Unsupervised Ensemble Selection for Multilayer Bootstrap Networks

Xiao-Lei Zhang, *Senior Member, IEEE*

Abstract—Multilayer bootstrap network (MBN), which is a recent simple unsupervised deep model, is sensitive to its network structure. How to select a proper network structure that may be dramatically different in different applications is a hard issue, given little prior knowledge of data. In this paper, we explore ensemble learning and selection techniques for determining the optimal network structure of MBN automatically. 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 as a reference for selecting the optimal MBN base models. The ensemble selection criteria can be categorized into two classes. The first kind employs optimization-like selection criteria, under the assumption that the number of classes of data is known as a prior. The second kind proposes distribution divergence criteria, when such a prior is unavailable. Experimental results on several benchmark datasets show that MBN-E yields good performance that is close to the optimal performance of MBN, while the ensemble selection techniques for MBN-E can further improve the performance. More importantly, MBN-E and its ensemble selection techniques maintain the simple formulation of MBN, and act like off-the-shelf methods that reach the state-of-the-art performance without manual hyperparameter tuning. 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

1 Introduction

Unsupervised learning is a fundamental task of machine learning. A core problem of unsupervised learning is how to deal with non-Gaussian and linearly inseparable distributions of data. A common thought is to transform the data into linearly separable distributions by some unsupervised nonlinear algorithms, e.g. kernel methods, neural networks, probabilistic models, and ensemble methods. Because little prior knowledge is available for unsupervised learning, simple and tuning-free algorithms are always desired in practice. However, it is not easy to achieve this goal. Many factors may make an algorithm suboptimal and fragile if not properly set, such as the hyperparameters for regularization terms and probability priors, network structures, depth of trees, and kernel width of Gaussian kernels. This paper aims to address this issue for a recent simple unsupervised deep ensemble model, named *multilayer bootstrap network* (MBN) [1].

MBN can be described simply as:

- Step 1, randomly sample k data points from the input data as the centroids of a clustering, and repeat the process for V times, which learns V mutually-independent one-hot representations of the data by the clusterings.
- Step 2, stack the clustering ensemble repeatedly for M times, where the concatenation of the one-hot representations is used as the input of the upper layer.

As shown in Fig. 2a, MBN builds an M-layer deep network after the stacking. Geometrically, it builds as many as $O(k2^V)$ agglomerative hierarchical trees on the original data space, instead of on data points. The tree structure is guaranteed by setting $k_m = \delta k_{m-1}$, 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.

Although MBN with its default setting has demonstrated good performance in previous studies, "good" does not mean "optimal". Several factors affect the performance of MBN, such as how many clusterings it should have per layer? how many layers it should be set? and how large the parameter k at the bottom layer should be? Although we could find regularities on most of the problems, and use a default setting to approximate the optimal setting, how to find the optimal δ, which determines the network structure, remains unsolved. When δ approaches to 0, MBN builds a shallow network with a single nonlinear layer, which is suitable for linearly separable data. When we enlarge δ towards 1, MBN becomes deeper and deeper, which is suitable for nonlinear and non-Gaussian data.

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. As we see from Fig. 2a, the optimal performance on the object recognition

- *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.*

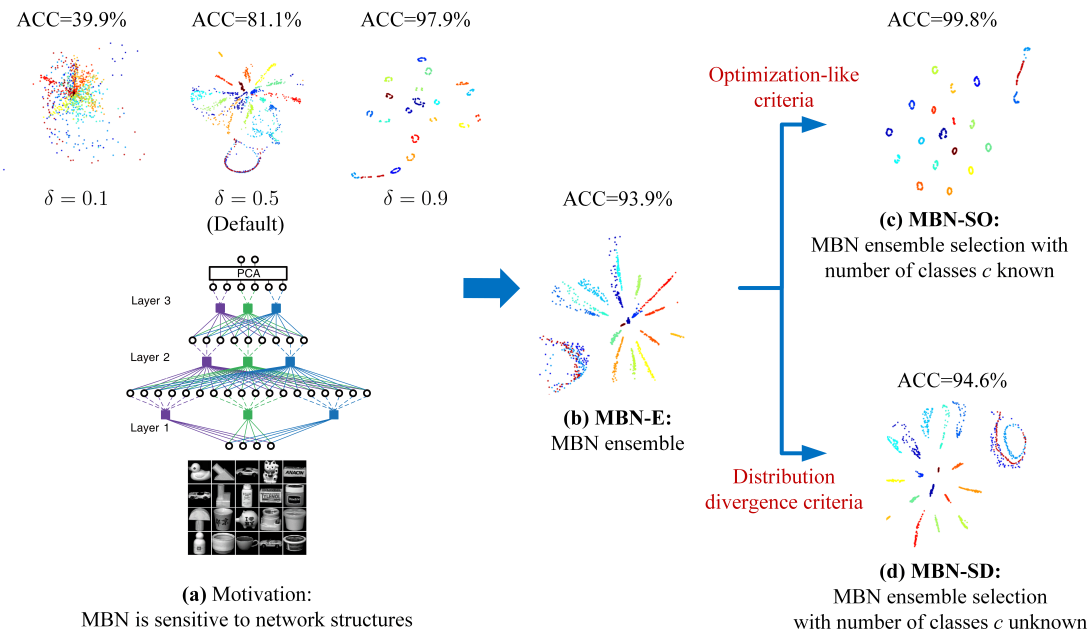

Fig. 1. **Motivation and contributions.** The hyperparameter "δ" controls the network structure of MBN. The words in red color are two ensemble selection criteria for MBN-SO and MBN-SD respectively. The word "ACC" is short for clustering accuracy. The demo data is the COIL20 dataset [2].

problem, which appears at around $\delta = 0.9$, is far better than the performance with the default setting $\delta = 0.5$. In other words, MBN needs to be built very deep to achieve the optimal performance on this highly nonlinear and non-Gaussian data. Some contrary examples can also be observed in [1, Fig. 10].

In this paper, we address the above issue by ensemble learning and ensemble selection. The contribution of this paper is summarized as follows:

- MBN ensemble (MBN-E) is proposed. It groups the sparse outputs of a number of MBN base models with different δ into a new representation. Because the main computational cost is at the bottom layer, we make the MBNs share the same bottom layer.
- 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 δ 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. MBN-SO and MBN-SD further improves the performance of MBN-E.

1.1 Related work

This subsection introduces the connection between the proposed methods, clustering ensemble, ensemble selection and reweighting, and unsupervised domain adaptation.

1.1.1 Clustering ensemble

Ensemble learning, such as *bagging* [3], *boosting* [4], and their variations, have demonstrated their effectiveness on many learning problems. Their success relies on a good selection of base models and a strong *diversity* among the base models, where the word "diversity" means that when the base models make predictions on an identical pattern, they are different from each other in terms of errors.

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 [5]–[9]. Similar to supervised ensemble learning [10], the methods for improving the diversity of the base clusterings can be partitioned generally into four groups [11]: (i) manipulating training examples [6], (ii) manipulating input features, (iii) manipulating training parameters [7], [12], [13], and (iv) manipulating different clustering algorithms [5]. Meta-clustering functions can be categorized generally to two classes [9]. 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 [14]. See [9], [15], [16] for the reviews of clustering ensemble.

Recently, some unsupervised deep ensemble learning has been proposed. In [17], the authors first learn a co-occurrence matrix of data by a number of basis clusterings, and then use the co-occurrence matrix as the input of a deep auto-encoder to refine the matrix. In [18], the authors aggregate an ensemble of auto-encoders for text summarization, where random noise is added into the data for enlarging the diversity between the auto-encoders. In [19], the authors decompose each layer of a deep neural network into an ensemble of encoders or decoders and mask operations, where different encoders or decoders capture different local complementary information. However, to our knowledge, unsupervised deep ensemble learning is not prevalent, due to that neural networks need supervised signals to maximize its discriminant ability.

To summarize, MBN-E is an unsupervised deep ensemble learning method. It learns deep representations without resorting to neural networks and hyperparameter tuning.

1.1.2 Clustering ensemble reweighting and selection

Most of the aforementioned research assigns the same weight to each base clustering. However, not all base clusterings contribute equivalently to the ensemble. Some base clusterings may contribute negatively to the ensemble, while some may be highly correlated with each other. Therefore, 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. [20]. A special case of this type is to constrain the weights of some weak base clusterings to zero, named *clustering selection* [21], [22]. However, weak base clusterings may also contain some high quality clusters, and vise versa. With this perspective, many reweighting strategies at levels of clusters [23], [24], data structures [25], and data points [26] were proposed.

The algorithms for determining the weights can be categorized into two types [27]. The first type calculates weights from the given clustering ensemble using specific validation criteria, e.g. [20], [21]. The weights reflecting the quality of the base clusterings are usually calculated by measuring the similarity between the predicted labels of the clustering ensemble and its base clusterings. The weights reflecting the diversity between the base clusterings are usually calculated according to the pairwise dissimilarity between the predicted labels of the base clusterings. A common algorithm for reweighting the base clusterings is to reweight the co-occurrence matrix. The second type treats the weights as variables of consensus functions which are obtained by advanced optimization algorithms, e.g. [28].

The criteria for measuring the diversity and quality of the base models [25], [29]–[33] can be categorized into two classes. The first class of measurements calculates the normalized mutual information [20], [21], adjusted rand index [34], clustering accuracies [35], and their variants [36] between the sets of the predicted labels. However, the predicted labels do not contain the information of data distributions. As we know, even if two sets of predicted labels are exactly the same, their underlying data distributions can be dramatically different. To remedy this weakness, the second class of validation criteria is based on data distributions [30], [32]. They usually calculate some kinds of statistics of data. For example, Naldi *et al,* [33] studied six cluster validation indices based on distances between samples and their first- and second-order statistics. Yu *et al.* [25] studied the first and second order statistics of Gaussian mixture models as new validation indices. Some systematical studies on cluster validation indices [30], [32] have been carried out as well.

To our knowledge, all cluster validation criteria in the second class takes the predicted labels of data as a requirement of measuring the discriminant ability of the data distribution. However, when the number of clusters is not given explicitly, the predicted labels may be highly unreliable. Moreover, unsupervised ensemble learning is beyond clustering ensemble. If we view the predicted labels as a representation of data, then clustering ensemble may be regarded as a special case of unsupervised ensemble learning. Therefore, it is needed to develop new validation criteria that measure the divergence of data distributions directly.

To summarize, when the number of classes is given, we evaluate the quality of the base models by *optimization-like criteria* [32], 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 di-*

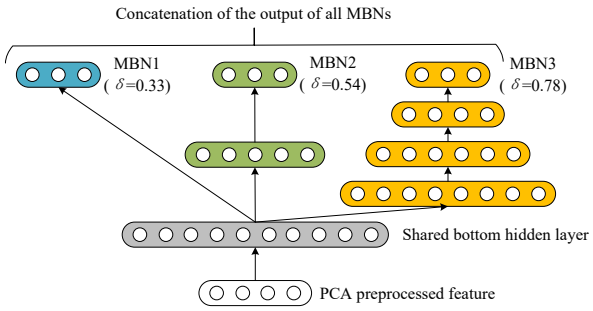

Fig. 2. **Architecture of MBN-E.** Different color represents different MBN base models with random δ values.

vergence criteria for MBN-SD, which measure the learned representations of data directly without predicted labels. To our knowledge, this is the first time that the distribution divergence criteria are applied to the unsupervised ensemble selection problem.

1.1.3 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 [37], 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 [38]. Other measurements include Kullback-Leibler divergence [39], total variation distance, second-order (covariance) statistics [40], and Hellinger distance [41]. Some very recent work incorporates supervised information into the measurement of the distribution divergence, e.g. [42].

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 discussing MMD.

2 Multilayer Bootstrap Network Ensemble

The architecture of MBN-E is shown in Fig. 2. It is an ensemble of MBN base models who share the same bottom layer and have different δ values. We present MBN-E in detail as follows:

- **Step 1: Share the bottom layer.**
  Given a dataset of n data points, MBN-E first trains a bottom layer. The bottom layer consists of V k_1-centroids clusterings, where the parameter $k_1 = n/2$. The centroids of each clustering are k_1 randomly sampled data points from the dataset. For each data point, a k_1-centroids clustering outputs a k_1-dimensional one-hot code indicating which centroid the data point belongs to. The V one-hot codes of the data point is concatenated as a new representation of the data.
- **Step 2: Train an ensemble of base MBN learners.**
  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 $k_m = \delta k_{m-1}$ where m is the index of the m-th layer. The training process of each layer of the MBN model is the same as that of the bottom layer. The entire training process stops when k_m reaches a predefined value k_o ($k_o \ll k_1$).
- **Step 3: 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 data. If we denote the output of the z-th MBN base model as $\{\mathbf{x}_{z,i}\}_{i=1}^n, \forall z = 1, \ldots, Z$, then the output representation of MBN-E is $\bar{\mathbf{x}}_i = [\mathbf{x}_{1,i}^T, \ldots, \mathbf{x}_{z,i}^T, \ldots, \mathbf{x}_{Z,i}^T]^T, \forall i = 1, \ldots, n$.

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 $\{\mathbf{y}_{z,i}\}_{i=1}^n$.

The reason why we make the MBN base models share the bottom layer is that the main computational complexity of MBN is at the bottom layer. For the same reason, we usually conduct PCA preprocessing before MBN-E, which not only reduces the computational complexity of the bottom layer but also de-correlates the input features.

Making the MBN base models share the same bottom layer will not affect the diversity between the base models. From the geometric analysis of MBN in [1], we know that the bottom layer of an MBN model will partition its input data space to as many as $O(k_1 2^V)$ fractions, which maintain sufficient details of data for the upper layers to generate accurate and diverse representations.

3 Unsupervised Ensemble Selection for MBN-E

In this section, we first present an unsupervised ensemble selection framework for MBN-E in Section 3.1, and then present MBN-SO and MBN-SD in Sections 3.2 and 3.3 respectively.

3.1 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 an 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.

3.2 MBN-SO: Ensemble selection with optimization-like criteria

When the number of classes c is given, we use optimization-like criteria to generate the weights of the base models. We follow the comparison conclusion on the optimization-like criteria [32], and pick the 4 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, 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:

3.2.1 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 $p \in \{1, \ldots, c\}$. 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 $q = 1, \ldots, c$, $q \neq p$. Then, the silhouette of $\mathbf{y}_i$ is defined as:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \tag{1}$$

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:

$$w^{\text{SWC}} = \frac{1}{n}\sum_{i=1}^n s_i \tag{2}$$

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

3.2.2 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:

$$d_w = \frac{1}{n}\sum_{i=1}^n a_i \tag{3}$$

$$d_b = \frac{1}{n}\sum_{i=1}^n \sum_{\{q|q=1,\ldots,c,q\neq p\}} \frac{n_q}{n - n_p} g_{q,i} \tag{4}$$

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 $q = 1, \ldots, c$ and $q \neq p$. Then, it is defined as:

$$w^{\text{PB}} = \frac{(d_b - d_w)\sqrt{w_d b_d / t^2}}{s_d} \tag{5}$$

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.

3.2.3 PBM

PBM is defined over between-class distances and within-class distances:

$$w^{\text{PBM}} = \left(\frac{1}{k}\frac{E_1}{E_K} D_K\right)^2 \tag{6}$$

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: $\{\bar{\bar{\mathbf{x}}}_i\}_{i=1}^n$, $\{\bar{\bar{\mathbf{y}}}_i\}_{i=1}^n$.

```
if c is given then
    {l_i}_{i=1}^n ← clustering({ȳ_i}_{i=1}^n, c)
    for z = 1 to Z do
        w_z ← f_MBN-SO({l_i}_{i=1}^n, {y_{z,i}}_{i=1}^n)
          (or w_z ← f_MBN-SO({l_i}_{i=1}^n, {x_{z,i}}_{i=1}^n))
    end for
else
    for z = 1 to Z do
        w_z ← f_MBN-SD({x̄_i}_{i=1}^n, {x_{z,i}}_{i=1}^n)
          (or w_z ← f_MBN-SD({ȳ_i}_{i=1}^n, {y_{z,i}}_{i=1}^n))
    end for
end if
Pick B sparse representations that correspond to the B largest weights, supposed to be {{x_{b,i}}_{i=1}^n}_{b=1}^B without loss of generality
x̄̄_i ← [x_{1,i}^T, ..., x_{B,i}^T]^T
ȳ̄_i ← PCA(x̄_i)
```

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:

$$E_1=\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{y}_i-\bar{\boldsymbol{\mu}}\| \tag{7}$$

$$E_K=\frac{1}{n}\sum_{p=1}^{c}\sum_{\{\mathbf{y}_i|l_i=p\}}\|\mathbf{y}_i-\boldsymbol{\mu}_p\| \tag{8}$$

$$D_K=\max_{p,q=1,\ldots,c}\|\boldsymbol{\mu}_p-\boldsymbol{\mu}_q\| \tag{9}$$

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.

3.2.4 Variance ratio criterion

VRC calculates the ratio of the between-class variance over within-class variance:

$$w^{\mathrm{VRC}}=\frac{1}{h}\frac{n-c}{c-1}\frac{\mathrm{tr}(\mathbf{B})}{\mathrm{tr}(\mathbf{W})} \tag{10}$$

where $\mathrm{tr}(\cdot)$ 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:

$$\mathbf{W}=\sum_{p=1}^{c}\mathbf{W}_p \tag{11}$$

$$\mathbf{W}_p=\sum_{\{\mathbf{y}_i|l_i=p\}}(\mathbf{y}_i-\boldsymbol{\mu}_p)(\mathbf{y}_i-\boldsymbol{\mu}_p)^T \tag{12}$$

$$\mathbf{B}=\sum_{p=1}^{c}n_p(\boldsymbol{\mu}_p-\bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_p-\bar{\boldsymbol{\mu}})^T \tag{13}$$

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.

3.3 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.

TABLE 1
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	$(0.35, 0.75)$

3.3.1 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 $\bar{\mathbf{x}}_i=[\mathbf{x}_{1,i}^T,\ldots,\mathbf{x}_{Z,i}^T]^T$, here we define MMD as follows:

$$v^{\mathrm{MMD}}=\frac{1}{Z}\frac{1}{n(n-1)}\sum_{i\neq j}\bar{\mathbf{x}}_i^T\bar{\mathbf{x}}_j + \frac{1}{n(n-1)}\sum_{i\neq j}\mathbf{x}_{z,i}^T\mathbf{x}_{z,j}-\frac{2}{Z}\frac{1}{n^2}\sum_{u=1}^{Z}\sum_{i,j}\mathbf{x}_{u,i}^T\mathbf{x}_{z,j} \tag{14}$$

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:

$$w^{\mathrm{MMD}}=1-\frac{v^{\mathrm{MMD}}-v_{\min}}{v_{\max}-v_{\min}} \tag{15}$$

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

4 Experiments

In this section, we first compare the proposed methods with the state-of-the-art methods on a number of benchmark datasets, then demonstrate the effectiveness of MBN-E by comparing it with 12 representative meta-clustering functions that use the same MBN base models, and finally demonstrate the effectiveness of MBN-SO and MBN-SD by comparing them with 5 representative cluster ensemble selection functions. The effect of the number of the selected base models on performance will be studied as well.

4.1 Datasets

We selected 8 benchmark datasets as summarized in Table 1. 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 [43]. 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 dimension. 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.

4.2 Parameter settings

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

- **MBN [1]:** We used its default setting as in [1]. Specifically, the number of k-centroids clusterings per layer was set to 400. The parameter k at the bottom layer was set to $n/2$ where n is the number of input data points. The parameter k at the top layer was set to $1.5c$. The parameter δ was set to 0.5. If PCA preprocessing was used, then random feature selection step was enabled. The ratio of the randomly selected features a was set to 0.5. The above method is denoted as "MBN (default)".
- **MBN-E:** MBN-E 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 $[0.05, 0.95]$.
- **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.

4.3 General comparison with the state-of-the-art 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 here we omit self-supervised deep learning based methods, given that they actually have strong handcrafted supervised signals to train the networks, where the signals are efficiently obtained from the intrinsic structure of data instead of heavy manual labeling. Note also that many of the referenced image clustering methods have adopted prior knowledge (e.g. convolutional structures for image data, or multi-modal information [47]), data augmentation (e.g. DDC-DA [59]), hyperparameter tuning with the ground-truth labels (e.g. DSSC [51]), etc., that our methods do not use. 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, which are CSPA [5], HGPA [5], MCLA [5], DREC [44], LinkClueE [48], [66], ARA1 [67], ARA2 [67], Borda [45], Cvote [13], Vote [52], ECPCS_MC [50], and ECPCS_HC [50], respectively. 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. [7], 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: four probabilistic models, including PLSI [68], LDA [60], LapPLSI [64], and LTM [55]; four nonnegative matrix factorization methods, including SPA [69], SNPA [70], XRAY [71], AnchorFree [63]; as well as one deep learning based method, i.e. DFPA [58]. 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.

Table 2 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. We also list the performance of the MBN with the optimal δ, denoted as MBN†. Note that because it is unlikely to select the optimal δ manually in real-world applications, MBN† 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

TABLE 2
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.

	Dermatology	New-Thyroid	UMIST*	Extended-Yale B*
kmeans	0.261	0.860	0.408	0.311
Rank1	0.313 (DREC [44])	0.863 (Borda [45])	**0.769 (DASC [46])**	**0.992 (DMSC [47])**
Rank2	0.307 (LinkClueE [48])	0.859 (LinkClueE [48])	0.750 (DSC-Net-L2 [49])	0.973 (DSC-Net-L2 [49])
Rank3	0.306 (HGPA [5])	0.853 (ECPCS_MC [50])	0.732 (J-DSSC [51]))	0.924 (J-DSSC [51]))
Rank4	0.299 (CSPA [5])	0.851 (MCLA [5])	0.728 (DSC-Net-L1 [49])	0.917 (A-DSSC [51])
Rank5	0.297 (ECPCS_HC [50])	0.845 (Vote [52])	0.725 (A-DSSC [51]))	0.776 (SSC-OMP [53])
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 [54])**	**0.911 (JULE [54])**	0.600 (LTM [55])	**0.979 (N2D [56])**
Rank2	0.858 (AGDL [57])	0.824 (A-DSSC [51])	0.523 (DFPA [58])	0.969 (DDC-DA [59])
Rank3	0.858 (GDL [57])	0.796 (J-DSSC [51]))	0.490 (LDA [60])	0.965 (PSSC [61])
Rank4	0.793 (DBC [62])	0.775 (DBC [62])	0.447 (AnchorFree [63])	0.964 (GDL [57])
Rank5	No	0.731 (GDL [57])	0.435 (LapPLSI [64])	0.939 (SR-K-means [65])
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

in the optimal operating range, MBN-E can still be competitive to "MBN (default)" if the 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 [5], [13], [44], [45], [48], [50], [52], [66], [67] do not show significant performance improvement over k-means. 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 [58] is inferior to the conventional probabilistic method LTM [55].

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.

4.4 Comparison with meta-clustering functions

MBN-E concatenates the learned representations from the MBN base models as a new meta-representation for clustering, while a conventional clustering ensemble method usually uses a meta-clustering function to fuse the predictions produced from a number of

TABLE 3
ACC comparison between MBN-E and the meta-clustering functions that use the same MBN base models as MBN-E. The abbreviations "Derm.", "NT", "Yale B", and "20-NG" are short for Dermatology, New-Thyroid, Extended-Yale B, and 20-Newsgroups, respectively. The term "N/A" means that a single run cannot be finished in 24 hours.

	Derm.	NT	UMIST	Yale B	COIL20	COIL100	20-NG	MNIST	Rank
CSPA [5]	0.721	0.491	0.592	0.966	0.816	0.677	0.581	0.106	8.125
HGPA [5]	0.306	0.698	0.083	0.027	0.050	0.010	0.053	0.113	12.000
MCLA [5]	0.791	**0.949**	0.602	0.961	0.830	0.726	0.586	**0.965**	5.125
DREC [44]	0.669	0.777	0.500	0.684	0.619	0.545	0.401	N/A	10.875
LinkClueE [66]	0.891	0.948	0.651	0.917	0.894	0.796	N/A	N/A	5.875
ARA1 [67]	0.866	0.897	0.587	0.921	0.837	0.586	0.578	N/A	7.750
ARA2 [67]	0.848	0.937	0.431	0.834	0.757	0.399	0.494	N/A	9.875
Borda [45]	0.922	0.940	0.539	0.888	0.656	0.536	0.516	**0.965**	7.375
Cvote [13]	0.685	0.683	0.631	0.965	**0.981**	0.831	0.204	**0.965**	5.750
Vote [52]	0.867	0.880	0.649	0.968	0.930	0.825	0.618	**0.965**	**3.250**
ECPCS_MC [50]	**0.935**	0.940	0.598	0.947	0.884	0.784	**0.633**	**0.965**	4.125
ECPCS_HC [50]	0.852	0.943	0.597	0.816	0.857	0.765	0.431	0.694	7.000
MBN-E	0.866	0.860	**0.670**	**0.973**	0.929	**0.832**	0.584	0.964	3.875
MBN†	0.971	0.964	0.770	0.969	0.994	0.901	0.623	0.965	

base clusterings. To evaluate whether this simple meta-representation exploits the potential of MBN-E fully, we compared it with the 12 meta-clustering functions described in Section 4.3. Unlike Section 4.3, the predictions of data for the meta-clustering functions here is obtained by applying agglomerative hierarchical clustering to the learned representations of the MBN base models.

Table 3 lists the comparison results of the MBN-E and 12 meta-clusterings that use the same MBN base models. From the table, we find that the proposed MBN-E ranks the second place, which is slightly worse than Vote [52]. If we look at the details, we find that MBN-E performs only 0.1% worse than Vote on Dermatology, COIL20, and MNIST, which accounts for the inferiority of MBN-E over Vote. We further observe that MBN-E wins the best performance on three datasets, which has the same highest number of championships as ECPCS_MC [50]. Comparing Table 3 with Table 2 on Dermatology and New-Thyroid, we find that it is the selection of a good base model rather than a carefully designed meta-clustering that determines the performance. To summarize, considering the "Occam's Razor" as the principle for designing algorithms, the simple MBN-E is recommended as the best choice of fusing multiple MBN base models.

If we further compare the results in Table 3 with MBN†, we find that none of the 13 comparison methods achieve comparable performance with MBN†—one of the base models that has been applied to all of the comparison methods. This phenomenon suggests that, if we could find MBN† from the candidate base models, then the performance could at least outperform the comparison methods, which motivates the invention of MBN-SO and MBN-SD.

4.5 Comparison with clustering ensemble selection functions

This section compares MBN-SO with five representative clustering ensemble selection functions, given the same MBN base models. They can be categorized into two classes. The first class conducts the ensemble selection according to the clustering results of the base models only. It consists of the sum of the normalized mutual information (SNMI) [21], joint criterion (JC) [21], and cluster and select (CAS) [21]. The selection criteria of the methods consider both the accuracy and diversity of the clustering results. The second class [33] picks the base models according to an optimization-like criterion, which is closely related to the proposed MBN-SO. Here we compare with the following representative ones:

- **Single index selection (SIS) [33]**: Contrary to MBN-SO which uses the predicted label from MBN-E as a reference to evaluate the discriminability of the output representation of each base model, SIS uses the predicted label from each base clustering as a reference to evaluate the discriminability of the original data representation, and uses a meta-clustering function to fuse the predicted labels from the top B base clusterings into the final prediction result. Because the original data representation is very noisy, we replaced it with the output representation of MBN-E, which improves SIS to a fair experimental setting with MBN-SO. Here we apply the criteria of SWC, PB, PBW, and VRC to SIS for a point-to-point comparison with MBN-SO. Following [33], we used CSPA as the meta-clustering function of SIS.
- **Sum of ranks (SR) [33]** It runs SIS with different optimization-like criteria, each of which produces a

TABLE 4
ACC comparison between MBN-SO and the clustering ensemble selection functions that use the same candidate MBN base models as MBN-SO.

	Derm.	NT	UMIST	Yale B	COIL20	COIL100	20-NG	MNIST	**Rank**
SNMI [21]	0.708	0.485	0.555	0.823	0.726	0.608	0.534	0.106	15.375
JC [21]	0.746	0.537	0.546	0.947	0.873	0.800	0.556	0.106	11.250
CAS [21]	0.734	0.479	0.560	0.940	0.698	0.617	0.462	0.106	14.250
SIS (SWC) [33]	0.686	0.528	0.559	0.929	0.880	0.776	0.544	0.106	13.000
SIS (PB) [33]	0.682	0.494	0.572	0.930	0.898	0.771	0.544	0.106	12.875
SIS (PBM) [33]	0.658	0.486	0.587	0.910	0.892	0.808	0.483	0.106	13.250
SIS (VRC) [33]	0.643	0.522	0.634	0.909	0.963	0.809	0.545	0.106	11.125
SR [33]	0.645	0.509	0.567	0.924	0.889	0.790	0.532	0.106	13.625
MBN-SO (SWC)	0.854	0.859	0.717	**0.968**	0.957	0.857	0.602	0.964	4.500
MBN-SO (PB)	0.851	0.880	0.699	0.960	0.956	0.884	0.591	0.964	5.250
MBN-SO (PBM)	0.852	0.630	0.718	0.961	0.990	0.866	0.602	0.962	4.750
MBN-SO (VRC)	0.714	0.771	**0.767**	0.941	**0.995**	**0.908**	**0.623**	0.964	4.750
MBN-SD	0.849	**0.940**	0.519	0.891	0.958	0.760	0.607	0.841	9.750
rSNMI	0.730	0.565	0.552	0.949	0.873	0.796	0.556	0.106	11.500
rMBN-SO (SWC)	0.867	0.885	0.625	0.966	0.920	0.823	0.611	**0.965**	5.125
rMBN-SO (PB)	0.806	0.938	0.656	0.934	0.965	0.852	0.617	**0.965**	4.625
rMBN-SO (PBM)	**0.905**	0.937	0.626	0.954	0.953	0.821	0.605	0.964	5.625
rMBN-SO (VRC)	0.855	0.937	0.654	0.945	0.952	0.830	0.611	0.962	5.875

ranking of the base models. Then, it averages the rankings for the final ranking of the base models. At last, it uses a meta-clustering function to fuse the predicted labels from the top B base clusterings into the final prediction result. Following [33], we used CSPA as the meta-clustering function of SR.

The top 2 parts of Table 4 lists the comparison result between MBN-SO and the referenced methods [21], [33]. From the ranking list of the table, we see that the variants of MBN-SO behave similarly with each other, and outperform the referenced methods apparently. The variants of SIS perform similarly as well, which outperform SNMI and CAS, and are inferior to JC. If we look at the details, we find that "MBN-SO (VRC)" achieves the top performance in five out of the eight datasets. As for the referenced methods, most of them do not behave fundamentally different. Particularly, they have failed to achieve reasonable results on MNIST, comparing to random guess.

Figs. 3 and 4 show the weights of the MBN base models of all comparison methods in a single run. After comparing the curves of the weights with the clustering accuracy of the MBN base models, we see that although the weights of MBN-SO are more accurate than the weights of the SIS variants, the performance gap between SIS and MBN-SO in Table 4 seem unnecessarily to be so large.

To investigate why the proposed MBN-SO has such a large advantage over the referenced methods, we first removed the ensemble selection criterion based on diversity in SNMI by simply picking the B base models that have the largest weights. The new method is named *revised SNMI* (rSNMI). From the result in Table 4, we see that rSNMI significantly outperforms SNMI and CAS, and performs as good as JC. That is to say, a simple ensemble selection strategy like MBN-SO is enough, while further exploring the diversity between the base models via complicated algorithms is unnecessary.

Then, we replaced the meta-clustering function of SIS by simply concatenating the output representations of the selected base models. Because the only difference between the revised algorithm and MBN-SO is that the revised algorithm uses the data representation produced by MBN-E as a reference to evaluate the clustering quality of each MBN base model, while MBN-SO uses the clustering result of MBN-E as a reference to evaluate the data representation learned by each MBN base model, we name the revised algorithm as *revised MBN-SO* (rMBN-SO). The bottom 2 parts of Table 4 lists the comparison result between MBN-SO and rMBN-SO. From the apple-to-apple comparison, we see that the ensemble selection strategy of MBN-SO is better than rMBN-SO. By comparing rMBN-SO and SIS, we see that the meta-clustering function is responsible for the large performance gap between MBN-SO and SIS.

4.6 Effect of the number of selected base models on performance

This subsection studies whether the proposed ensemble selection methods are sensitive to the number of the selected base models, i.e. hyperparameter B. For MBN-SO and MBN-SD, we set B to $\{1, 2, 3, 5, 10\}$ respectively. From the result in Fig. 5, we see that the MBN-SO variants are not sensitive to the number of the base

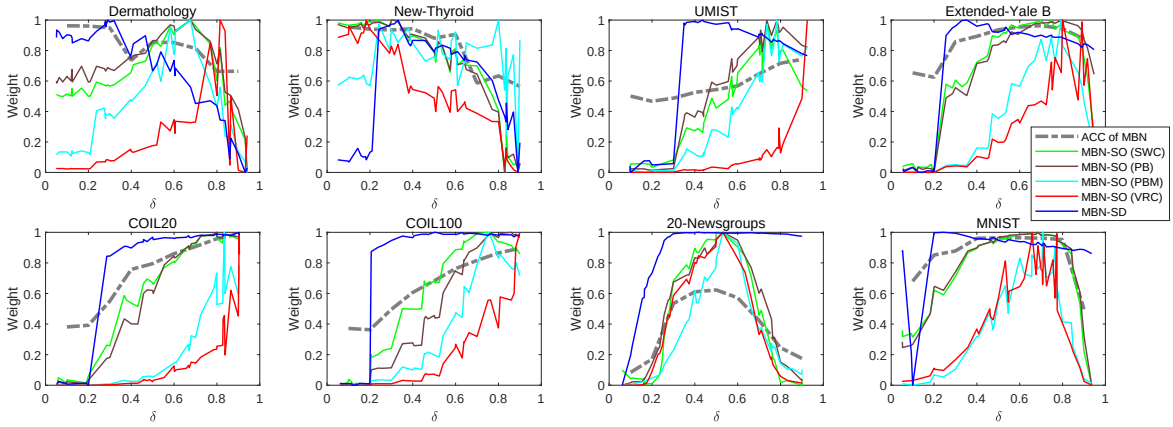

Fig. 3. Weights of the MBN base models of MBN-SO and MBN-SD.

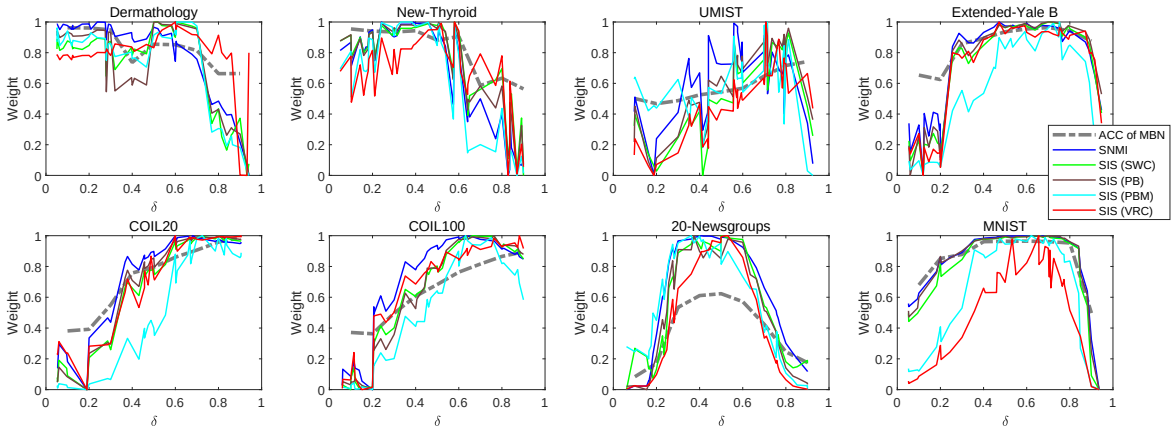

Fig. 4. Weights of the MBN base models of the SNMI and SIS functions.

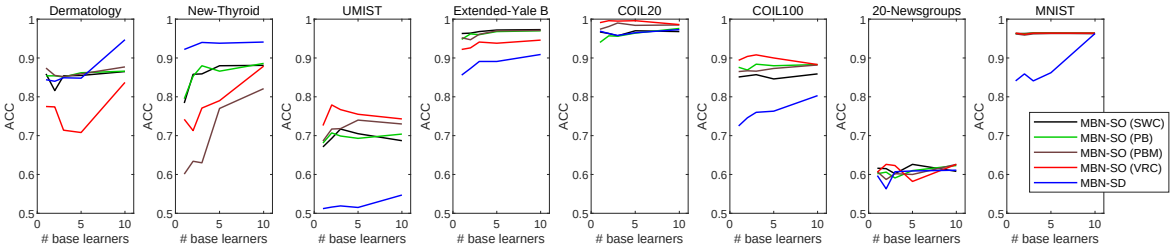

Fig. 5. Effect of the number of the base models of MBN-SO and MBN-SD on performance.

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.

5 Conclusions

In this paper, we have solved the network structure selection problem of MBN by ensemble learning and selection. Specifically, we have first proposed MBN-E, which concatenates the sparse output of a number of MBN base models with different δ to a meta-representation. Then, we take the meta-representation as a guidance to select the optimal base models. We have introduced two unsupervised ensemble selection methods. The first one, named MBN-SO, uses the clustering result of MBN-E to select the base models whose output distributions have the highest discriminability in terms of the optimization-like criteria. The second method, named MBN-SD, uses the meta-representation of MBN-E directly for selecting the optimal base models in terms of distribution divergence criteria. 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; 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 a number of meta-clustering functions and clustering ensemble selection functions.

REFERENCES

[1] X.-L. Zhang, "Multilayer bootstrap networks," *Neural Networks*, vol. 103, pp. 29–43, 2018.

[2] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," *Technical Report CUCS-005-96*, 1996.

[3] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[4] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[5] 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.

[6] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1866–1881, 2005.

[7] 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.

[8] H. Wang, H. Shan, and A. Banerjee, "Bayesian cluster ensembles," in *Proc. 9th SIAM Int. Conf. Data Min.*, 2009, pp. 1–12.

[9] 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.

[10] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Cagliari, Italy: Springer, 2000, pp. 1–15.

[11] S.-o. Abbasi, S. Nejatian, H. Parvin, V. Rezaie, and K. Bagherifard, "Clustering ensemble selection considering quality and diversity," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 1311–1340, 2019.

[12] H. G. Ayad and M. S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 1, pp. 160–173, 2007.

[13] ——, "On voting-based consensus of cluster ensembles," *Pattern Recognition*, vol. 43, no. 5, pp. 1943–1953, 2010.

[14] 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.

[15] 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.

[16] M. Ganaie, M. Hu *et al.*, "Ensemble deep learning: A review," *arXiv preprint arXiv:2104.02395*, 2021.

[17] 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.

[18] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Systems with Applications*, vol. 68, pp. 93–105, 2017.

[19] 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.

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

[21] 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.

[22] J. Azimi and X. Z. Fern, "Adaptive cluster ensemble selection." in *Ijcai*, vol. 9, 2009, pp. 992–997.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] M. Zhang, "Weighted clustering ensemble: A review," *arXiv preprint arXiv:1910.02433*, 2019.

[28] T. Li and C. Ding, "Weighted consensus clustering," in *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, 2008, pp. 798–809.

[29] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[30] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of intelligent information systems*, vol. 17, no. 2, pp. 107–145, 2001.

[31] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin, "Cluster ensemble selection based on a new cluster stability measure," *Intelligent Data Analysis*, vol. 18, no. 3, pp. 389–408, 2014.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[38] 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.

[39] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation." *Journal of Machine Learning Research*, vol. 8, no. 5, 2007.

[40] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[41] M. Baktashmotlagh, M. Harandi, and M. Salzmann, "Distribution-matching embedding for visual domain adaptation," *Journal of Machine Learning Research*, vol. 17, pp. Article–number, 2016.

[42] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and H. T. Shen, "Maximum density divergence for domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[43] J. Maggu, A. Majumdar, E. Chouzenoux, and G. Chierchia, "Deeply transformed subspace clustering," *Signal Processing*, vol. 174, p. 107628, 2020.

[44] J. Zhou, H. Zheng, and L. Pan, "Ensemble clustering based on dense representation," *Neurocomputing*, vol. 357, pp. 66–76, 2019.

[45] X. Sevillano, F. Alías, 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] D. Lim, R. Vidal, and B. D. Haeffele, "Doubly stochastic subspace clustering," *arXiv preprint arXiv:2011.14859*, 2020.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] Y. Ren, N. Wang, M. Li, and Z. Xu, "Deep density-based image clustering," *Knowledge-Based Systems*, vol. 197, p. 105841, 2020.

[60] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[61] A. Villar-Corrales and V. I. Morgenshtern, "Scattering transform based image clustering using projection onto orthogonal complement," *arXiv preprint arXiv:2011.11586*, 2020.

[62] F. Li, H. Qiao, and B. Zhang, "Discriminatively boosted image clustering with fully convolutional auto-encoders," *Pattern Recognition*, vol. 83, pp. 161–173, 2018.

[63] 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.

[64] 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.

[65] 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.

[66] N. Iam-on, S. Garrett *et al.*, "Linkclue: A matlab package for link-based cluster ensembles," *Journal of Statistical Software*, vol. 36, no. 9, pp. 1–36, 2010.

[67] B. G. Mirkin and A. Shestakov, "Least square consensus clustering: criteria, methods, experiments," in *European Conference on Information Retrieval*. Springer, 2013, pp. 764–767.

[68] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," *Journal of Computer and System Sciences*, vol. 61, no. 2, pp. 217–235, 2000.

[69] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithms-for separable nonnegative matrix factorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 4, pp. 698–714, 2013.

[70] N. Gillis, "Successive nonnegative projection algorithm for robust nonnegative blind source separation," *SIAM Journal on Imaging Sciences*, vol. 7, no. 2, pp. 1420–1450, 2014.

[71] A. Kumar, V. Sindhwani, and P. Kambadur, "Fast conical hull algorithms for near-separable non-negative matrix factorization," in *International Conference on Machine Learning*, 2013, pp. 231–239.