A Self-Supervised Framework for Clustering Ensemble

Liang Du[1,2], Yi-Dong Shen[1], Zhiyong Shen[3], Jianying Wang[4], and Zhiwu Xu[1,2]

[1] State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[2] Graduate University of Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
[3] Baidu Inc. Beijing 100085, China
[4] Computing Center of Shanghai University, Shanghai University, Shanghai, China
{duliang,ydshen,zhiwu}@ios.ac.cn, shenzhiyong@baidu.com, wangjianying@shu.edu.cn

Abstract. Clustering ensemble refers to combine a number of base clusterings for a particular data set into a consensus clustering solution. In this paper, we propose a novel self-supervised learning framework for clustering ensemble. Specifically, we treat the base clusterings as pseudo class labels and learn classifiers for each of them. By adding priors to the parameters of these classifiers, we capture the relationships between different base clusterings and meanwhile obtain a a single consolidated clustering result. In the proposed framework, we are able to incorporate the original data features to improve the performance of clustering ensemble. Another advantage, which distinguishes the proposed framework from the traditional clustering ensemble approaches, is with the generalization capability, i.e. it is able to assign the incoming data instances to the consensus clusters directly based on the original data features. We conduct extensive experiments on multiple real world data sets to show the effectiveness of our method.

Keywords: Cluster Ensemble, Self-Supervised Learning, Logistic Regression.

1 Introduction

Clustering is a common technique for data analysis used in many AI fields, such as machine learning, data mining, pattern recognition. In the last decade, many approaches have been developed to solve the problem of clustering ensemble. Clustering ensemble (See in Figure 1(a)), a.k.a. clustering aggregation [1] or consensus clustering [2], reconciles multiple clustering results $(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_m)$ of a data set, coming from multiple base clusterings, into a single consolidated clustering result $(\boldsymbol{\lambda})$ using a *consensus function* (Γ). To construct a consensus function, the key challenges include: 1) characterizing the relationships between the base clusterings; 2) conducting a single consensus clustering result.

J. Wang et al. (Eds.): WAIM 2013, LNCS 7923, pp. 253–264, 2013.

In the light of these challenges, we propose a novel consensus framework for clustering ensemble, called *Self-Supervised Framework for Clustering Ensemble* (SSCE). The key idea of the proposed framework (See in Figure 1(b)) is to treat X, a matrix induced from base clusterings $(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_m)$ as a *pseudo* data matrix. On the other hand, each individual base cluster assignments could be viewed as *pseudo* class labels. We call the task of learning a set of linear classifiers (with parameters $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m$) over X and $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_m$ as *self-supervised* learning, since there is actually no supervised information. Viewing from bayesian perspective, we assume the parameters of each classifier share the same prior distribution (with parameter $\boldsymbol{\theta}$). We then call the linear classifier, who take expectation ($\boldsymbol{\mu}$) of the prior distribution as parameters, the *consensus classifier*. Based on this consensus classifier, we are able to *classify* a data instance to a consensus cluster assignment ($\boldsymbol{\lambda}$).

There is a significant feature in current clustering ensemble approaches, i.e., the consensus clustering is directly learned from the base clusterings *without* accessing the original data ($\boldsymbol{X}$), which is widely recognized as an advantage of clustering ensemble methods. It is unknown, however, whether we could boost the performance of consensus clustering *with* accessing the original data. Suppose there are incoming data instances, this is another case in which we may also consider the original feature of data instances. It is hard to assign them to the base clusters without accessing the original feature. Therefore, as to our best knowledge, there is no clustering ensemble approach which is capable of handling incoming data instances. This raises another question whether we could assign the incoming data instances to the consensus clusters directly based on their original features.

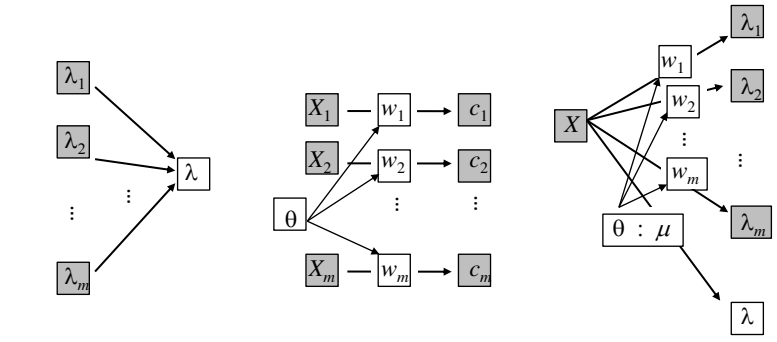
The proposed SSCE framework could be extended to address the above issues by simply replacing the pseudo data matrix with the original data matrix (See in Figure 1(c)). By this means, SSCE incorporates the original data via characterizing the relationship between the original data and base clusterings. The consensus classifier learned from $\boldsymbol{X}$ and $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_m$ could be applied to an incoming data instance to obtain its consensus cluster assignment directly.

The main contributions of this paper include:

1. We propose a novel framework to characterize the linear relationship between the base cluster assignments for clustering ensemble.
2. We extend the proposed framework to incorporate the original data for the purpose of increasing the clustering effectiveness.
3. As to our best knowledge, this work is the first to handle the incoming data instances in clustering ensemble.
4. We conduct extensive empirical evaluations with real life data sets to demonstrate the effectiveness of the proposed framework.

2 Related Work

In this section, we introduce some existing works in the fields of clustering ensemble and multi-task learning, which are closely related to the problem studied in this paper.



(a) Clustering Ensembles (b) Bayesian Multi-task Learning (c) SSCE

Fig. 1. Shaded and unshaded nodes indicate inputs and outputs, respectively

Clustering Ensemble aims to generate a stable and robust consensus clustering by combining multiple base clustering of a dataset. In general, previous works in this area can be grouped into three categories. The first category is graph based approaches. For instance, [3] proposed three graph-based methods. The cluster-based similarity partitioning algorithm (CSPA) uses METIS to partition the induced similarity graph (vertex = objects, edge weight = cluster-based similarity). The hyper graph partitioning algorithm (HGPA) uses HMETIS to partition the hypergraph (vertex = objects, hyperedge = cluster). The meta clustering algorithm (MCLA) collapses related hyperedges and assigns each object to the collapsed hyperedge in which it participates most strongly. In addition, [4] proposed the hybrid bipartite graph partition algorithm, which partitions the bipartite graph (vertex = objects and cluster) by spectral graph partition. [5] proposed an approach to partition weighted similarity graph.

In the second category the algorithms take advantage of probabilistic graphical models. [6] represents objects as a set of attributes from multiple clusterings, and offers a probabilistic model of consensus using a finite mixture of multinomial distribution in a space of clusterings. [7] proposed bayesian cluster ensemble (BCE), which is a generative probabilistic model for learning cluster ensemble.

The third category is matrix factorization based methods. It has been shown [8] that consensus clustering can be formulated within the framework of nonnegative matrix factorization(NMF). [9] proposed weighted consensus clustering, where each input clustering is weighted in such a way that the final consensus clustering provides a better quality solution. [10] proposed weighted graph regularized NMF method which incorporates both the feature based representation and multiple binary relationships based representation.

Multi-task Learning [11, 12] aims to perform multiple learning tasks together to improve individual performance. Rather differently, in clustering ensemble, we aim to produce a *single* high quality consensus clustering. Moreover, multiple tasks are often performed on different data sets. While, clustering ensemble operated on an *identical* data set to reach a consensus.

3 SSCE

In this section, we propose a novel **S**elf-**S**upervised learning framework for **C**lustering **E**nsemble (SSCE) to produce high quality consensus clusterings.

3.1 Notations and Preliminaries

Suppose we are given a data set with n samples $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ and m base clusterings (or partitions) $\Lambda = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, ..., \boldsymbol{\lambda}_m\}$ of the data. $\boldsymbol{\lambda}_i(x_j) \in \{1, ..., k\}$ denote the cluster of $\boldsymbol{x}_j$ given by the i-th base clustering. We use k to denote the cluster numbers.

In this paper we investigate the usefulness of the features of the data set for clustering ensemble. The data set can be represented by original features, if it is available, or pseudo features. The pseudo data matrix based on base clusterings can be constructed as follows: For each base clustering $\lambda_i \in \mathcal{R}^n$, we construct the binary indicator matrix Λ^i, with a column for each cluster. Entries of this matrix $\Lambda^i_{j,k} = 1$ if the j object is assigned to cluster k. Then we concatenate all the block matrix $\Lambda = (\Lambda^i, \ldots, \Lambda^m)$. The pseudo matrix is actually cluster based representation, which is also used in [3] to construct the graph.

3.2 Consistent Labeling

To employ supervised learning approach for clustering ensemble we should align the inconsistency input cluster labels from different base clusterings. In order to achieve the most consistent labeling of clusters between two base clusterings $\boldsymbol{\lambda}_i$ and $\boldsymbol{\lambda}_j$, we must solve an assignment problem, which is also equivalent to a maximum weight bipartite matching problem and can be formulated as follows:

$$I_{k\times k} = \arg\max_{I} \sum_{i=1}^{k}\sum_{j=1}^{k} S_{i,j} I_{i,j}$$
$$s.t. \sum_{j=1}^{k} I_{ij} = \sum_{i=1}^{k} I_{ij} = 1, I_{ij} \in \{0, 1\} \tag{1}$$

where $\{S_{ij}\}$ are the cardinality of intersection of objects labeled i by $\boldsymbol{\lambda}_i$ and objects labeled j by $\boldsymbol{\lambda}_j$, and $\{I_{ij}\}$ are indicators which determine the correspondence between the clusters in the two partitions. An optimal solution of the problem (1) can be found by Hungarian algorithm [13].

A consistent re-labeling of all the base clusterings can be obtained by using a single reference partition λ_r. Ideally, the true label is the best choice for a reference λ_r, however, it is unavailable for clustering ensemble. In practice, any base clustering can be choose as a reference. Then all the remaining base clusterings can be relabeled by solving the problem in Eq. (1) for every pari of partitions $\lambda_r, \lambda_i, i = 1, ..., l, i \neq r$. Once all the base clusterings are relabeled and aligned, they can be seen as a set of *self supervised* labels of the data set.

3.3 Probabilistic Framework

Given an object $\boldsymbol{x}_j$ with original or pseudo features and self supervised label L_j^i under i base clustering. we want to find a consensus mapping function $f : \mathcal{X} \rightarrow \boldsymbol{\lambda}$. In this paper we use logistic regression as discriminative model. For 2-class problem the classification model for $\boldsymbol{x}_j$ under i base clustering can be written in the form

$$P(L_j^i = \pm 1 | \boldsymbol{x}_j, \boldsymbol{w}_i) = \sigma(L_j^i \boldsymbol{w}_i^T \boldsymbol{x}_j) = \frac{1}{1 + \exp(-L_j^i \boldsymbol{w}_i^T \boldsymbol{x}_j)} \tag{2}$$

For multi-class problem, the discriminative model for $\boldsymbol{x}_j$ under i base clustering takes the form

$$P(L_j^i = k | \boldsymbol{x}, \boldsymbol{w}) = \frac{\exp(\boldsymbol{w}_{ik}^T \boldsymbol{x})}{\sum_{k'=1}^{K} \exp(\boldsymbol{w}_{ik'}^T \boldsymbol{x})} \tag{3}$$

where $\boldsymbol{w}_{ik}$ denote the model parameters of class k under i base clustering. These models $\{\boldsymbol{w}_i\}_{i=1}^m$ can be seen as instances which are generated from a consensus model $\boldsymbol{\mu}$. We use the assumption that the base clusterings are i.i.d. given. The joint distribution of data and model parameters reads

$$\begin{aligned} P(W|X, \theta) &\propto P(X, \theta|W) P(W|\theta) \\ &= \prod_{i=1}^{m} P(X, \lambda_i | \boldsymbol{w}_i) P(\boldsymbol{w}_i | \boldsymbol{\mu}) P(W|\theta) \end{aligned} \tag{4}$$

where $P(\boldsymbol{w}_i|\boldsymbol{\mu})$ is a gaussian prior on each base clustering independently. To model the relationships among these base clusterings we add matrix normal distribution $P(W|\theta) = \mathcal{MN}(W|\boldsymbol{\mu}\mathbf{1}_m^T, I \otimes \Omega)$ [14], where the covariance matrix I_d captures the relationships between features and the covariance matrix Ω models the relationships among different base clusterings. Then the MAP estimation of W and MLE estimation of $\theta = \{\boldsymbol{\mu}, \Omega\}$ can be obtained by minimizing the following objective function

$$\begin{aligned} \mathcal{L} = &\sum_{i=1}^{m} \left\{ \sum_{j=1}^{n} l(\boldsymbol{x}_j, L_j^i, \boldsymbol{w}_i) + \gamma_1 ||\boldsymbol{w}_i - \boldsymbol{\mu}||^2 \right\} \\ &+ \gamma_2 \text{trace}((W - \boldsymbol{\mu}\mathbf{1}_m^T)\Omega^{-1}(W - \boldsymbol{\mu}\mathbf{1}_m^T)^T) + d \ln |\Omega| \end{aligned} \tag{5}$$

which is the negative log likelihood of posterior of W. The first three terms in Eq. (5) is convex with respect to both W and θ if we take convex loss function. The last term is concave which make the problem difficult to optimize. The last term $\ln |\Omega|$ is used to penalize the complexity of Ω. To simplify the optimize procedure, we replace the last term with a constraint $\text{trace}(\Omega) \leq 1$ to control the complexity, which has been adopted in [11] [12], the above problem becomes

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{m}\left\{\sum_{j=1}^{n} l(\boldsymbol{x}_j, L_j^i, \boldsymbol{w}_i) + \gamma_1||\boldsymbol{w}_i - \boldsymbol{\mu}||^2\right\} \\ &+ \gamma_2 \text{trace}((W - \boldsymbol{\mu}\mathbf{1}_m^T)\Omega^{-1}(W - \boldsymbol{\mu}\mathbf{1}_m^T)^T) \\ s.t. \quad &\Omega \succeq 0 \\ &\text{trace}(\Omega) \leq 1\end{aligned} \tag{6}$$

4 Learning Algorithm

Though the optimization problem in Eq. (6) is convex w.r.t. all the variables jointly, it is not easy to optimize the problem w.r.t. all the variables simultaneously. We solve problem Eq. (6) by alternatively minimizing the Eq. (6) with respect to each variable by fixing the others. This procedure is repeated until it converges.

4.1 Optimize W by Fixing $\boldsymbol{\mu}$ and Ω

We keep $\boldsymbol{u}$ and Ω fixed and minimize over W, that is we solve the problem

$$\begin{aligned}\min_{W} \quad &\sum_{i=1}^{m}\left\{\sum_{j=1}^{n} l(\boldsymbol{x}_j, L_j^i, \boldsymbol{w}_i) + \gamma_1||\boldsymbol{w}_i - \boldsymbol{\mu}||^2\right\} \\ &+ \gamma_2 \text{trace}((W - \boldsymbol{\mu}\mathbf{1}_m^T)\Omega^{-1}(W - \boldsymbol{\mu}\mathbf{1}_m^T)^T)\end{aligned} \tag{7}$$

One straightforward way to learn W is to set the gradient w.r.t. W to 0 and solve the corresponding linear system. Because the above problem is convex w.r.t. W, it is also convex w.r.t. $\boldsymbol{w}_i$ with all other variables fixed. In this paper we adopt an alternative strategy to perform optimize on W, which is to optimize one column of $\boldsymbol{w}_i$ at a time with the other column fixed. this alternative strategy will be guaranteed to converge to the optimal solution. For 2-class problem, the negative log likelihood of Eq. (2) is

$$l(\boldsymbol{x}_j, L_j^i, \boldsymbol{w}_i) = L_j^i \boldsymbol{w}_i^T \boldsymbol{x}_j - \log(1 + \exp(L_j^i \boldsymbol{w}_i^T \boldsymbol{x}_j)) \tag{8}$$

Hence, the gradient of the above problem with respect to $\boldsymbol{w}_i$ is

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_i} = &\sum_{j=1}^{n}(\delta(L_j^i \boldsymbol{w}_i^T \boldsymbol{x}_j) - 1)L_j^i \boldsymbol{x}_j + 2\gamma_1(\boldsymbol{w}_i - \boldsymbol{\mu}) \\ &+ \gamma_2(W - \boldsymbol{\mu}\mathbf{1}_m^T)\Omega^{-1}(:, i)\end{aligned} \tag{9}$$

For multi-class problem, the negative log likelihood of Eq. (3) is:

$$l(\boldsymbol{x}_j, L_j^i, \boldsymbol{w}_i) = \boldsymbol{w}_{ik}^T \boldsymbol{x}_j - \log(\sum_{k'=1}^{K} \exp(\boldsymbol{w}_{ik'}^T \boldsymbol{x}_j))$$

The gradient of w.r.t. $\boldsymbol{w}_i$ is

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_i} = \sum_{j=1}^{n}(\boldsymbol{w}_{ik}^T \boldsymbol{x}_j - \log(\sum_{k'=1}^{K} \exp(\boldsymbol{w}_{ik'}^T \boldsymbol{x}_j))) + 2\gamma_1(\boldsymbol{w}_i - \boldsymbol{\mu}) + \gamma_2(W - \boldsymbol{\mu}\mathbf{1}_m^T)\Omega^{-1}(:, i) \tag{10}$$

4.2 Optimize $\boldsymbol{\mu}$ by Fixing $\boldsymbol{W}$ and $\boldsymbol{\Omega}$

By setting the gradient of 6 w.r.t. $\boldsymbol{\mu}$ be 0, we get the close form solution of $\boldsymbol{\mu}$

$$\boldsymbol{u} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{w}_i \tag{11}$$

4.3 Optimize $\boldsymbol{\Omega}$ by Fixing $\boldsymbol{W}$ and $\boldsymbol{\mu}$

Fixing W and $\boldsymbol{\mu}$, Ω is determined by following problem:

$$\begin{aligned} \min_{\Omega} \quad & \text{trace}((W - \boldsymbol{\mu}\mathbf{1}_m^T)\Omega^{-1}(W - \boldsymbol{\mu}\mathbf{1}_m^T)^T) \\ s.t. \quad & \Omega \succeq 0 \\ & \text{tr}(\Omega) \leq 1 \end{aligned}$$

The close form solution of above problem is given by

$$\Omega = \frac{((W - \boldsymbol{\mu}\mathbf{1}_m^T)^T(W - \boldsymbol{\mu}\mathbf{1}_m^T))^{\frac{1}{2}}}{\text{tr}(((W - \boldsymbol{\mu}\mathbf{1}_m^T)^T(W - \boldsymbol{\mu}\mathbf{1}_m^T))^{\frac{1}{2}})} \tag{12}$$

where the proof can be found in [11] and [12]. The overall approach, called SSCE, is summarized in Algorithm 1.

5 Experiments

In this section, we empirically evaluate the proposed SSCE framework over multiple benchmark data sets. We begin with a description of these data sets with the evaluation metrics, and then provide the evaluation results of the consensus clustering as well as the generalization capability.

5.1 Dataset Description

We carry out our experiments on totally 13 data sets from UCI machine learning repository. These data sets have been widely used in literatures of clustering ensemble, including [7] and [8]. An overview of these data sets, including numbers of instances, features and classes in each data set, is given in Table 1.

Algorithm 1: SSCE algorithm

Input: data matrix X, base clusterings $\Lambda = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, ..., \boldsymbol{\lambda}_m\}$, γ_1 and γ_2
Output: Cluster labels for data points, classification model $\boldsymbol{\mu}$
Initialization set $\Omega = \frac{I_{m \times m}}{m}$;
while *convergence condition is not true* **do**
  for $i = 1, ..., m$ **do**
    compute $\boldsymbol{w}_i$ using Eq. (9) or Eq. (10);
  end
  update $\boldsymbol{\mu}$ using Eq. (11);
  set Ω by Eq. (12)
end

5.2 Evaluation Measure

We evaluate the clustering ensemble results by comparing the consensus clustering produced by clustering ensemble algorithms with the provided class labels. Specifically, Accuracy of Clustering (AC) is adopted to measure the performance, which discovers the one-to-one relationship between clusters and classes. Given a point $\boldsymbol{x}_i$, let p_i and q_i be the clustering result and the ground truth label, respectively. The ACC is defined as follows:

$$\text{ACC} = \frac{1}{n} \sum_{i=1}^{n} \delta(q_i, map(p_i)), \tag{13}$$

where n is the total number of samples and $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise, and $map(\cdot)$ is the permutation mapping function that maps each cluster index to a true class label. The best mapping can be found by using the Kuhn-Munkres algorithm [15]. The greater clustering accuracy means the better clustering performance.

5.3 Compared Methods

To demonstrate how the performance of clustering ensemble can be improved by our method, we compare the proposed approach with the results of running k-means on the original data set or the base clusterings (KC). These are often used as baselines to verify the clustering ensemble approaches often produce more accurate and robust results against single clustering methods. We also compare our method with the bayesian cluster ensemble (BCE) in [7], the NMF-based consensus clustering (NMFC) in [8], the cluster-based similarity partitioning algorithm (CSPA), the hyper graph partitioning algorithm (HGPA), and the meta-clustering algorithm (MCLA) described in [3]. For the last three methods, we use the authors' matlab implementation ClusterPack [1]. We test our algorithm on data sets with original features (SSCE). we also report the results by experiments on pseudo data matrix (SSCE-P), which is always available for clustering ensemble.

[1] `www.lans.ece.utexas.edu/~strehl/`

Table 1. Description of 13 data sets

Data sets	Samples	Dimensions	Classes
balance	625	4	3
bupa	345	6	2
glass	214	9	6
ionosphere	351	34	2
iris	150	4	3
magic04	19020	10	2
pima	768	8	2
protein	116	20	6
wdbc	569	30	2
wine	178	13	3
segment	2310	19	7
soybean	47	35	4
zoo	101	18	7

5.4 Experiments on Consensus Clustering

We use similar settings with BCE [7] to show the effectiveness of our proposed algorithm. For all reported results, there are two steps leading to the final consensus clustering. First, given n objects, we run k-means 2000 times with random initialization and obtain 2000 base clustering results, which are further divided into 100 subsets, with an $n \times 20$ base clustering matrix each. Then we run clustering ensemble algorithm 100 times on these subsets. All the data sets have been preprocessed such that each feature has zero mean and unit standard deviation. To simplify our model, we set $\gamma_1 = \gamma_2$ in all the experiments, and the best parameter is obtained by search on the grid of $\{0.01, 0.1, 1, 10, 100\}$.

Table 2 shows the clustering accuracy on the data sets. It is observed that the advantage of the proposed algorithm is much more clear for clustering ensemble. For example, the average improvement of SSCE on original features over BCE, the second best algorithm, achieves 5.7% on all the data sets, the average improvement of our approach on pseudo data matrix (SSCE-P) over BCE is 3.4%. Besides, the proposed approach perform statistically significantly better than the compared methods on 10/13 data sets at 95% significance level. Its success can be explained by the fact that the features (Original or Pseudo) are complementary to the base clusterings. We notice that SSCE failed to achieve comparable performance on iris data set with original features. The reason may lies in the iris data set only have 3 features, which make it less discriminative in a supervised manner. When iris is trained on pseudo data matrix, which has 3×20 features, our algorithm produce comparable results. We also observed that our algorithm perform similar on multi class data sets, and SSCE on original features perform better than its counterpart SSCE-P for 2-class problem.

Table 2. Experimental Results in Clustering Accuracy. The × entry for CSPA is due to out of memory (4GB).

Data sets	Approaches								
	Kmeans	MCLA	CSPA	HGPA	KC	BCE	NMFC	SSCE-P	SSCE
balance	0.519	0.498	0.516	0.412	0.522	0.513	0.517	**0.530**	0.526
bupa	0.554	0.554	0.562	0.508	0.554	0.557	0.554	**0.580**	0.577
glass	0.517	0.464	0.412	0.375	0.467	0.503	0.440	0.553	**0.555**
ionosphere	0.712	0.712	0.678	0.584	0.712	0.711	0.712	**0.712**	0.704
iris	0.825	**0.893**	0.874	0.602	0.796	0.881	0.773	0.885	0.775
magic04	0.649	0.649	×	0.500	0.649	0.649	0.648	0.649	**0.680**
pima	0.660	0.660	0.544	0.503	0.660	0.659	0.660	0.660	**0.678**
protein	0.525	0.574	**0.586**	0.569	0.522	0.553	0.570	0.538	0.543
segment	0.527	0.548	0.523	0.464	0.515	0.547	0.543	0.557	**0.588**
soybean	0.706	0.723	0.697	0.726	0.676	0.702	0.616	0.731	**0.770**
wdbc	0.854	0.854	0.670	0.515	0.854	0.825	0.854	0.854	**0.950**
wine	0.673	0.702	0.679	0.563	0.651	0.692	0.702	0.702	**0.786**
zoo	0.690	0.756	0.578	0.574	0.648	0.673	0.639	0.793	**0.807**

5.5 Experiments on Generalization Capability

One of the advantages of SSCE over other clustering ensemble methods is that it has an explicit mapping function over original features. Since SSCE has explicit mapping function, we can choose part of the data to learn a mapping function and use this mapping function to map the rest of data points to the clusters. To evaluate the generalization capability of SSCE, we design the following experiments. For each data set, we firstly randomly split the data set into two parts (60% and 40%), with the 60% used to train the model and the 40% used as the hold-out test set. Then we run kmeans 100 times on train set to generate the base clusterings. We then run clustering ensemble algorithms 5 times, each run with 20 kmeans as input base clusterings, we predict the cluster label of the test set. This whole procedure is repeated 10 times and the average accuracy, that is over $5 * 10$ results, are reported.

Most of the clustering ensemble methods can not directly predict the label of unseen test data. To do this, we assign the cluster label of unseen data with the label of its nearest cluster center, which is computed from train set and consensus clustering result. [7] is a generative graphic model, which can be used to infer the posterior distribution of the clusters. To do this, we firstly run k-means on test set to construct cluster-based representation, then we use the learned BCE model to infer the cluster assignment of the test data (BCE-Infer). we also report the results of prediction of BCE by nearest cluster center strategy (BCE-NC).

Table 3 summarizes results of the second series of experiments. Similar to results in the first experiment, our proposed approach SSCE usually outperforms the other approaches. For example, the average improvement of SSCE on original features over BCE, the second best algorithm, achieves 4.2% on all the data sets.

Table 3. Experimental Results in Classification Accuracy

Data sets	Approaches								
	Kmeans	CSPA	HGPA	MCLA	KC	BCE-Infer	BCE-NC	NMFC	SSCE
balance	0.504	0.469	0.490	0.518	0.487	0.480	**0.532**	0.509	0.470
bupa	0.547	0.547	0.556	0.546	0.547	0.557	0.556	0.547	**0.590**
glass	0.536	0.515	0.473	0.480	0.498	0.550	0.488	0.507	**0.561**
ionosphere	0.717	0.624	0.624	0.624	0.624	0.702	0.693	0.704	**0.720**
iris	0.818	**0.921**	0.918	0.804	0.859	0.898	0.681	0.838	0.777
magic04	0.649	0.649	0.600	0.576	0.649	0.649	0.649	0.648	**0.655**
pima	0.660	0.660	0.592	0.589	0.660	0.659	0.660	0.660	**0.690**
protein	0.543	0.554	**0.582**	0.577	0.549	0.544	0.460	0.547	0.555
segment	0.529	0.558	0.545	0.517	0.522	0.549	0.382	0.545	**0.583**
soybean	0.737	0.770	0.759	0.757	0.751	0.764	0.623	0.676	**0.826**
wdbc	0.855	0.856	0.907	0.906	0.856	0.847	0.805	0.856	**0.945**
wine	0.675	0.696	0.719	0.700	0.689	0.700	0.625	0.696	**0.795**
zoo	0.749	0.734	0.719	0.718	0.729	0.735	0.634	0.721	**0.839**

6 Conclusions

In this paper, we design a novel consensus function for clustering ensemble. We treat the base clusterings as pseudo class labels and learn base classifiers for each of them. By adding priors to the parameters of these classifiers, we capture the relationships between different base clusterings and meanwhile obtain a single consolidated clustering result. We provide the algorithms to estimate the parameters of the base classifiers as well as the prior parameters, from which we induce the consensus classifier. With empirical evaluations over multiple benchmark data sets, we show that the proposed consensus function outperforms the traditional ones. We also demonstrate we may improve the performance of clustering ensemble via incorporating the original data features. Moreover, we exam the generalization capability of the proposed framework and show its advantage in handling incoming data instances. One area of future work is to investigate optimizing the label correspondence together with the parameter estimation. We may directly handel inconsistent labeling problem from multiple base clustering with different number of clusters.

Acknowledgments. We would like to thank all anonymous reviewers for their helpful comments. This work is supported in part by NSFC grant 60970045 and China National 973 project 2013CB329305.

References

1. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. ACM Transactions on Knowledge Discovery from Data (TKDD) 1(1), 4 (2007)
2. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52(1), 91–118 (2003)

3. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research 3, 583–617 (2003)
4. Fern, X.Z., Brodley, C.E.: Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the Twenty-first International Conference on Machine Learning, p. 36. ACM (2004)
5. Al-Razgan, M., Domeniconi, C.: Weighted clustering ensembles. In: Proceedings of 6th SIAM International Conference on Data Mining, pp. 258–269 (2006)
6. Topchy, A., Jain, A.K., Punch, W.: A mixture model for clustering ensembles. In: Proceedings of 4th SIAM International Conference on Data Mining, pp. 379–390 (2004)
7. Wang, H., Shan, H., Banerjee, A.: Bayesian cluster ensembles. Statistical Analysis and Data Mining 4(1), 54–70 (2011)
8. Li, T., Ding, C., Jordan, M.I.: Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In: Proceedings of the 7th IEEE International Conference on Data Mining, pp. 577–582 (2007)
9. Li, T., Ding, C.: Weighted consensus clustering. In: Proceedings of the 8th SIAM International Conference on Data Mining, pp. 798–809 (2008)
10. Du, L., Li, X., Shen, Y.-D.: Cluster ensembles via weighted graph regularized nonnegative matrix factorization. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part I. LNCS, vol. 7120, pp. 215–228. Springer, Heidelberg (2011)
11. Evgeniou, A.A.T., Pontil, M.: Multi-task feature learning. In: Advances in Neural Information Processing Systems, vol. 19, pp. 41–48 (2007)
12. Zhang, Y., Yeung, D.Y.: A Convex Formulation for Learning Task Relationships in Multi-Task Learning. In: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, pp. 733–742 (2010)
13. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics, 32–38 (1957)
14. Gupta, A.K., Nagar, D.K.: Matrix variate distributions, vol. 104. Chapman & Hall/CRC (1999)
15. Lovász, L., Plummer, M.: Matching theory. Elsevier Science Ltd. (1986)
16. Fern, X., Brodley, C.: Random projection for high dimensional data clustering: A cluster ensemble approach. In: Proceedings of the 20th International Conference on Machine Learning, pp. 186–193 (2003)