Pattern Recognition 156 (2024) 110739

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Robust multilayer bootstrap networks in ensemble for unsupervised representation learning and clustering

Xiao-Lei Zhang [a,b,c,*], Xuelong Li [b]

[a] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China
[b] Institute of Artificial Intelligence (TeleAI), China Telecom, China
[c] Research and Development Institute of Northwestern Polytechnical University in Shenzhen, China

ARTICLE INFO

Keywords:
Ensemble selection
Cluster ensemble
Multilayer bootstrap networks
Unsupervised learning

ABSTRACT

It is known that unsupervised nonlinear learning is sensitive to the selection of hyperparameters, which hinders its practical use. How to determine the optimal hyperparameter setting that may be dramatically different across applications is a hard issue. In this paper, we aim to address this issue for multilayer bootstrap networks (MBN), a recent unsupervised model, in a way as simple as possible. 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. Empirically, comparing to a number of advanced clustering methods, the proposed methods reach reasonable performance in their default settings. fMBN-E is empirically 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.

1. Introduction

Unsupervised learning and clustering is a fundamental task of machine learning. It finds wide applications in data mining, text analysis, etc [1]. Early works, e.g. 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. 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.

This paper aims to address this hard issue for *multilayer bootstrap network* (MBN) [3], given little prior knowledge of data. Specifically, MBN learns data representation in an unsupervised manner by multiple layers of nonlinear transforms. It has a simple formulation, which consists of the components of one-nearest-neighbor optimization, stacking, and random sampling only. It yields good clustering performance if the hyperparameters were set properly, such as a proper number of nonlinear layers and network structures. However, its performance varies dramatically with different network structures and data. Here we show the performance of MBN on COIL20 as an example in 1a . From the figure, we see that, when the hyperparameter δ of MBN ($\delta \in (0,1)$) is set to different values, the low-dimensional feature produced by MBN and its corresponding clustering performance vary dramatically in a large range. See Section 6 for a detailed theoretical analysis on the problem.

To address this issue, ensemble clustering, which groups a set of *base clusterings* effectively, is possibly a simple solution. It has been applied successfully to, e.g. feature selection [4] or hyperparameter tuning [5]. The theoretical base for its success is that, for a binary-class problem, as if the base clusterings of an ensemble clustering are stronger than random guess, then the ensemble clustering may be stronger than any of its base clusterings. It combines the base clusterings with a so-called *meta-clustering function*, a.k.a *consensus function*, for enhancing the stability and accuracy of the base clusterings [6,7]. Meta-clustering functions can be categorized generally to two classes [7]. The first class analyzes and optimizes 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

[*] Corresponding author.
E-mail addresses: xiaolei.zhang@nwpu.edu.cn (X.-L. Zhang), xuelong_li@ieee.org (X. Li).

https://doi.org/10.1016/j.patcog.2024.110739
Received 31 December 2023; Received in revised form 14 June 2024; Accepted 27 June 2024
Available online 2 July 2024

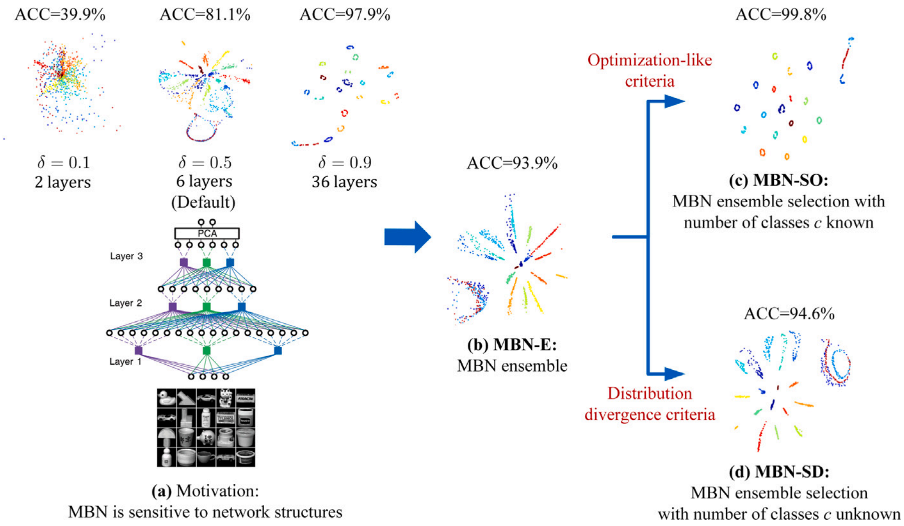

Fig. 1. Connections between MBN and the proposed algorithms. (a) Motivation of MBN. (b) MBN-E. (c) MBN-SO and MBN-SD. Note that each square in MBN in figure (a) represents a base clustering, while the black circles connected to the square represent the input/output of the base clustering. The hyperparameter "δ" controls the network structure of MBN, and $\delta \in (0, 1)$. 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 [21].

[8,9]. It is the recent research focus. For example, in [10], the authors extended random vector functional link networks to an unsupervised ensemble architecture. The authors in [11] proposed self-taught mechanism based on convex combination and centroid graph fusion to learn an optimal consensus similarity graph for graph partitioning. See [7] for a review of the fundamentals of clustering ensemble.

Although ensemble clustering is much more stable than its base clusterings in hyperparameter tuning, some base clusterings may contribute negatively to ensemble learning, which makes the ensemble clustering suboptimal. Therefore, how to identify the strongest base clusterings and group them into a new ensemble, via so called *ensemble reweighting and selection*, needs further investigation. This topic 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. the early work [12] or recent deep learning based methods [13]. A special case of this type is to constrain the weights of some weak base clusterings to zero, named *clustering selection* [14]. However, weak base clusterings may also contain some high quality clusters, and vise versa. With this perspective, many reweighting strategies at levels of clusters [15], data structures, data points [16], and data views [17] were proposed. The algorithms for calculating the weights can be categorized into two types [18]. The first type calculates weights by measuring the similarity between the predicted labels of the clustering ensemble and its base clusterings [12,14]. The second type treats the weights as variables of consensus functions which are obtained by advanced optimization algorithms, e.g. NMF-based optimization in [19] and $\ell_{2,1}$-norm-based sparse optimization in [20]. See [18] for a recent overview of weighted clustering ensemble.

In this paper, we aim to apply ensemble clustering, reweighting and selection to MBN, which produces an off-the-shelf toolbox that can be easily used without heavy human labor. Although many ensemble selection methods may be applied to MBN successfully, we aim to choose, to our knowledge, the simplest and most efficient way, following an expectation that an algorithm obeying the rule of Occam's Razor may be suitable to wide applications. The contributions of this paper are listed as follows:

- We theoretically prove that increasing the depth of MBN (i.e. number of layers) does not always improve the performance, which induces the network structure selection problem of MBN. To address this issue, 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 also theoretically proved that simplifying MBN-E to fMBN-E 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.
- 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 the advantages of the above algorithms, not only over the original MBN but also over a number of advanced clustering algorithms. To demonstrate the potentially wide applications of the proposed algorithms beyond the benchmark datasets, we further applied them successfully to image segmentation and graph data mining.

The connections between MBN and the proposed algorithms can be summarized in Fig. 1, where the performance on the COIL20 data is given as an example.

The rest of the paper is organized as follows. In Section 2, we present related work. In Section 3, we review MBN. In Sections 4 and 5, we present MBN-E, fMBN-E, MBN-SO, and MBN-SD, respectively. In Section 6, we analyze the structure selection problem of MBN both theoretically and empirically, which is the motivation of the proposed algorithms. In Section 7, we present an extensive experiment on clustering. In Section 8, we study the nonlinear representation learning ability empirically on synthetic data. In Section 9, we apply the proposed methods to image segmentation and graph data mining. Finally, in Section 10, we conclude the paper.

2. Related work

In this section, we review ensemble selection criteria. Some of the criteria are applied to the proposed MBN-SO and MBN-SD.

In ensemble cluster reweighting and selection, the criteria for measuring the diversity and quality of the base clusterings of a clustering ensemble lie in the core. They can be categorized into two classes—optimization-based criteria and distribution divergence criteria. Optimization-based criteria calculate the normalized mutual information [12,14], adjusted rand index [22], clustering accuracies [23], and their variants [24] between the sets of the predicted labels of the base clusterings. Distribution divergence criteria is based on the data distributions [25] related to the base clusterings without resorting to the predicted labels. They usually calculate some kinds of statistics of data [26]. Some systematical studies on cluster validation indices [25] have been carried out as well.

The distribution divergence criteria are mostly explored in domain adaptation, instead of clustering ensemble. They define the similarity between the source domain and the target domain of a domain adaptation problem. The most popular measurement is maximum mean discrepancy (MMD) [27]. 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.

Motivated by the above work, we evaluate the quality of the base models by *optimization-like criteria* [25], for MBN-SO, when the number of classes is given; we 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, when the number of classes is not given.

3. Preliminaries

This section presents MBN and its theoretical foundation briefly. See Appendices 1 and 2 of the Supplementary Material for the summary of important notations and detailed description of MBN as well as its geometric and theoretical foundations.

As shown in Fig. 1a, MBN contains multiple layers of nonlinear transforms. 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 relationship:

  $$k_m = \delta k_{m-1} \tag{1}$$

  where k_m and k_{m-1} are the parameter k at the mth 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

  $$k_1 > k_2 > \cdots > k_m > \cdots > k_o \tag{2}$$

  where k_o is the parameter k at the top layer. Note that, the number of layers of MBN M can be automatically calculated by solving $rk_o \geq \delta^{(M-1)}k_1 \geq k_o$, given k_1, k_o, and δ, where the hyperparameter r is manually determined, and usually set to 1.5 for class balanced problems, and set larger for class imbalanced problems.[1]

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

From the above formulation, we can see that the nonlinear property of MBN is implemented by the one-hot encoding in the one nearest neighbor optimization step.

4. Multilayer bootstrap network ensemble (MBN-E)

Because MBN is sensitive to δ (see Section 6 for the detailed description on this problem), 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, which simply concatenates the outputs of a set of MBN into a new representation of data.

Particularly, 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 $\{\mathbf{u}_{z,i}\}_{i=1}^n$. Note that ideally, it is expected that each principal component corresponds to one class in the downstreaming task. In practice, we need slightly more principal components to incorporate more useful information.

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 1. *The computational complexity of MBN-E approximates to $Z(\mathcal{O}(\alpha kVn) + \mathcal{O}(kVn))$ empirically, where $\mathcal{O}(\alpha kVn)$ and $\mathcal{O}(kVn)$ 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.*

[1] We have to guarantee that at least one data point per class is selected to be a center of the k-centroids clusterings at the top layer at a high probability, so as to make sure that the k-centroids clusterings are stronger than random guess, which is the fundamental for the success of ensemble learning. To meet this requirement, we set k_o to $1.5c$ for class-balanced data, and larger for class-imbalanced data.

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: $\{\bar{\mathbf{y}}_i\}_{i=1}^n$
/* train a shared bottom layer */
$\{\mathbf{y}_i\}_{i=1}^n \leftarrow \mathrm{MBN}(\{\mathbf{x}_i\}_{i=1}^n, k_1 - 1, \delta = 0)$
/* train an ensemble of fast MBN */
for $z = 1, \ldots, Z$ do
    $\mathbf{x}_{z,i} \leftarrow \mathbf{y}_i,\ \forall i = 1, \ldots, n$
    $m \leftarrow 2$
    Randomly generate $\delta$ from the range $[0.05, 0.95]$
    while $k_m \geq k_o$ do
        for $v = 1, \ldots, V$ do
            Calculate pairwise similarity matrix $\mathbf{B} = \mathbf{X}_z^T \mathbf{X}_z$ where $\mathbf{X}_z =$
            $[\mathbf{x}_{z,1}, \ldots, \mathbf{x}_{z,n}]$
            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
            for $i = 1, \ldots, n$ do
                Find the largest element of the $i$th row of $\mathbf{B}$, supposed to be the
                $j$th element
                Derive a one-hot code $\mathbf{s}_{i,v} = [s_{i,v,1}, \ldots, s_{i,v,k_m}]^T$ where
                $$s_{i,v,t} = \begin{cases} 1, & \text{if } t = j \\ 0, & \text{otherwise} \end{cases}, \ \forall t = 1, \ldots, k_m$$
            end for
        end for
        $\mathbf{x}_{z,i} \leftarrow [\mathbf{s}_{i,1}^T, \ldots, \mathbf{s}_{i,k_m}^T]^T,\ \forall i = 1, \ldots, n$
        $k_{m+1} \leftarrow \delta k_m$
        $m \leftarrow m + 1$
    end while
    $\bar{\mathbf{y}}_{z,i} \leftarrow \bar{\mathbf{x}}_{z,i},\ \forall i = 1, \ldots, n,\ \forall z = 1, \ldots, Z$
end for
$\bar{\mathbf{y}}_i \leftarrow [\mathbf{y}_{1,i}^T, \mathbf{y}_{2,i}^T, \ldots, \mathbf{y}_{Z,i}^T]^T,\ \forall i = 1, \ldots, n$
```

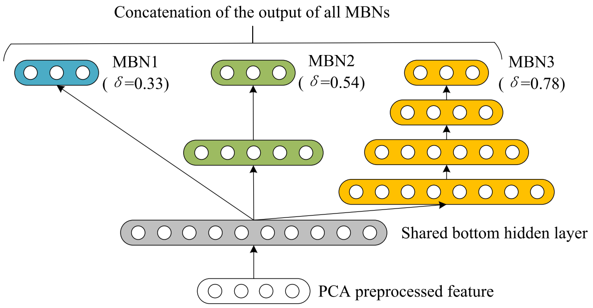
Concatenation of the output of all MBNs
MBN1
(δ=0.33)
MBN2
(δ=0.54)
MBN3
(δ=0.78)
Shared bottom hidden layer
PCA preprocessed feature

Fig. 2. Architecture of fMBN-E. Different color represents different MBN base models with random δ values from a range of $(0, 1)$, e.g., $0.33, 0.54, 0.78$.

4.1. A fast version of MBN-E (fMBN-E)

fMBN-E is described in Algorithm 2. Its architecture is shown in Fig. 2. It accelerates MBN-E by over hundreds of times without performance degradation via the following two aspects:

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

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: $\{\bar{\bar{\mathbf{y}}}_i\}_{i=1}^n$, $\{\bar{\bar{\mathbf{u}}}_i\}_{i=1}^n$.
if $c$ is given then
    $\{l_i\}_{i=1}^n \leftarrow \mathrm{clustering}(\{\bar{\mathbf{u}}_i\}_{i=1}^n, c)$
    for $z = 1$ to $Z$ do
        $\omega_z \leftarrow f_{\text{MBN-SO}}(\{l_i\}_{i=1}^n, \{\mathbf{u}_{z,i}\}_{i=1}^n)$
          (or $\omega_z \leftarrow f_{\text{MBN-SO}}(\{l_i\}_{i=1}^n, \{\mathbf{y}_{z,i}\}_{i=1}^n)$)
    end for
else
    for $z = 1$ to $Z$ do
        $\omega_z \leftarrow f_{\text{MBN-SD}}(\{\bar{\mathbf{y}}_i\}_{i=1}^n, \{\mathbf{y}_{z,i}\}_{i=1}^n)$
          (or $\omega_z \leftarrow f_{\text{MBN-SD}}(\{\bar{\mathbf{u}}_i\}_{i=1}^n, \{\mathbf{u}_{z,i}\}_{i=1}^n)$)
    end for
end if
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
$\bar{\bar{\mathbf{x}}}_i \leftarrow [\mathbf{x}_{1,i}^T, \ldots, \mathbf{x}_{B,i}^T]^T,\ \forall i = 1, \ldots, n$
$\{\bar{\bar{\mathbf{y}}}_i\}_{i=1}^n \leftarrow \mathrm{PCA}(\{\bar{\bar{\mathbf{x}}}_i\}_{i=1}^n)$
```

From the above algorithm, we can easily obtain that:

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

Comparing Theorems 1 and 2, 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 $(\mathcal{O}(\alpha k V n)+\mathcal{O}(k V n))$ [3], we can see that fMBN-E may be even faster than a single MBN described in [3] since that V is larger than Z in practice. Empirically, fMBN-E may be hundreds of times faster than MBN-E.

Moreover, fMBN-E does not degrade the performance of MBN-E both empirically and theoretical analysis. See Appendices for the theoretical proofs.

5. Ensemble selection for MBN-E

In this section, we first present an unsupervised ensemble selection framework for MBN-E in Section 5.1, and then present MBN-SO and MBN-SD in Sections 5.2 and 5.3 respectively.

5.1. Framework

The unsupervised ensemble selection framework contains three steps:

- Step 1: Take the output of MBN-E as a reference representation.
- Step 2: Calculate a score between the reference representation and the representation of each MBN base model in MBN-E via a *criterion*.
- Step 3: Pick the top B MBN base models whose scores are the highest, and concatenate their sparse outputs into a new sparse representation.

Algorithm 3 describes the detail of the framework: 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 zth MBN base model by an 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 $\{\bar{\bar{\mathbf{x}}}_i\}_{i=1}^n$.

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.

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

MBN-SO follows the comparison conclusion on the optimization-like criteria [25], 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 VRC criterion as follows, leaving the other criteria in Appendix 3 of the Supplementary Material.

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

$$\omega^{\text{VRC}} = \frac{1}{h}\frac{n-c}{c-1}\frac{\text{tr}(\mathbf{D})}{\text{tr}(\mathbf{W})} \tag{3}$$

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

$$\mathbf{W} = \sum_{p=1}^{c} \mathbf{W}_p \tag{4}$$

$$\mathbf{W}_p = \sum_{\{\mathbf{u}_i|l_i=p\}} (\mathbf{u}_i - \boldsymbol{\mu}_p)(\mathbf{u}_i - \boldsymbol{\mu}_p)^T \tag{5}$$

$$\mathbf{D} = \sum_{p=1}^{c} n_p(\boldsymbol{\mu}_p - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_p - \bar{\boldsymbol{\mu}})^T \tag{6}$$

where $\bar{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^n \mathbf{u}_i$ is the grand mean of the data, $\boldsymbol{\mu}_p = \frac{1}{n_p}\sum_{\{\mathbf{u}_i|l_i=p\}} \mathbf{u}_i$ is the center of the pth cluster centroid. 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.

5.3. 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.

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 $\{\bar{\mathbf{y}}_i\}_{i=1}^n$ and $\{\mathbf{y}_{z,i}\}_{i=1}^n$. Since $\bar{\mathbf{y}}_i = [\mathbf{y}_{1,i}^T, \ldots, \mathbf{y}_{Z,i}^T]^T$, here we define MMD as follows:

$$\begin{aligned}\upsilon^{\text{MMD}} &= \frac{1}{Z}\frac{1}{n(n-1)}\sum_{i\neq j}\bar{\mathbf{y}}_i^T\bar{\mathbf{y}}_j \\ &+ \frac{1}{n(n-1)}\sum_{i\neq j}\mathbf{y}_{z,i}^T\mathbf{y}_{z,j} - \frac{2}{Z}\frac{1}{n^2}\sum_{u=1}^{Z}\sum_{i,j}\mathbf{y}_{u,i}^T\mathbf{y}_{z,j}\end{aligned} \tag{7}$$

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 $\{\bar{\mathbf{y}}_i\}_{i=1}^n$ and $\{\mathbf{y}_{z,i}\}_{i=1}^n$ are. To make MMD satisfy Algorithm 3, we transform υ^{MMD} by:

$$\omega^{\text{MMD}} = 1 - \frac{\upsilon^{\text{MMD}} - \upsilon_{\min}}{\upsilon_{\max} - \upsilon_{\min}} \tag{8}$$

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

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.

6. Theoretical analysis

It is expected that adding more layers to a multilayer 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 review the estimation error of a single layer of MBN in Section 6.1, which is important for the analysis of the weakness of MBN. Then, we give an empirical demo on how different network structures affect the performance of MBN in Section 6.2. Finally, we derive the estimation error of the entire MBN in Section 6.3 by extending Theorem 3 to the multilayer scenario, which explains the empirical phenomenon theoretically and motivates the novel algorithms of this paper.

6.1. Review: Estimation error of a single layer of MBN

The author in [3] 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 imagine that each k-centroids clustering contributes a nearest neighbor $\mathbf{w}_\upsilon$ to $\mathbf{x}$, $\forall \upsilon = 1, \ldots, V$, 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:

$$\hat{\mathbf{x}} = \frac{1}{V}\sum_{\upsilon=1}^{V}\mathbf{w}_\upsilon \tag{9}$$

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

Under the locally linear assumption, the estimation error $\mathbb{E}(\mathbf{x}-\hat{\mathbf{x}})$ can be decomposed into the following form using the famous *bias–variance decomposition of expectation risk* [28]:

$$\begin{aligned}\mathbb{E}((\mathbf{x}-\hat{\mathbf{x}})^2) &= (\mathbf{x}-\mathbb{E}(\hat{\mathbf{x}}))^2 + \mathbb{E}\left((\mathbf{x}-\mathbb{E}(\hat{\mathbf{x}}))^2\right) \\ &= \text{Bias}^2(\hat{\mathbf{x}}) + \text{Var}(\hat{\mathbf{x}})\end{aligned} \tag{10}$$

Given (10), we can derive the following theorem for the estimation error of a single layer of MBN:

Theorem 3. *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:*

$$\mathbb{E}_{\text{ensemble}} = \left(\frac{1}{V} + \left(1 - \frac{1}{V}\right)\rho\right)\mathbb{E}_{\text{single}} \tag{11}$$

where ρ is the pairwise positive correlation coefficient between the k-centroids clusterings, $0 \le \rho \le 1$ [3].

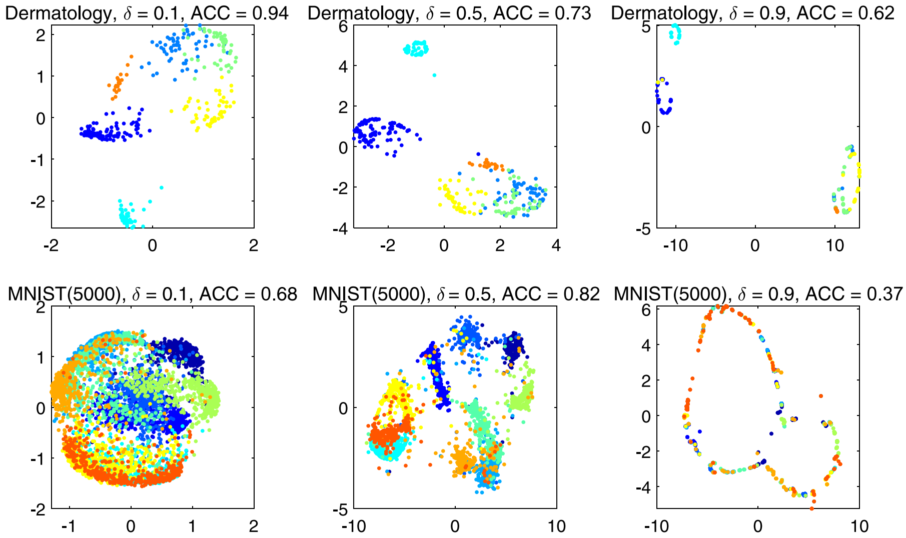

Fig. 3. Visualization of features produced by MBN with different δ on the Dermatology and MNIST(5000) datasets, where Dermatology is a dataset from UCI, and MNIST(5000) is a subset of MNIST dataset that consists of 5000 randomly selected data points.

6.2. Empirical justification of the network structure problem of MBN

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. 3, 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 2 and Fig. 6 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 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.

6.3. Theoretical explanation of the network structure problem of MBN

A fundamental element of MBN is the locally linear assumption defined in (9). 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. 4 explains the empirical phenomenon in Section 6.2.

When we set $\delta \ll \delta_0$, the locally linear assumption (9) may be violated, which makes MBN fail to learn correct representations. For example, in Fig. 4a, 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 4. *When $\delta > \delta_0$, the estimation error of MBN is:*

$$\mathbb{E}_{\text{MBN}} \geq \sum_{m=1}^{M} \left(\frac{1}{V} + \left(1 - \frac{1}{V}\right) \left(\frac{ak_1}{n}\right)^2 \delta^{2(m-1)} \right) \mathbb{E}_{(\text{single},1)} \tag{12}$$

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 (9) does not hold, which makes Theorem 3 do not hold as well. Because the following proof is built on Theorem 3, Theorem 4 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

$$\rho = (ak/n)^2 \tag{13}$$

We denote the correlation at the mth layer as ρ_m. Substituting (1) into (13) derives

$$\rho_m = (ak_{m-1}/n)^2 \delta^2 = \cdots = (ak_1/n)^2 \delta^{2(m-1)} \tag{14}$$

We denote the estimation error of a single k-centroids clustering and an ensemble of clusterings at the mth 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 (14) into (11) derives:

$$\mathbb{E}_{(\text{ensemble},m)} \geq \left(\frac{1}{V} + \left(1 - \frac{1}{V}\right) \left(\frac{ak_1}{n}\right)^2 \delta^{2(m-1)} \right) \mathbb{E}_{(\text{single},1)} \tag{15}$$

Because $\mathbb{E}_{\text{MBN}}$ accumulates $\mathbb{E}_{(\text{ensemble},m)}$ of all layers from bottom-up, we can derive the overall estimation error of MBN as (12). □

We further derive the following corollary from Theorem 4:

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

$$\mathbb{E}_{\text{MBN}} \geq C \sum_{m=1}^{M} \delta^{2(m-1)} \tag{16}$$

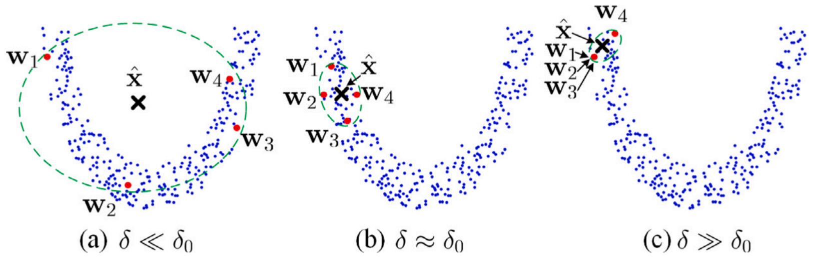

Fig. 4. Diagram of the density estimation process of MBN with different δ. The notation δ_0 denotes the optimal δ. The black cross $\hat{\mathbf{x}}$ denotes the coordinate of the learned representation of an input data $\mathbf{x}$. The four red points, which are $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$ 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.

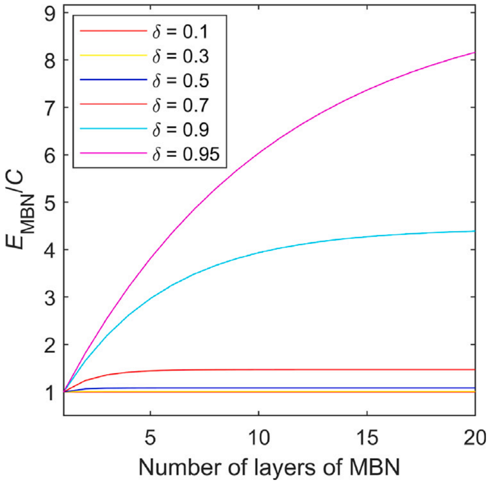

Fig. 5. Connection between the estimation error of MBN and δ when $\delta > \delta_0$, where $C = \left(ak_1/n\right)^2 \mathbb{E}_{(\text{single},1)}$ is a constant.

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

Corollary 1 can be visualized in Fig. 5. From the figure, we see that, when δ approaches to 1, $\mathbb{E}_{\text{MBN}}$ is increased exponentially.

Fig. 4c 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. 4b, 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 MBN to δ is not straightforward, which motivates the proposed methods.

7. Experiments on clustering

In this section, we introduce the experimental settings in Sections 7.1 to 7.3, and compare the proposed methods with a number of representative methods on several benchmark datasets in Section 7.4. Then, we demonstrate how fMBN-E accelerates MBN-E without sacrificing accuracy in Section 7.5, and compare the ensemble selection criteria in Section 7.6. Finally, we present the experimental conclusions of some important aspects in Section 7.7.

7.1. Datasets

We selected 8 benchmark datasets as summarized in Table 2. 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 [39]. 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.

7.2. Parameter settings

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

- **MBN (default)** [3]: We used its default setting as in [3].
- **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.

Table 1
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 the top algorithms on the website “https://paperswithcode.com/”. The number in bold denotes the best performance [29–38].

	Dermatology	New-Thyroid	UMIST*	Extended-Yale B*
kmeans	0.261	0.860	0.408	0.311
Rank1	0.313 (DREC [29])	0.863 (Borda [30])	**0.769 (DASC)**	**0.992 (DMSC)**
Rank2	0.307 (LinkClueE [31])	0.859 (LinkClueE [31])	0.750 (DSC-Net-L2)	0.973 (DSC-Net-L2)
Rank3	0.306 (HGPA [6])	0.853 (ECPCS_MC [32])	0.732 (J-DSSC))	0.924 (J-DSSC))
Rank4	0.299 (CSPA [6])	0.851 (MCLA [6])	0.728 (DSC-Net-L1)	0.917 (A-DSSC)
Rank5	0.297 (ECPCS_HC [32])	0.845 (Vote [33])	0.725 (A-DSSC)	0.776 (SSC-OMP)
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)**	**0.911 (JULE)**	0.600 (LTM [34])	**0.979 (N2D)**
Rank2	0.858 (AGDL)	0.824 (A-DSSC)	0.523 (DFPA [35])	0.969 (DDC-DA)
Rank3	0.858 (GDL)	0.796 (J-DSSC)	0.490 (LDA [36])	0.965 (PSSC)
Rank4	0.793 (DBC)	0.775 (DBC)	0.447 (AnchorFree [37])	0.964 (GDL)
Rank5	N/A	0.731 (GDL)	0.435 (LapPLSI [38])	0.939 (SR-K-means)
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

Table 2
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	32 256	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	18 846	26 214	20	Text	$(0.4, 0.5)$
MNIST	70 000	768	10	Images	$(0.35, 0.75)$

7.3. 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 the self-supervised deep learning based methods listed on the website explore strong hand-crafted image features, such as the random rotations and random color changes that are not suitable to other types of data, from augmented data to obtain implicit supervision information [40], we omit them from the rank lists 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. 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. [41], 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 [42] 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 the one that has the minimum objective value. All reported results are average ones over 5 independent runs.

7.4. General results

Table 1 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†. 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 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. 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

Table 3
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 4
Running time (in seconds) of the bottom layers of MBN(default), MBN-E, fMBN-E, and fMBN-Ev2.

	Dermatology	New-Thyroid	UMIST	Extended-Yale B	COIL20	COIL100	20-Newsgroups	MNIST
MBN(default)	0.53	0.40	0.74	5.62	2.97	78.03	505.82	3347.12
MBN-E	225.08	14.96	118.00	2190.72	834.64	22 148.48	59 997.16	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 5
Running time (in seconds) of the upper layers of MBN(default), MBN-E, fMBN-E, and fMBN-Ev2.

	Dermatology	New-Thyroid	UMIST	Extended-Yale B	COIL20	COIL100	20-Newsgroups	MNIST
MBN(default)	1.07	0.89	1.30	6.70	4.17	53.79	714.58	7852.32
MBN-E	293.85	165.15	508.75	1829.94	1413.17	5617.11	26 002.17	63939.58
fMBN-E	3.02	1.63	3.38	31.85	20.05	206.46	2085.35	9108.11
fMBN-Ev2	1.95	1.34	2.37	21.52	10.17	103.35	1141.76	8638.58

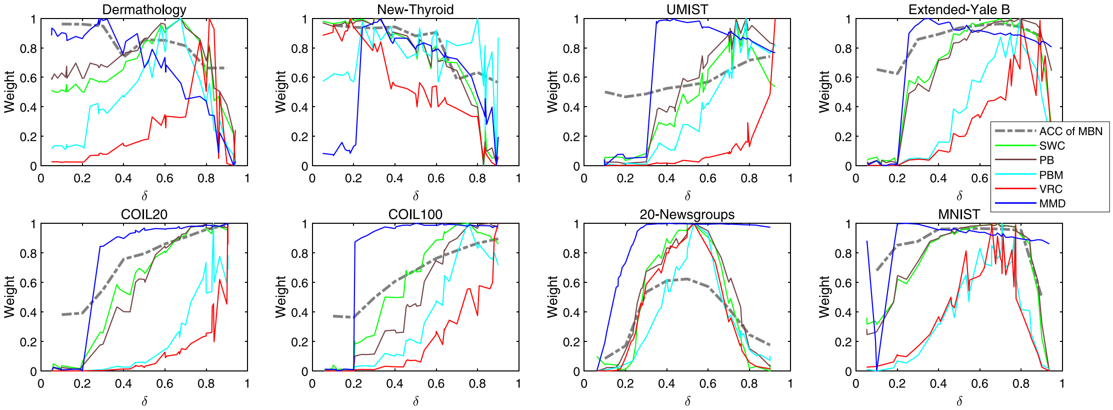

Fig. 6. Weights of the MBN base models produced by different ensemble selection criteria, where SWC, PB, PBM and VRC are optimization-like criteria for MBN-SO, and MMD is a distribution divergence criterion for MBN-SD. The dotted lines in gray color are the accuracies of MBN with respect to δ, which are references for evaluating the effectiveness of the weights.

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 criteria—into clustering ensemble, which may motivate new criteria beyond MMD for further improving the performance of MBN-SD.

7.5. Comparison between MBN-E and fMBN-E

Table 3 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 4 and 5 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. Their computational complexities are similar with the complexity of MBN(default). The phenomenon supports the theoretical analysis of Theorem 2.

7.6. 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. 6. 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.

7.7. Discussions

This subsection reports the main conclusions of some important aspects, leaving the detailed description of the experiments in Appendix 4 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 needs 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.

8. Experiments on unsupervised representation learning

The success of the proposed methods lies in that they project a nonlinear and nonuniform distribution into a linearly separable and uniform distribution, so that a very simple linear classifier can yield a high classification accuracy.

To support the above claim, in this section, we conducted a controllable experiment on a nonlinear and nonuniform "two-moon" data as shown in the "original data" of Fig. 7. We compared the proposed methods with four well known unsupervised nonlinear learning algorithms, i.e. isometric mapping (Isomap) [43], locally linear embedding (LLE) [44], spectral clustering [2], and t-distributed stochastic neighbor embedding (t-SNE) [45].

Fig. 7 shows the two dimensional embedding features learned by the comparison methods. From the figure, we see that the density of the two classes produced by the proposed methods is similar, while the comparison methods fail to do so. Moreover, MBN-SO(VRC) and MBN-SD yield more compact representations than MBN(default) and fMBN-E.

Fig. 8 shows the clustering results on the produced embedding features. From the figure, we see that MBN-SO(VRC) produces the best clustering result, followed by fMBN-E.

9. Applications

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

Table 6
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

9.1. 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 datasets[2] 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. 9, while more examples are listed in Appendix 5 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.

9.2. 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 6. 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 [46]. Because the modularity can be calculated in an unsupervised manner by comparing $\mathbf{B}$ with the

[2] https://cocodataset.org

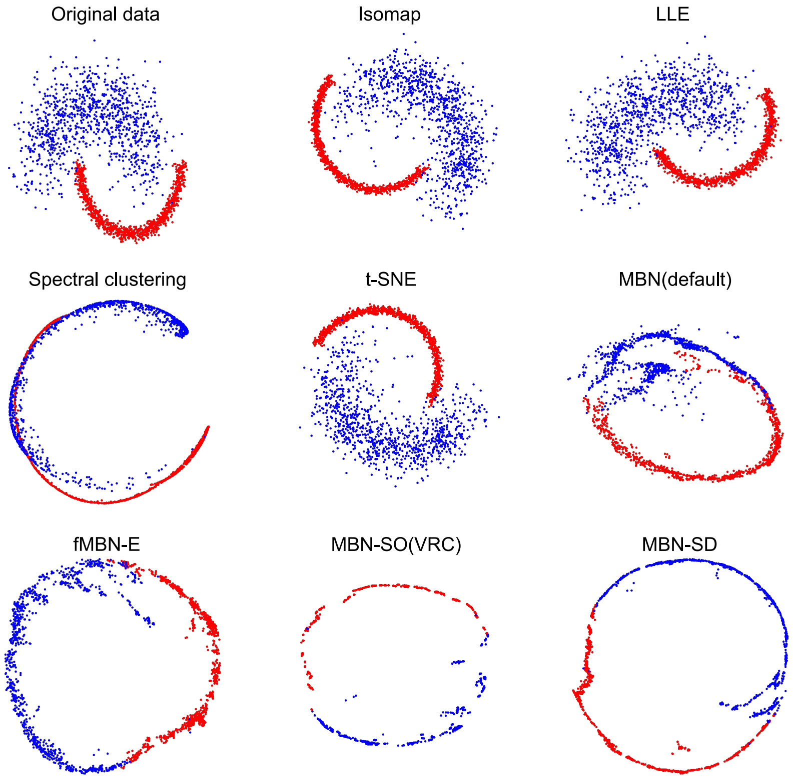

Fig. 7. Visualizations of the two-dimensional embedding features learned by the comparison methods on the two-moon synthetic data.

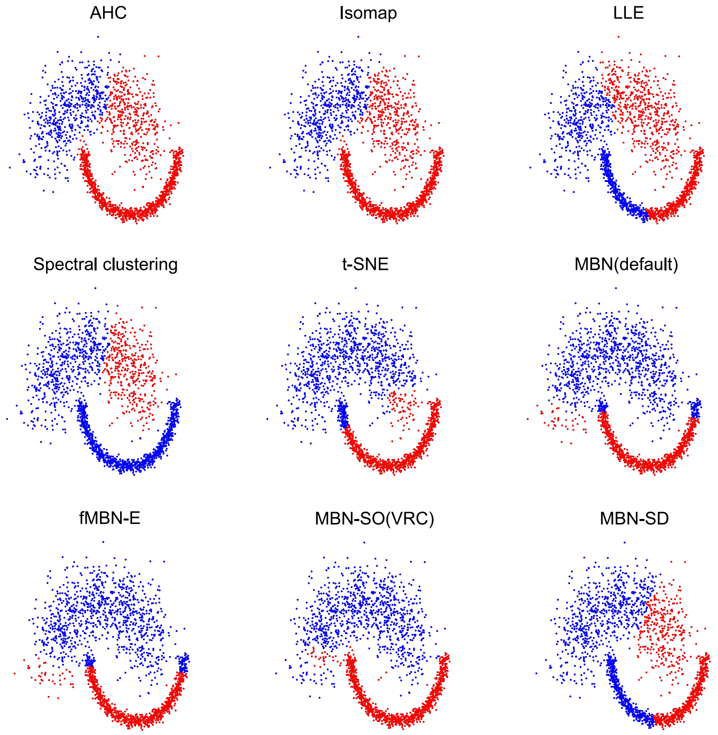

Fig. 8. Clustering results on the produced embedding features of the comparison methods.

prediction result, we are able to search for the optimal modularity results as [46]. Specifically, we searched the parameter k_o of fMBN-E from $\{15, 30, 45, 60\}$ 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 [46], we reported the average results over 5 independent runs. Table 7 lists the comparison results with four well-known community detection algorithms [47–50]. 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

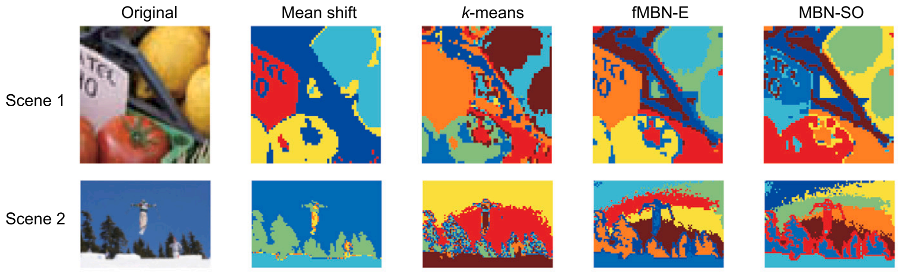

Fig. 9. Results of the image segmentation methods on 2 randomly selected examples from the 2017 Val images of the COCO datasets.

Table 7
Modularity of the community detection algorithms on the GEMSEC-facebook datasets. Note that, because our experimental settings on the datasets are exactly the same with those in [46](Table 3), we directly copied the results of the referenced methods [47–50] from [46](Table 3) for a fair comparison.

	Politicians	Companies	Athletes	News sites	Public figures	Artists	Government	TV shows	Ranking
Overlap factorization [47]	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 [48]	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 [49]	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 [50]	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**

the walktrap algorithm [48]. Note that because MBN-SD yields almost identical performance with fMBN-E, we omit its result here.

10. Conclusions

In this paper, we proposed simple and tuning-free unsupervised learning methods based on MBN by ensemble learning and selection. Specifically, we first proposed MBN-E which simply concatenates the sparse output of a number of MBN base models with different δ to a meta-representation. Then, we proposed MBN-SO and MBN-SD which use the output of MBN-E to select the base models whose output distributions have the highest discriminability. 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 clustering toolbox. Experimental comparison results on a wide variety of benchmark datasets show that the proposed methods reach good performance in data clustering, image segmentation, and graph data mining. Further analysis show the effectiveness of each proposed method.

The main advantages of the proposed methods may be as follows. (i) As non-neural-network methods in the deep learning era, they yield good performance that is comparable to advanced neural network methods with the default setting in the investigated cases, which fascinates their practical use. (ii) They are mathematically simple, efficient, and support parallel computing naturally. (iii) Their advanced performance can be proved theoretically from classic frequentist statistics, e.g. the bias–variance decomposition of expectation risk, which may enrich the unsupervised ensemble learning theory.

One weakness is that the performance improvement of the proposed methods in the self-supervised learning scenario is limited compared to the neural-network-based methods, where self-supervised learning means that one can extract supervised labels implicitly from the sample-level inner structures of some kinds of data, e.g. images, by random transformations. This phenomenon is mainly caused by that the proposed methods cannot handle supervised information of data effectively, which limits them to conventional unsupervised learning setting where the data samples are regarded simply as feature vectors. Fortunately, we have observed some positive phenomenon when using augmented data instead of the original data. How to handle very large-scale augmented data efficiently still needs to be investigated, given that the current methods use data resampling to learn the neighboring relationship between data points which is computationally expensive.

In the future, we will also investigate how to add supervised information of data into the model training of the proposed methods, and then generalize the proposed methods to supervised learning, semi-supervised learning, self-supervised learning, etc.

CRediT authorship contribution statement

Xiao-Lei Zhang: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Xuelong Li:** Writing – review & editing, Resources, Methodology, Formal analysis.

Declaration of competing interest

We declare that there is no conflict of interests.

Data availability

The data that has been used is confidential.

Acknowledgments

This work was supported in part by the National Science Foundation of China (NSFC) under Grant 62176211, and in part by the Project of the Science, Technology, and Innovation Commission of Shenzhen Municipality, China under Grant JCYJ20210324143006016 and JSGG20210802152546026.

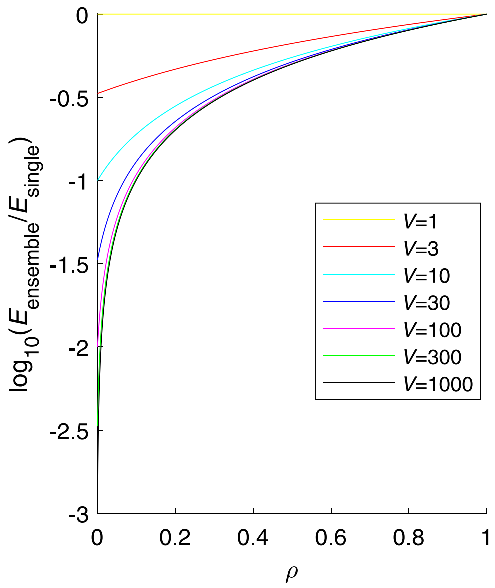

Fig. A.10. Relationship between the estimation error $\mathbb{E}_{\text{ensemble}}/\mathbb{E}_{\text{single}}$, correlation coefficient ρ, and number of k-centroids clusterings per layer V.

Appendix A. On the first novel aspect of fMBN-E

Based on Theorem 3, we can draw the connections between $\mathbb{E}_{\text{ensemble}}/\mathbb{E}_{\text{single}}$, ρ, and V in Fig. A.10, and further derive the following corollary from (11).

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:*

$$\frac{\mathbb{E}_{\text{fMBN-E}}}{\mathbb{E}_{\text{MBN-E}}} = \frac{\left(\frac{1}{V} + \left(1 - \frac{1}{V}\right)\rho\right)\mathbb{E}_{\text{single}}}{\left(\frac{1}{ZV} + \left(1 - \frac{1}{ZV}\right)\rho\right)\mathbb{E}_{\text{single}}} = \frac{Z + (ZV - Z)\rho}{1 + (ZV - 1)\rho} \tag{A.1}$$

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:*

$$\mathbb{E}_{\text{fMBN-E}} \approx \mathbb{E}_{\text{MBN-E}} \tag{A.2}$$

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 ρ. For the first case when $\rho \to 0$, $\mathbb{E}_{\text{fMBN-E}} \approx Z\mathbb{E}_{\text{MBN-E}}$; 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 (13), we also observe that, when k is enlarged, ρ is enlarged as well. According to Theorem 3, 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. □

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

Appendix B. On the second novel aspect of fMBN-E

This subsection explains why fMBN-E is able to discard the random feature 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 (13) if we remove the random feature selection step by setting $a = 1$, we may have $\rho \approx 1/2^6$. From Fig. A.10, 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 $\mathbb{E}_{\text{ensemble}}$. □

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.

References

[1] L. Bai, J. Liang, A categorical data clustering framework on graph representation, Pattern Recognit. 128 (2022) 108694.
[2] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, Adv. Neural Inf. Process. Syst. 14 (2001).
[3] X.-L. Zhang, Multilayer bootstrap networks, Neural Netw. 103 (2018) 29–43.
[4] T. Wu, Y. Hao, B. Yang, L. Peng, ECM-EFS: An ensemble feature selection based on enhanced co-association matrix, Pattern Recognit. 139 (2023) 109449.
[5] T. Qiu, Y. Li, Enhancing in-tree-based clustering via distance ensemble and kernelization, Pattern Recognit. 112 (2021) 107731.
[6] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2003) 583–617.
[7] S. Vega-Pons, J. Ruiz-Shulcloper, A survey of clustering ensemble algorithms, Int. J. Pattern Recognit. Artif. Intell. 25 (03) (2011) 337–372.
[8] T. Li, C. Ding, 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.
[9] N. Nguyen, R. Caruana, Consensus clusterings, in: Seventh IEEE International Conference on Data Mining, ICDM 2007, IEEE, 2007, pp. 607–612.
[10] M. Hu, P.N. Suganthan, Representation learning using deep random vector functional link networks for clustering, Pattern Recognit. 129 (2022) 108744.
[11] G. Zhong, C.-M. Pun, Self-taught multi-view spectral clustering, Pattern Recognit. 138 (2023) 109349.
[12] Z.-H. Zhou, W. Tang, Clusterer ensemble, Knowl.-Based Syst. 19 (1) (2006) 77–83.
[13] M. Zhao, W. Yang, F. Nie, Deep multi-view spectral clustering via ensemble, Pattern Recognit. 144 (2023) 109836.
[14] X.Z. Fern, W. Lin, Cluster ensemble selection, Stat. Anal. Data Min. ASA Data Sci. J. 1 (3) (2008) 128–141.
[15] Q. Huang, R. Gao, H. Akhavan, An ensemble hierarchical clustering algorithm based on merits at cluster and partition levels, Pattern Recognit. 136 (2023) 109255.
[16] F. Li, Y. Qian, J. Wang, C. Dang, L. Jing, Clustering ensemble based on sample's stability, Artificial Intelligence 273 (2019) 37–55.
[17] D. Huang, C.-D. Wang, J.-H. Lai, Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity, IEEE Trans. Knowl. Data Eng. (2023) 1–16.
[18] M. Zhang, Weighted clustering ensemble: A review, Pattern Recognit. 124 (2022) 108428.
[19] T. Li, C. Ding, Weighted consensus clustering, in: Proceedings of the 2008 SIAM International Conference on Data Mining, SIAM, 2008, pp. 798–809.
[20] Z. Wang, S. Zhao, Z. Li, H. Chen, C. Li, Y. Shen, Ensemble selection with joint spectral clustering and structural sparsity, Pattern Recognit. 119 (2021) 108061.
[21] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (coil-20), Technical Report CUCS-005-96, 1996.
[22] J. Jia, X. Xiao, B. Liu, L. Jiao, Bagging-based spectral clustering ensemble selection, Pattern Recognit. Lett. 32 (10) (2011) 1456–1467.
[23] Y. Hong, S. Kwong, H. Wang, Q. Ren, Resampling-based selective clustering ensembles, Pattern Recognit. Lett. 30 (3) (2009) 298–305.
[24] D. Huang, C.-D. Wang, J.-H. Lai, C.-K. Kwoh, Toward multidiversified ensemble clustering of high-dimensional data: From subspaces to metrics and beyond, IEEE Trans. Cybern. 52 (11) (2021) 12231–12244.

[25] L. Vendramin, R.J.G.B. Campello, E.R. Hruschka, Relative clustering validity criteria: A comparative overview, Stat. Anal. Data Min. ASA Data Sci. J. 3 (4) (2010) 209–235.
[26] M.C. Naldi, A. Carvalho, R.J.G.B. Campello, Cluster ensemble selection based on relative validity indexes, Data Min. Knowl. Discov. 27 (2) (2013) 259–289.
[27] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.-P. Kriegel, B. Schölkopf, A.J. Smola, Integrating structured biological data by kernel maximum mean discrepancy, Bioinformatics 22 (14) (2006) e49–e57.
[28] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2009.
[29] J. Zhou, H. Zheng, L. Pan, Ensemble clustering based on dense representation, Neurocomputing 357 (2019) 66–76.
[30] X. Sevillano, F. Alías, 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.
[31] N. Iam-On, T. Boongoen, S. Garrett, C. Price, A link-based approach to the cluster ensemble problem, IEEE Trans. Pattern Anal. Mach. Intell. 33 (12) (2011) 2396–2409.
[32] D. Huang, C.-D. Wang, H. Peng, J. Lai, C.-K. Kwoh, Enhanced ensemble clustering via fast propagation of cluster-wise similarities, IEEE Trans. Syst. Man Cybern. Syst. 51 (1) (2018) 508–520.
[33] E. Dimitriadou, A. Weingessel, K. Hornik, A combination scheme for fuzzy clustering, Int. J. Pattern Recognit. Artif. Intell. 16 (07) (2002) 901–912.
[34] D. Cai, X. Wang, 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.
[35] R. Henao, Z. Gan, J. Lu, L. Carin, Deep Poisson factor modeling, Adv. Neural Inf. Process. Syst. 28 (2015) 2800–2808.
[36] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[37] X. Fu, K. Huang, N.D. Sidiropoulos, Q. Shi, M. Hong, Anchor-free correlated topic modeling, IEEE Trans. Pattern Anal. Mach. Intell. 41 (5) (2018) 1056–1071.
[38] D. Cai, Q. Mei, J. Han, C. Zhai, Modeling hidden topics on document manifold, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, 2008, pp. 911–920.
[39] J. Maggu, A. Majumdar, E. Chouzenoux, G. Chierchia, Deeply transformed subspace clustering, Signal Process. 174 (2020) 107628.
[40] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.
[41] A.L.N. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 835–850.
[42] J. Wang, X.-L. Zhang, Deep NMF topic modeling, Neurocomputing 515 (2023) 157–173.
[43] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[44] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[45] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).
[46] B. Rozemberczki, R. Davies, R. Sarkar, 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, URL https://arxiv.org/pdf/1802.03997v1.pdf.
[47] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, A.J. Smola, Distributed large-scale natural graph factorization, in: Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 37–48.
[48] P. Pons, M. Latapy, Computing communities in large networks using random walks, in: International Symposium on Computer and Information Sciences, Springer, 2005, pp. 284–293.
[49] A. Clauset, M.E.J. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (6) (2004) 066111.
[50] S. Gregory, Finding overlapping communities in networks by label propagation, New J. Phys. 12 (10) (2010) 103018.

Xiao-Lei Zhang received the Ph.D. degree in information and communication engineering from Tsinghua University, Beijing, China, in 2012. He is currently a Professor with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China. He was a Postdoctoral Researcher with Perception and Neurodynamics Laboratory, The Ohio State University, Columbus, OH, USA. His research interests include machine learning, statistical signal processing, and artificial intelligence.

Xuelong Li is the CTO and Chief Scientist of China Telecom, and he founded China Telecom Institute of Artificial Intelligence (TeleAI).