , which is the similarity scores
        between the input data and the centroids of the
         $v$ -th clustering at the  $m$ -th layer
12:      for  $i = 1, \dots, n$  do
13:        Find the largest element of the  $i$ th row of  $\mathbf{B}$ ,
        supposed to be the  $j$ th element
14:        Derive a one-hot code  $\mathbf{s}_{i,v} = [s_{i,v,1}, \dots, s_{i,v,k_m}]^T$  where
            
$$s_{i,v,t} = \begin{cases} 1, & \text{if } t = j \\ 0, & \text{otherwise} \end{cases}, \forall t = 1, \dots, k_m$$

15:      end for
16:    end for
17:     $\mathbf{x}_{z,i} \leftarrow [\mathbf{s}_{i,1}^T, \dots, \mathbf{s}_{i,k_m}^T]^T, \forall i = 1, \dots, n$ 
18:     $k_{m+1} \leftarrow \delta k_m$ 
19:     $m \leftarrow m + 1$ 
20:  end while
21:   $\bar{\mathbf{y}}_{z,i} \leftarrow \bar{\mathbf{x}}_{z,i}, \forall i = 1, \dots, n, \forall z = 1, \dots, Z$ 
22: end for
23:  $\bar{\mathbf{y}}_i \leftarrow [\bar{\mathbf{y}}_{1,i}^T, \bar{\mathbf{y}}_{2,i}^T, \dots, \bar{\mathbf{y}}_{Z,i}^T]^T, \forall i = 1, \dots, n$ 

```

ρ . For the first case when $\rho \rightarrow 0$, $\mathbb{E}_{\text{fMBN-E}} \approx Z\mathbb{E}_{\text{MBN-E}}$; for the second case when $\rho \gg 0$, $\mathbb{E}_{\text{fMBN-E}} \approx \mathbb{E}_{\text{MBN-E}}$. In the following, we show that the second case is true.

It is easy to know that enlarging k reduces $\mathbb{E}_{\text{single}}$. From (7), we also observe that, when k is enlarged, ρ is enlarged as well. According to Theorem 1, for the bottom layer of MBN, empirically, setting k to a proper number balances $\mathbb{E}_{\text{single}}$ and ρ , which produces the minimum $\mathbb{E}_{\text{ensemble}}$. Here we take the common setting $k = n/2$ and $a = 0.5$ as an example. In this setting, we may have $\rho \approx 0.0625$, which supports that $\mathbb{E}_{\text{fMBN-E}} \approx \mathbb{E}_{\text{MBN-E}}$. Corollary 3 is proved. \square

Corollary 3 motivates us to train a single bottom layer as fMBN-E, instead of training Z independent bottom layers as MBN-E.

2) *On the second novel aspect of fMBN-E:* This subsection explains why fMBN-E is able to discard the random feature

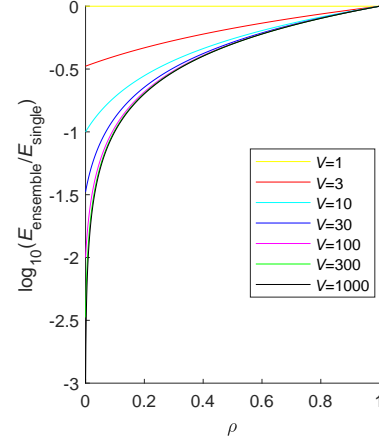


Fig. 6. Relationship between the estimation error $\mathbb{E}_{\text{ensemble}}/\mathbb{E}_{\text{single}}$, correlation coefficient ρ , and number of k -centroids clusterings per layer V .

selection step of MBN when training the upper layers.

Corollary 4. *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, the parameter k is usually far smaller than n , e.g. $k = n/2^3$ at the third layer from bottom-up. According to (7) if we remove the random feature selection step by setting $a = 1$, we may have $\rho \approx 1/2^6$. From Fig. 6, we see that $\mathbb{E}_{\text{ensemble}}$ is far smaller than $\mathbb{E}_{\text{single}}$ when $\rho \approx 1/2^6$. Therefore, we do not need the random feature selection step to further pursue a marginal reduction of $\mathbb{E}_{\text{ensemble}}$. \square

Corollary 4 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.

Following a similar explanation with the proof of Corollary 4, we can obtain:

Corollary 5. *The random feature selection step reduces the estimation error of the bottom layer of fMBN-E significantly.*

Corollary 5 motivates us to retain the random feature selection step at the bottom layer of fMBN-E.

VI. UNSUPERVISED NETWORK STRUCTURE SELECTION

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

A. Framework

Algorithm 3 presents the unsupervised ensemble selection framework for MBN-E. If the number of classes c is given, it adopts MBN-SO to select B effective MBN base models. Specifically, 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 ω_z for the z -th MBN base model by an

Algorithm 3 Unsupervised ensemble selection for MBN-E.

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

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

optimization-like criterion $f_{\text{MBN-SO}}(\{l_i\}_{i=1}^n, \{\mathbf{y}_{z,i}\}_{i=1}^n)$. The larger the weight ω_z is, the more important the corresponding MBN base model is.

If c is not given, it adopts MBN-SD to select the base models. Specifically, it first calculates the weight ω_z by evaluating the difference between 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 $\{\omega_z\}_{z=1}^Z$, it concatenates the sparse output of the B ($B \ll Z$) MBN base models whose weights are the B largest ones among $\{\omega_z\}_{z=1}^Z$ into a new sparse representation of data $\{\bar{\mathbf{x}}_i\}_{i=1}^n$.

Note that there are a vast number of ensemble selection algorithms manipulating on $\{\omega_z\}_{z=1}^Z$. 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

MBN-SO follows the comparison conclusion on the optimization-like criteria [42], and picks four best criteria, which are the silhouette width criterion (SWC), point-biserial (PB), PBM, and variance ratio criterion (VRC), respectively. Because they are defined in Euclidian spaces, MBN-SO takes the low-dimensional representations $\{\mathbf{y}_{z,i}\}_{z=1}^Z$ of the MBN base models for evaluation. Due to the length limitation of the paper, we present the four criteria in Appendix C of the Supplementary Material.

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)

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

MBN-SD adopts MMD, which is a common distribution divergence criterion in unsupervised domain adaptation, to evaluate the distribution divergence between the outputs of MBN-E and its MBN base models. See Appendix C of the Supplementary Material for the detailed derivation of MMD.

Note that, we have studied many probability distribution divergence criteria in literature, including the Kullback-Leibler divergence, total variance distance, L2-norm distance, Hellinger distance, Wasserstein distance, Bhattacharyya distance, etc. Unfortunately, they do not work for MBN-SD. However, it does not mean that MMD is the only choice, which needs further investigation in the future.

VII. EXPERIMENTS

In this section, we first compare the proposed methods with a number of representative methods on several benchmark datasets in Section VII-D. Then, we demonstrate how fMBN-E accelerates MBN-E without sacrificing accuracy in Section VII-E, and compare the ensemble selection criteria in Section VII-F. Finally, we present the experimental conclusions of some important aspects in Section VII-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 the dense scale invariant feature transform as in [48]. For 20-Newsgroups, we extracted the term frequency-inverse document frequency (TF-IDF) 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.

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 [49])	0.863 (Borda [50])	0.769 (DASC [51])	0.992 (DMSC [52])
Rank2	0.307 (LinkClueE [53])	0.859 (LinkClueE [53])	0.750 (DSC-Net-L2 [7])	0.973 (DSC-Net-L2 [7])
Rank3	0.306 (HGPA [20])	0.853 (ECPCS_MC [54])	0.732 (J-DSSC [55])	0.924 (J-DSSC [55])
Rank4	0.299 (CSPA [20])	0.851 (MCLA [20])	0.728 (DSC-Net-L1 [7])	0.917 (A-DSSC [55])
Rank5	0.297 (ECPCS_HC [54])	0.845 (Vote [56])	0.725 (A-DSSC [55])	0.776 (SSC-OMP [57])
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 [58])	0.911 (JULE [58])	0.600 (LTM [59])	0.979 (N2D [60])
Rank2	0.858 (AGDL [61])	0.824 (A-DSSC [55])	0.523 (DFPA [62])	0.969 (DDC-DA [63])
Rank3	0.858 (GDL [61])	0.796 (J-DSSC [55])	0.490 (LDA [64])	0.965 (PSSC [65])
Rank4	0.793 (DBC [66])	0.775 (DBC [66])	0.447 (AnchorFree [67])	0.964 (GDL [61])
Rank5	N/A	0.731 (GDL [61])	0.435 (LapPLSI [68])	0.939 (SR-K-means [69])
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

B. Parameter settings

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

- **MBN (default) [17]:** We used its default setting as in [17].
- **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 MBN-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. The source code is available at <http://www.xiaolei-zhang.net/mbn-e.htm>.

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 explore strong handcrafted features from augmented data [70], 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

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

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 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 the performance of text clustering is strongly related to text features. If bag-of-words is used instead of TF-IDF, then the performance of all referenced methods on 20-Newsgroups degrades significantly. To improve the performance on text clustering, new text features that incorporate context information of words may be helpful.

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

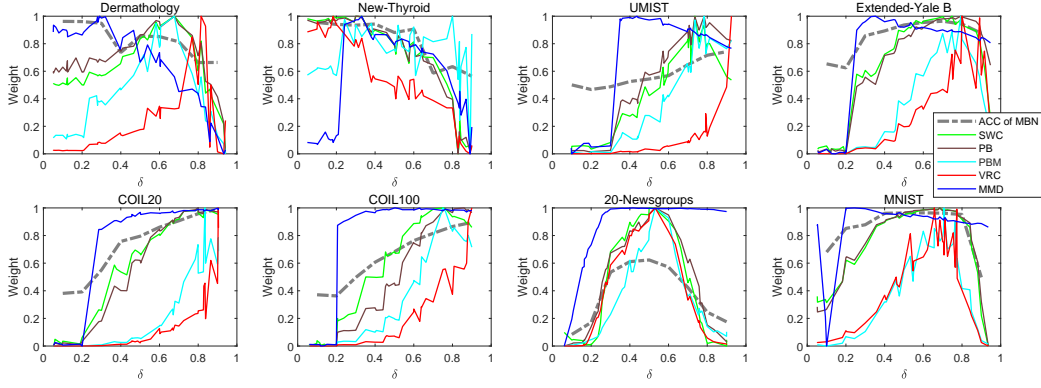


Fig. 7. Weights of the MBN base models produced by different ensemble selection criteria, where SWC, PB, PBM and VRC are optimization-like criteria for MBN-SO, and MMD is a distribution divergence criterion for MBN-SD. The dotted lines in grey color are the accuracies of MBN with respect to δ , which are references for evaluating the effectiveness of the weights.

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 Corollaries 3 and 4. Moreover, fMBN-E behaves better than fMBN-Ev2, particularly on Dermatology, New-Thyroid, and Extended Yale-B, which supports the correctness of Corollary 5.

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 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 4.

F. Comparison between different ensemble selection criteria for MBN-SO and MBN-SD

To study how different ensemble selection criteria affect the weights of the MBN base models, we compared the weights with the clustering accuracy of the MBN base models in a single run in Fig. 7. From the figure, we see that the weights produced by all ensemble selection criteria can cleverly reflect the quality of the base models on most datasets except Dermatology. Particularly, the weights produced by “VRC” seem to be the most accurate among the ensemble selection criteria. Although the weights produced by “MMD”, which is a distribution divergence criterion, seem not as accurate as the optimization-like criteria, if we pick a number of MBN base models, then the optimal MBN base models may be selected as well.

G. Discussions

This subsection reports the main conclusions of some important aspects, leaving the detailed description of the experiments in Appendix D of the Supplementary Material.

1) *Effect of number of selected base models on MBN-SO and MBN-SD*: To study how the number of MBN base models affect the performance of MBN-SO and MBN-SD, we tuned the hyperparameter B from 1 to 10. We find that, for MBN-SO, we can set the hyperparameter B to a small number for saving the computing resource; however, for MBN-SD, we should set B to a large number in order to achieve the optimal performance.

2) *Effect of the referenced labels on MBN-SO*: 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. After studying different generation methods of referenced labels, including (i) randomly generated labels, (ii) predicted labels from “MBN (default)”, (iii) predicted labels from MBN-E, and (iv) ground-truth labels, we find that the accuracy of the referenced labels has significant impact on the performance, and that the predicted labels generated from MBN-E yield good performance.

3) *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.

4) *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.

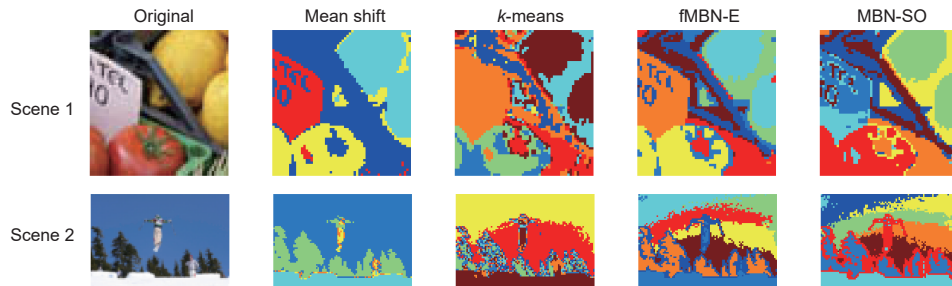


Fig. 8. Results of the image segmentation methods on 2 randomly selected examples from the 2017 Val images of the COCO datasets.

TABLE VI
DESCRIPTION OF THE GEMSEC-FACEBOOK DATASETS.

	Number of nodes	Density	Transitivity
Politicians	5,908	0.0024	0.3011
Companies	14,113	0.0005	0.1532
Athletes	13,866	0.0009	0.1292
News sites	27,917	0.0005	0.1140
Public figures	11,565	0.0010	0.1666
Artists	50,515	0.0006	0.1140
Government	7,057	0.0036	0.2238
TV shows	3,892	0.0023	0.5906

VIII. APPLICATIONS

In this section, we apply the proposed algorithms to image segmentation and graph data mining.

A. Application to image segmentation

Image segmentation partitions an image into multiple image segments, so as to simplify the analysis of the image. It is a process of assigning a label to every pixel of an image such that the pixels with the same label share certain characteristics. It is a core task of image signal processing. It can be either unsupervised or supervised. Unsupervised image segmentation, which is usually used as a preprocessing of supervised segmentation, is formulated as a clustering problem on pixels such that the pixels with similar colors and nearby locations are grouped into the same cluster.

We randomly selected several images from the 2017 Val images of the COCO datasets¹ for evaluation. We reduced the length and width of each image to about 1/7 of their original sizes, and further transformed the color space from RGB to CIELAB. Finally, for each pixel, we concatenated its three-dimensional colors and its two-dimensional coordinates as the feature. We compared with the classic mean-shift clustering and k-means clustering. The bandwidth of mean-shift was set to 0.2. The clustering number of both k-means clustering and the proposed methods was set to 8. We applied k-means clustering to the output of the proposed methods.

Two examples of the comparison results are shown in Fig. 8, while more examples are listed in Appendix E of the Supplementary Materials. From the figure, we see that the proposed methods not only maintain sufficient details of the

images than mean-shift, but also yield smoother and more accurate results than k-means. As for the proposed methods, MBN-SO behaves similarly to fMBN-E.

B. Application to graph data mining

All of the aforementioned experiments were conducted on the data whose features are given explicitly. However, the data points in many real-world applications do not have explicit features, e.g. graph data where only the connections between the data points are given. Here we give an example on how to apply the proposed methods to graph data.

Community detection is a method for finding groups within complex systems that are represented on a graph. It is a core task of network science, and finds its applications in network security, recommendation systems, etc. As collected in <https://snap.stanford.edu/data/>, the data in community detection are various sparse graphs. Here we used the undirected GEMSEC-facebook data in the collection for evaluation.

The statistics of the GEMSEC-facebook data is summarized in Table VI. For each link between a node i and a node j , we set the elements $b_{i,j}$ and $b_{j,i}$ of the graph \mathbf{B} to the weight of the link. Because the pairwise similarity between the nodes has already been given as \mathbf{B} , the output of each k -centroids clustering at the bottom layer of fMBN-E is simply a random sample of the columns of \mathbf{B} . Because the ground-truth number of communities is unknown, we used *modularity* as the evaluation metric as that in [73]. Because the modularity can be calculated in an unsupervised manner by comparing \mathbf{B} with the prediction result, we are able to search for the optimal modularity results as [73]. Specifically, we set parameter k_o of fMBN-E to $1.5c$ where c was set to 10, 20, 30, and 40 respectively. For each k_o , we grouped the nodes to 2 to 50 communities, and picked the optimal result in terms of the modularity. We applied k-means clustering to the output of the proposed methods. Following [73], we reported the average results over 5 independent runs. Table VII lists the comparison results with four well-known community detection algorithms [74]–[77]. From the average ranking over the 8 community detection tasks, we see that the proposed fMBN-E ranks the second, which is slightly worse than the walktrap algorithm [75]. Note that because MBN-SD yields almost identical performance with fMBN-E, we omit its result here.

¹<https://cocodataset.org>

TABLE VII
MODULARITY OF THE COMMUNITY DETECTION ALGORITHMS ON THE GEMSEC-FACEBOOK DATASETS. THE RESULTS OF THE REFERENCED METHODS ARE COPIED FROM [73].

	Politicians	Companies	Athletes	News sites	Public figures	Artists	Government	TV shows	Ranking
Overlap factorization [74]	0.810 (± 0.008)	0.553 (± 0.010)	0.601 (± 0.020)	0.471 (± 0.016)	0.551 (± 0.01)	0.474 (± 0.018)	0.608 (± 0.024)	0.786 (± 0.008)	4.57
Walktrap [75]	0.841 (± 0.023)	0.639 (± 0.016)	0.670 (± 0.021)	0.514 (± 0.023)	0.628 (± 0.023)	0.554 (± 0.026)	0.675 (± 0.043)	0.790 (± 0.036)	2.00
Fast greedy [76]	0.819 (± 0.008)	0.665 (± 0.014)	0.605 (± 0.026)	0.531 (± 0.020)	0.630 (± 0.011)	0.464 (± 0.023)	0.615 (± 0.046)	0.835 (± 0.006)	2.86
Label propagation [77]	0.826 (± 0.009)	0.647 (± 0.075)	0.647 (± 0.094)	0.243 (± 0.159)	0.612 (± 0.027)	0.393 (± 0.018)	0.659 (± 0.041)	0.839 (± 0.004)	3.29
fMBN-E	0.830 (± 0.004)	0.549 (± 0.011)	0.657 (± 0.002)	0.518 (± 0.014)	0.580 (± 0.015)	0.502 (± 0.003)	0.681 (± 0.009)	0.809 (± 0.005)	2.29

IX. 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 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 formulation, comparing to as many as 20 candidate meta-clustering functions and clustering ensemble selection functions. At last, we show the advantage of the proposed methods in image segmentation and graph data mining.

REFERENCES

- [1] Z. Fu, Y. Zhao, D. Chang, Y. Wang, and J. Wen, "Latent low-rank representation with weighted distance penalty for clustering," *IEEE Transactions on Cybernetics*, 2022.
- [2] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.
- [3] C. Soares, P. B. Brazdil, and P. Kuba, "A meta-learning method to select the kernel width in support vector regression," *Machine learning*, vol. 54, no. 3, pp. 195–209, 2004.
- [4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.
- [9] 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.
- [10] J. Wang, Z. Ma, F. Nie, and X. Li, "Progressive self-supervised clustering with novel category discovery," *IEEE Transactions on Cybernetics*, 2021.
- [11] T. Wang, X. Zhang, L. Lan, and Z. Luo, "Local-to-global deep clustering on approximate uniform manifold," *IEEE Transactions on Knowledge and Data Engineering*, 2022.
- [12] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: A generative approach to clustering," *CoRR*, 2016.
- [13] W. Xia, X. Zhang, Q. Gao, and X. Gao, "Adversarial self-supervised clustering with cluster-specificity distribution," *Neurocomputing*, vol. 449, pp. 38–47, 2021.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] X.-L. Zhang, "Multilayer bootstrap networks," *Neural Networks*, vol. 103, pp. 29–43, 2018.
- [18] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," *Technical Report CUCS-005-96*, 1996.
- [19] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Cagliari, Italy: Springer, 2000, pp. 1–15.
- [20] 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.

- [21] 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.
- [22] 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.
- [23] N. Nguyen and R. Caruana, "Consensus clusterings," in *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, 2007, pp. 607–612.
- [24] Z. Yu, D. Wang, X.-B. Meng, and C. P. Chen, "Clustering ensemble based on hybrid multiview clustering," *IEEE Transactions on Cybernetics*, 2020.
- [25] 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.
- [26] 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.
- [27] M. Hu and P. Suganthan, "Representation learning using deep random vector functional link networks for clustering," *Pattern Recognition*, vol. 129, p. 108744, 2022.
- [28] M. Ganaie, M. Hu *et al.*, "Ensemble deep learning: A review," *arXiv preprint arXiv:2104.02395*, 2021.
- [29] Z.-H. Zhou and W. Tang, "Cluster ensemble," *Knowledge-Based Systems*, vol. 19, no. 1, pp. 77–83, 2006.
- [30] 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.
- [31] J. Azimi and X. Z. Fern, "Adaptive cluster ensemble selection," in *Ijcai*, vol. 9, 2009, pp. 992–997.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] M. Zhang, "Weighted clustering ensemble: A review," *arXiv preprint arXiv:1910.02433*, 2019.
- [37] T. Li and C. Ding, "Weighted consensus clustering," in *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, 2008, pp. 798–809.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] D. Huang, C.-D. Wang, J.-H. Lai, and C.-K. Kwoh, "Toward multidiversified ensemble clustering of high-dimensional data: From subspaces to metrics and beyond," *IEEE Transactions on Cybernetics*, 2022.
- [42] 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.
- [43] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of intelligent information systems*, vol. 17, no. 2, pp. 107–145, 2001.
- [44] 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.
- [45] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [46] 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.
- [47] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009.
- [48] J. Maggu, A. Majumdar, E. Chouzenoux, and G. Chierchia, "Deeply transformed subspace clustering," *Signal Processing*, vol. 174, p. 107628, 2020.
- [49] J. Zhou, H. Zheng, and L. Pan, "Ensemble clustering based on dense representation," *Neurocomputing*, vol. 357, pp. 66–76, 2019.
- [50] 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.
- [51] 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.
- [52] 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.
- [53] 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.
- [54] D. Huang, C.-D. Wang, H. Peng, J. Lai, and C.-K. Kwoh, "Enhanced ensemble clustering via fast propagation of cluster-wise similarities," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
- [55] D. Lim, R. Vidal, and B. D. Haeffele, "Doubly stochastic subspace clustering," *arXiv preprint arXiv:2011.14859*, 2020.
- [56] 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.
- [57] 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.
- [58] 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.
- [59] 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.
- [60] R. McConville, R. Santos-Rodríguez, 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.
- [61] 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.
- [62] 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.
- [63] Y. Ren, N. Wang, M. Li, and Z. Xu, "Deep density-based image clustering," *Knowledge-Based Systems*, vol. 197, p. 105841, 2020.
- [64] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
- [65] A. Villar-Corralles and V. I. Morgenshtern, "Scattering transform based image clustering using projection onto orthogonal complement," *arXiv preprint arXiv:2011.11586*, 2020.
- [66] F. Li, H. Qiao, and B. Zhang, "Discriminatively boosted image clustering with fully convolutional auto-encoders," *Pattern Recognition*, vol. 83, pp. 161–173, 2018.
- [67] 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.
- [68] 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.
- [69] 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.
- [70] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [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.

- [73] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, "Gemsec: Graph embedding with self clustering," in *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019, pp. 65–72. [Online]. Available: <https://arxiv.org/pdf/1802.03997v1.pdf>
- [74] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, "Distributed large-scale natural graph factorization," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 37–48.
- [75] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *International symposium on computer and information sciences*. Springer, 2005, pp. 284–293.
- [76] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.
- [77] S. Gregory, "Finding overlapping communities in networks by label propagation," *New journal of Physics*, vol. 12, no. 10, p. 103018, 2010.