Supplementary Materials To: "Robust Multilayer Bootstrap Networks in Ensemble for Unsupervised Learning and Clustering"

April 10, 2024

Abstract

The content of this supplementary material is listed as follows:
Appendix 1. Important notations of the main text and the appendix.
Appendix 2. Detailed description of MBN and its geometric and theoretical foundations.
Appendix 3. Description of the ensemble selection criteria of MBN-SO and MBN-SD.
Appendix 4. Discussions of some important aspects, including:
4.1 Effect of number of selected base models on MBN-SO and MBN-SD.
4.2 Effect of the referenced labels on MBN-SO.
4.3 On candidate meta-clustering functions of MBN-E.
4.4 On candidate ensemble selection methods of MBN-SO
Appendix 5. Application to image segmentation

1 Important notations

Table 1: Important notations of the main text and appendices.

Notation	Description
$\{\mathbf{x}_i\}_{i=1}^n$	h-dimensional input dataset with n data points
c	Number of classes of the input data
k	Number of centroids per k-centroids clustering in MBN. This is a general description
k_m	Parameter k at the mth layer of MBN, where $m = 1, 2, \ldots$
k_1	Parameter k at the bottom layer of MBN (i.e. k_m with $m = 1$)
k_o	Parameter k at the top layer of MBN
V	Number of k-centroids clusterings of MBN per layer
δ	Core parameter that controls the network structure of MBN. It is defined as $k_{m+1} = \delta k_m$. $\delta \in (0, 1)$
a	Percentage of randomly selected features over all features of the input of a layer. $a \in (0, 1]$
$\{\mathbf{y}_i\}_{i=1}^n$	Sparse output representation produced by MBN
$\{\mathbf{u}_i\}_{i=1}^n$	Low dimensional representation of $\{\mathbf{y}_i\}_{i=1}^n$ made by PCA
Z	Number of MBN base models in MBN-E or fMBN-E
$\{\mathbf{y}_{z,i}\}_{i=1}^n$	Sparse output representation produced by the zth MBN base model of MBN-E or fMBN-E
$\{\mathbf{u}_{z,i}\}_{i=1}^n$	Low dimensional representation of $\{\mathbf{y}_{z,i}\}_{i=1}^n$ made by PCA
$\{\bar{\mathbf{y}}_i\}_{i=1}^n$	Sparse output representation produced by MBN-E or fMBN-E. $\bar{\mathbf{y}}_i = [\mathbf{y}_{1,i}^T, \ldots, \mathbf{y}_{Z,i}^T]^T$
$\{\bar{\mathbf{u}}_i\}_{i=1}^n$	Low dimensional representation of $\{\bar{\mathbf{y}}_i\}_{i=1}^n$ made by PCA
B	Number of selected MBN base models by MBN-SO or MBN-SD
$\{\bar{\bar{\mathbf{y}}}_i\}_{i=1}^n$	Sparse output representation produced by MBN-SO or MBN-SD
$\{\bar{\bar{\mathbf{u}}}_i\}_{i=1}^n$	Low dimensional representation of $\{\bar{\bar{\mathbf{y}}}_i\}_{i=1}^n$ made by PCA

2 MBN and its theoretical foundations

This section first reviews MBN in Appendix 2.1, then reviews its geometric and theoretical principles in Appendices 2.2 and 2.3 respectively, and finally reviews its computational complexity in Appendix 2.4.

2.1 MBN

2.1.1 Network structure of MBN

MBN is a multilayer nonlinear network [1]. Its network structure is shown in Fig. 1. Specifically, each layer of MBN is a clustering ensemble, which consists of V mutually-independent k-centroids clusterings. Each k-centroids clustering takes the output of its lower layer as its input, and partitions the input data into k clusters, which yields a one-hot sparse representation for each input data point. The outputs of all clusterings in the same layer are concatenated as the input of their upper layer.

The network structure of MBN is determined by the parameter k. Suppose the parameter k at the m-th layer is k_m. Then, we must have

$$k_1 > k_2 > \ldots > k_m > \ldots > k_o \tag{1}$$

where k_o is the parameter k at the top layer. The above inequality is usually controlled simply by:

$$k_{m+1} = \delta k_m \tag{2}$$

where $\delta \in (0, 1)$ is a tunable hyperparameter. Note that the total number of layers of MBN is usually determined automatically by k_1, k_o, and δ.

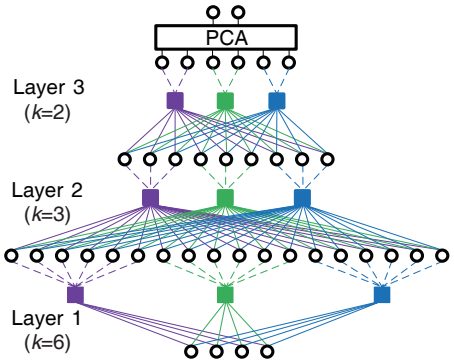


Figure 1: Network structure of MBN. The dimension of the input data for this demo network is 4. Each colored square represents a k-centroids clustering. Each layer contains 3 clusterings. Parameters k at layers 1, 2, and 3 are set to 6, 3, and 2 respectively. The outputs of all clusterings in a layer are concatenated as the input of their upper layer [1].

2.1.2 Training process of MBN

Given an unlabeled dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ that consists of c classes of data, the detailed training process of MBN is summarized in **Algorithm 1**.

The following two issues of Algorithm 1 need to be further clarified.

- The similarity measurement between the centroid $\mathbf{w}_j$ and an input data point $\mathbf{x}_i$ is customized at the bottom layer, and predefined as $\mathbf{w}_j^T\mathbf{x}_i$ at the other layers,
- The parameter k_o should be set to guarantee that at least one data point per class is randomly selected in probability when building the k-centroids clusterings at the top layer, therefore it is set to $k_o = \lfloor 1.5c \rfloor$ for class-balanced data, and set larger for class-imbalanced data, e.g. $k_o = 5c$ or $k_o = 10c$. For clarity, the notations of the important variables of the paper are summarized in Table 1.

2.2 Review of the geometric principle of MBN

The principle for the success of MBN is as follows [1]:

Theorem 1. *MBN builds as many as $O(k_o 2^V)$ agglomerative hierarchical trees on the original data space. The leaf nodes of the trees represent the local areas of the original data space, which are as many as $O(k_1 2^V)$.*

MBN learns a sparse representation for each input data point. The sparse representation encodes the root node where the data point locates at.

To understand Theorem 1, we draw an example in Fig. 2. Specifically, we first imagine that a single k-centroids clustering partitions the input space to k disconnected fractions. Thereafter, V clusterings partition the input space to $O(k2^V)$ fractions at the maximum. Given parameters $k_1 > k_2 >, \ldots, > k_o$, it is easy to see that $O(k_1 2^V) > O(k_2 2^V) >, \ldots, > O(k_o 2^V)$. As a result, between any two adjacent layers, there must be $O(k_{m-1}2^V) - O(k_m 2^V)$ nodes at the $(m-1)$-th layer absorbed into other nodes, which builds tree structures.

Note that, although we draw multiple data points in a single local area in Fig. 2, this situation is unlikely to happen, since that the number of root nodes which is as high as $O(k_o 2^V)$ is usually exponentially larger than the number of data points. That is to say, it is unlikely that MBN makes two data points share the same sparse representation.

From this geometric view, we see that a single root node may represent a large and nonlinear area in the original data space. That is to say, MBN may be able to transform a nonlinear and non-uniform distribution into a linearly-separable and uniform distribution.

2.3 Review of the theoretical foundation of MBN

MBN is theoretically rooted at the famous *bias-variance decomposition of expectation risk* which is the foundation of ensemble learning [2]:

Algorithm 1 MBN.

Input: A h-dimensional unlabeled dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, parameter k_o, and network structure controller δ

Initialization: $m = 1$, $k_1 = \lfloor n/2 \rfloor$, number of base clusterings per layer $V = 400$, percentage of the randomly selected features over all features $a = 0.5$

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

while $k_m > k_o$ **do**
  for $v = 1, \ldots, V$ **do**
    Randomly select $\lfloor ah \rfloor$ dimensions of $\mathcal{X}$ to form a new dataset $\mathcal{E} = \{\mathbf{e}_i\}_{i=1}^n$
    Randomly select k_m data points from $\mathcal{E}$ as the centroids of the v-th clustering at the m-th layer, denoted as $\{\mathbf{w}_j\}_{j=1}^{k_m}$
    for $i = 1, \ldots, n$ **do**
      Find the closest centroid to the data point $\mathbf{e}_i$, supposed to be $\mathbf{w}_j$
      Derive a one-hot code $\mathbf{s}_{i,v} = [s_{i,v,1}, \ldots, s_{i,v,k_m}]^T$ where

$$s_{i,v,t} = \begin{cases} 1, \text{ if } t = j \\ 0, \text{ otherwise} \end{cases}, \ \forall t = 1, \ldots, k_m \tag{3}$$

    end for
  end for
  for $i = 1, \ldots, n$ **do**
    $\mathbf{x}_i \leftarrow [\mathbf{s}_{i,1}^T, \ldots, \mathbf{s}_{i,k_m}^T]^T$
  end for
  $h \leftarrow k_m V$
  $k_{m+1} \leftarrow \delta k_m$
  $m \leftarrow m + 1$
end while
$\mathbf{y}_i \leftarrow \mathbf{x}_i, \quad \forall i = 1, \ldots, n$

Theorem 2. *(Bias-variance decomposition of expectation risk) Suppose the ground-truth prediction is x, and the estimated prediction is $\hat{x}$, then the bias-variance decomposition of the expectation risk $\mathbb{E}(x - \hat{x})$ is:*

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

From Theorem 2, we can derive the following theorem for 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:*

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

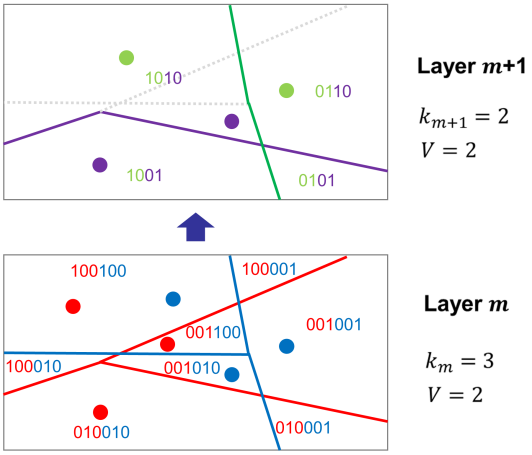


Figure 2: Geometric principle of MBN. Both of the two rectangles that contain dots, lines, and digit codes represent the same original data space. The dots in the same color represent the centroids of a k-centroids clustering, which are randomly sampled data points. The solid lines in the same color are the borders made by a k-centroids clustering, which partition the data space into local areas. The digit codes encode the local areas of the input data. The data point that falls into a local area takes the digit code of the area as its representation learned by MBN. In this example, 2 clusterings at the mth and $(m+1)$th layers partition the input data space into 8 and 4 fractions respectively, where the number of fractions $O(k2^V)$ in the theoretical analysis should be 12 and 8 respectively at the maximum.

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

To understand Theorem 3, we focus on the representation learning problem of a single input data point $\mathbf{x}$ at a layer of MBN. Each of the V k-centroids clustering at the layer estimates $\mathbf{x}$ as the nearest centroid to $\mathbf{x}$ among the k centroids, supposed to be $\mathbf{w}_v$, $\forall v = 1, \ldots, V$. We assume that the elements in the vector $\mathbf{w}_v$ are mutually-independent, and each element, denoted as w_v, follows a Gaussian distribution assumption. With the assumptions, we can have

$$\mathbb{E}(w_v) = \mu, \ \mathbb{E}(w_v^2) = \sigma^2, \ \mathbb{E}(w_{v_1} w_{v_2}) = \rho\sigma^2 + \mu^2 \tag{6}$$

where μ and σ are the mean and variance of the Gaussian distribution respectively, and v_1 and v_2 are the indices of two k-centroids clusterings.

For a single k-centroids clustering, we have $\hat{x} = w_v$. Given (6), we can further derive $\text{Bias}^2_{\text{single}}(\hat{x}) = 0$ and $\text{Var}_{\text{single}}(\hat{x}) = \sigma^2$. For an ensemble of k-centroids

clusterings, with a *locally linear assumption*, we have

$$\hat{x} = \frac{1}{V}\sum_{v=1}^{V} w_v \tag{7}$$

and can further derive $\text{Bias}^2_{\text{ensemble}}(\hat{x}) = 0$ and $\text{Var}_{\text{ensemble}}(\hat{x}) = \frac{\sigma^2}{V} + (1 - \frac{1}{V})\rho\sigma^2$. Finally, substituting the above derivation into (4), we can derive Theorem 3.

From Theorem 3, we can further derive the following two corollaries:

Corollary 1. *When ρ approaches to 0, then $\mathbb{E}_{\text{ensemble}}$ approaches to a lower-bound $\mathbb{E}_{\text{single}}/V$.*

Corollary 2. *When ρ approaches to 1, then $\mathbb{E}_{\text{ensemble}}$ approaches to an upper-bound $\mathbb{E}_{\text{single}}$.*

2.4 Review of the computational complexity of MBN

Theorem 4. *The computational complexity of MBN approximates to $\mathcal{O}(\alpha kVn) + \mathcal{O}(kVn)$ empirically, where $\mathcal{O}(\alpha kVn)$ and $\mathcal{O}(kVn)$ are the complexity of MBN at the bottom layer and the other layers respectively, and α is a constant that is related to the sparse property of the input data [1].*

Theoretically, according to Algorithm 1, each layer of MBN calculates the pairwise similarity of data by V times. Because the complexity of computing the pairwise similarity of data is $\mathcal{O}(n^2)$, it seems that the computational complexity of MBN should be $\mathcal{O}(Vn^2)$. However, in practice, because the learned sparse representation for any input data contains only k nonzero elements, the calculation of the pairwise similarity is only $\mathcal{O}(kVn)$ empirically. A special case is that the sparsity of the input of MBN at the bottom layer is related to the sparse property of the original data, which accounts for the empirical complexity of MBN at the bottom layer.

3 Ensemble selection criteria of MBN-SO and MBN-SD

In this section, we present the four model selection criteria of MBN-SO in Appendices 3.1 to 3.4, and the MMD selection criterion of MBN-SD in Appendix 3.5.

3.1 Silhouette width criterion

SWC calculates the ratio of the geometric compactness and separation of clusters. Suppose the i-th data point $\mathbf{u}_i$ belongs to a cluster $p \in \{1, \ldots, c\}$. Let the average distance of $\mathbf{u}_i$ to all other data points in cluster p be denoted by a_i. Let the average distance of $\mathbf{u}_i$ 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:

$$d_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \tag{8}$$

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

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

$$\omega^{\mathrm{SWC}} = \frac{1}{n} \sum_{i=1}^{n} d_i \tag{9}$$

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

3.2 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:

$$d_w = \frac{1}{n} \sum_{i=1}^{n} a_i \tag{10}$$

$$d_b = \frac{1}{n} \sum_{i=1}^{n} \sum_{\{q|q=1,\ldots,c,q\neq p\}} \frac{n_q}{n - n_p} g_{q,i} \tag{11}$$

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:

$$\omega^{\mathrm{PB}} = \frac{(d_b - d_w)\sqrt{w_d b_d / t^2}}{\sigma_d} \tag{12}$$

where σ_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.

3.3 PBM

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

$$\omega^{\mathrm{PBM}} = \left(\frac{1}{c}\frac{E_1}{E_c}D_c\right)^2 \tag{13}$$

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

$$E_1 = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{u}_i - \bar{\boldsymbol{\mu}}\| \tag{14}$$

$$E_c = \frac{1}{n}\sum_{p=1}^{c}\sum_{\{\mathbf{u}_i|l_i=p\}}\|\mathbf{u}_i - \boldsymbol{\mu}_p\| \tag{15}$$

$$D_c = \max_{p,q=1,\ldots,c}\|\boldsymbol{\mu}_p - \boldsymbol{\mu}_q\| \tag{16}$$

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 p-th cluster centroid. A large PBM score implies a good separation ability of the representation.

3.4 Variance ratio criterion

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

$$\omega^{\mathrm{VRC}} = \frac{1}{h}\frac{n-c}{c-1}\frac{\mathrm{tr}(\mathbf{D})}{\mathrm{tr}(\mathbf{W})} \tag{17}$$

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

$$\mathbf{W} = \sum_{p=1}^{c}\mathbf{W}_p \tag{18}$$

$$\mathbf{W}_p = \sum_{\{\mathbf{u}_i|l_i=p\}}(\mathbf{u}_i - \boldsymbol{\mu}_p)(\mathbf{u}_i - \boldsymbol{\mu}_p)^T \tag{19}$$

$$\mathbf{D} = \sum_{p=1}^{c} n_p(\boldsymbol{\mu}_p - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_p - \bar{\boldsymbol{\mu}})^T \tag{20}$$

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.

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

$$\begin{aligned} v^{\mathrm{MMD}} = & \frac{1}{Z}\frac{1}{n(n-1)}\sum_{i\neq j}\bar{\mathbf{y}}_i^T\bar{\mathbf{y}}_j \\ & + \frac{1}{n(n-1)}\sum_{i\neq j}\mathbf{y}_{z,i}^T\mathbf{y}_{z,j} - \frac{2}{Z}\frac{1}{n^2}\sum_{u=1}^{Z}\sum_{i,j}\mathbf{y}_{u,i}^T\mathbf{y}_{z,j} \end{aligned} \tag{21}$$

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 in the main text, we transform v^{MMD} by:

$$\omega^{\mathrm{MMD}} = 1 - \frac{v^{\mathrm{MMD}} - v_{\min}}{v_{\max} - v_{\min}} \tag{22}$$

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

4 Discussions

This appendix is the full version of the Section 8.7 in the main text.

4.1 Effect of number of selected base models on MBN-SO and MBN-SD

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

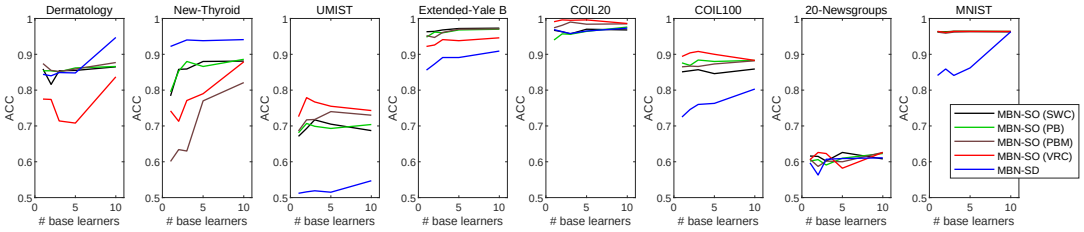
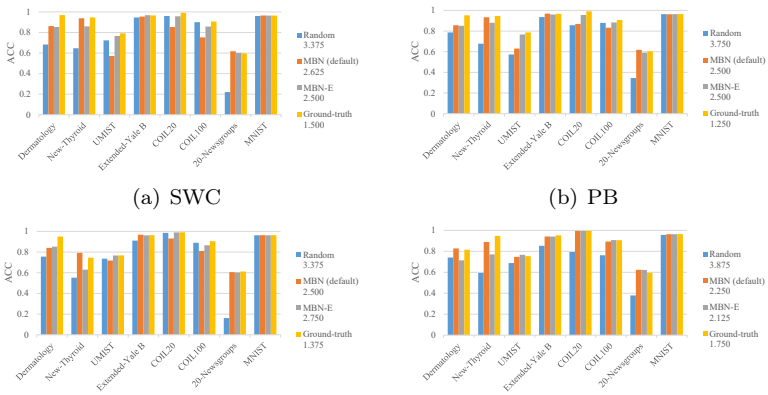


Figure 3: Effect of the number of the selected base models of MBN-SO and MBN-SD on performance.

4.2 Effect of the referenced labels of MBN-SO on performance

The optimization-like criteria of MBN-SO need referenced labels to calculate the weights of the MBN base models, where we adopt the predicted labels from MBN-E as the reference. Here we study whether MBN-SO is sensitive to the referenced labels by generating the labels in different ways, which are (i) randomly generated labels, (ii) predicted labels from "MBN (default)", (iii) predicted labels from MBN-E, and (iv) ground-truth labels.

Fig. 4 shows the comparison results of different referenced-label generation methods. From the figure, we observe the following interesting phenomena. First, using the predicted labels from either "MBN (default)" and MBN-E is equivalently good in terms of the ranking list. Moreover, the methods of using the predicted labels from both MBN-E and "MBN (default)" perform generally very close to the method with the ground-truth labels in terms of ACC, even though the predicted labels themselves do not have a high accuracy, e.g. on



(a) SWC (b) PB

(c) PBM (d) VRC

Figure 4: Effect of the referenced labels on the performance of the MBN-SO variants. The four sub-figures show the results with the selection criteria of (a) SWC, (b) PB, (c) PBM, and (d) VRC, respectively. The numbers in the caption of each sub-figure are the ranks of the comparison methods.

UMIST and 20-Newsgroups. In other words, MBN-SO is insensitive to the accuracy of the referenced labels.

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

Fig. 5 further draws the effect of the referenced labels on the weight calculation of the MBN base models on UMIST and 20-Newsgroups, where the predicted labels from MBN-E and "MBN (default)" are far less accurate than the ground-truth labels. It further manifests the correctness of the aforementioned conclusion. Specifically, from the figure, we see that, although the predicted labels are inaccurate, the weight curves of MBN-E are quite close to those produced by the ground-truth labels, which supports the empirical correctness of using MBN-E to generate the referenced labels for MBN-SO. Although the weight curves of "MBN (default)" are slightly different from those produced by the ground-truth labels, it is still able to select the top MBN base models. At last, we see that the weight curves produced by the randomly generated labels are irregular. Comparing Fig. 5 with Fig. 4, we can further explain the phenomena why the performance with the randomly generated labels seems not so bad is caused by that a number of randomly selected MBN base models are able to produce a reasonable result.

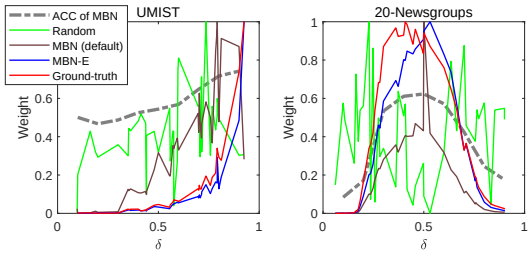


Figure 5: Effect of different referenced-label generation methods on the weights of the base models of "MBN-SO (VRC)".

4.3 On candidate meta-clustering functions of MBN-E

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 base clusterings. From the perspective of ensemble learning, we may also adopt other candidate meta-clustering functions to fuse the clustering results of the MBN base models. In this section, we study the effect of the meta-clustering approaches on performance.

We adopted 12 meta-clustering functions, which are CSPA [3], HGPA [3], MCLA [3], DREC [4], LinkClueE [5,6], ARA1 [7], ARA2 [7], Borda [8], Cvote [9], Vote [10], ECPCS_MC [11], and ECPCS_HC [11], respectively. 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 2 lists the comparison results of the standard 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 [10]. 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 [11]. 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 2 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.

Table 2: 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.

	Dermatology	New-Thyroid	UMIST	Extended-Yale B	COIL20	COIL100	20-Newsgroups	MNIST	**Rank**
CSPA [3]	0.721	0.491	0.592	0.966	0.816	0.677	0.581	0.106	8.125
HGPA [3]	0.306	0.698	0.083	0.027	0.050	0.010	0.053	0.113	12.000
MCLA [3]	0.791	**0.949**	0.602	0.961	0.830	0.726	0.586	**0.965**	5.125
DREC [4]	0.669	0.777	0.500	0.684	0.619	0.545	0.401	N/A	10.875
LinkClueE [5]	0.891	0.948	0.651	0.917	0.894	0.796	N/A	N/A	5.875
ARA1 [7]	0.866	0.897	0.587	0.921	0.837	0.586	0.578	N/A	7.750
ARA2 [7]	0.848	0.937	0.431	0.834	0.757	0.399	0.494	N/A	9.875
Borda [8]	0.922	0.940	0.539	0.888	0.656	0.536	0.516	**0.965**	7.375
Cvote [9]	0.685	0.683	0.631	0.965	**0.981**	0.831	0.204	**0.965**	5.750
Vote [10]	0.867	0.880	0.649	0.968	0.930	0.825	0.618	**0.965**	**3.250**
ECPCS_MC [11]	**0.935**	0.940	0.598	0.947	0.884	0.784	**0.633**	**0.965**	4.125
ECPCS_HC [11]	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	

4.4 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.

This section applies five representative clustering ensemble selection functions to MBN-SO, 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) [12], joint criterion (JC) [12], and cluster and select (CAS) [12]. The selection criteria of the methods consider both the accuracy and diversity of the clustering results.

The second class [13] 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) [13]**: 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 pre-

Table 3: ACC comparison between MBN-SO and the clustering ensemble selection functions that use the same candidate MBN base models as MBN-SO.

	Dermatology	New-Thyroid	UMIST	Extended-Yale B	COIL20	COIL100	20-Newsgroups	MNIST	**Rank**
SNMI [12]	0.708	0.485	0.555	0.823	0.726	0.608	0.534	0.106	15.375
JC [12]	0.746	0.537	0.546	0.947	0.873	0.800	0.556	0.106	11.250
CAS [12]	0.734	0.479	0.560	0.940	0.698	0.617	0.462	0.106	14.250
SIS (SWC) [13]	0.686	0.528	0.559	0.929	0.880	0.776	0.544	0.106	13.000
SIS (PB) [13]	0.682	0.494	0.572	0.930	0.898	0.771	0.544	0.106	12.875
SIS (PBM) [13]	0.658	0.486	0.587	0.910	0.892	0.808	0.483	0.106	13.250
SIS (VRC) [13]	0.643	0.522	0.634	0.909	0.963	0.809	0.545	0.106	11.125
SR [13]	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

diction 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 [13], we used CSPA as the meta-clustering function of SIS.

- **Sum of ranks (SR) [13]** It runs SIS with different optimization-like criteria, each of which produces a 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 [13], we used CSPA as the meta-clustering function of SR.

The top 2 parts of Table 3 lists the comparison result between MBN-SO and the referenced methods [12, 13]. 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

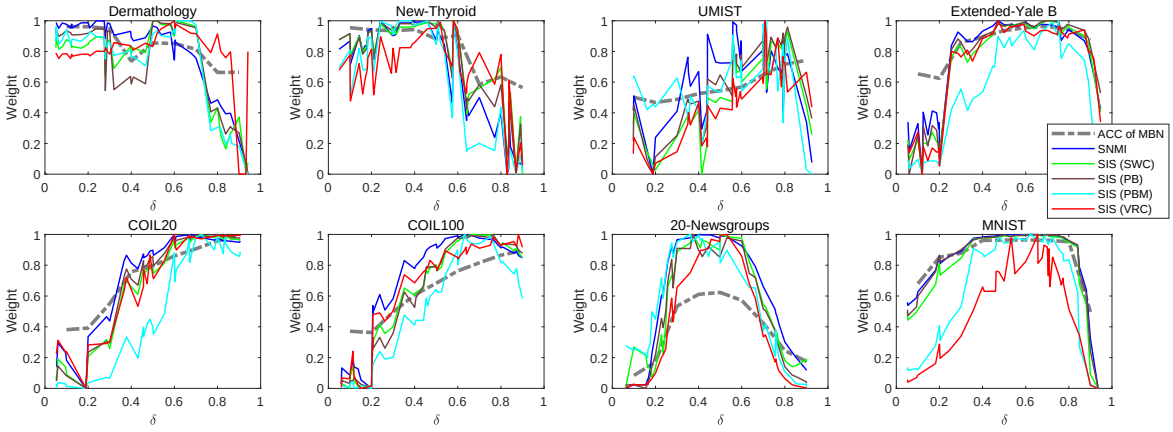


Figure 6: Weights of the MBN base models of the SNMI and SIS functions.

not behave fundamentally different. Particularly, they have failed to achieve reasonable results on MNIST, comparing to random guess.

Fig. 6 shows the weights of the MBN base models of the SNMI and SIS functions 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 3 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 3, 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 3 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.

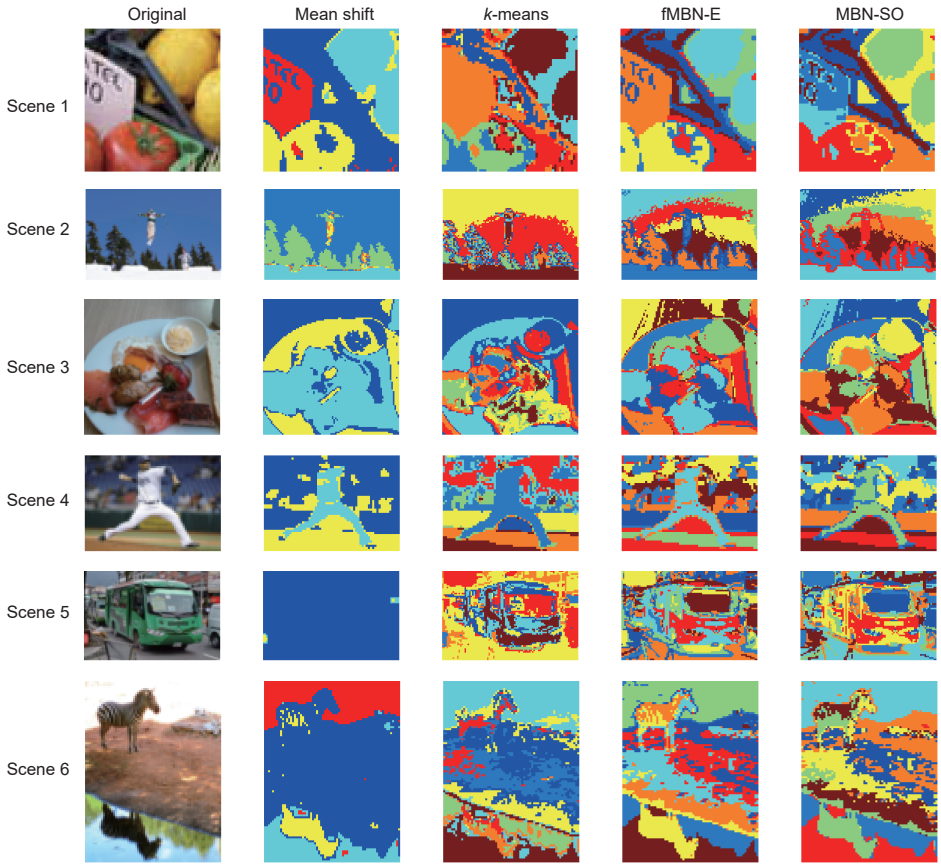


Figure 7: Results of the image segmentation methods on 2 randomly selected examples from the 2017 Val images of the COCO datasets.

5 Applications to image segmentation

Six examples of the comparison results on image segmentation are shown in Fig. 7. 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.

References

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

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009.

[3] 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.

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

[5] 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.

[6] 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.

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

[8] 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.

[9] H. G. Ayad and M. S. Kamel, "On voting-based consensus of cluster ensembles," *Pattern Recognition*, vol. 43, no. 5, pp. 1943–1953, 2010.

[10] 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.

[11] 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.

[12] 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.

[13] 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.