Yinghui Zhang*

A Kernel Probabilistic Model for Semi-supervised Co-clustering Ensemble

https://doi.org/10.1515/jisys-2017-0513
Received October 12, 2017; previously published online December 30, 2017.

Abstract: Co-clustering is used to analyze the row and column clusters of a dataset, and it is widely used in recommendation systems. In general, different co-clustering models often obtain very different results for a dataset because each algorithm has its own optimization criteria. It is an alternative way to combine different co-clustering results to produce a final one for improving the quality of co-clustering. In this paper, a semi-supervised co-clustering ensemble is illustrated in detail based on semi-supervised learning and ensemble learning. A semi-supervised co-clustering ensemble is a framework for combining multiple base co-clusterings and the side information of a dataset to obtain a stable and robust consensus co-clustering. First, the objective function of the semi-supervised co-clustering ensemble is formulated according to normalized mutual information. Then, a kernel probabilistic model for semi-supervised co-clustering ensemble (KPMSCE) is presented and the inference of KPMSCE is illustrated in detail. Furthermore, the corresponding algorithm is designed. Moreover, different algorithms and the proposed algorithm are used for experiments on real datasets. The experimental results demonstrate that the proposed algorithm can significantly outperform the compared algorithms in terms of several indices.

Keywords: Co-clustering, co-cluster ensemble, semi-supervised learning, kernel probabilistic model, recommend system.

2010 Mathematics Subject Classification: 97R40.

1 Introduction

Co-clustering [4, 5] has recently received much attention in recommendation system applications. Co-clustering and the motivations were first illustrated in a paper [16]. The term co-clustering was later used by Mirkin. A co-clustering algorithm based on variance [5] was proposed for biological gene expression data analysis, in one of the most important literature reports in gene expression analysis. There are also some research works [6, 7] that applied cluster algorithms for bio-information processes. Two algorithms [9] were presented to apply co-clustering to documents and words. The two algorithms were designed based on bipartite spectral graph partitioning and information theory. A co-clustering algorithm-based weighted Bregman distance instead of KL distance [2] was proposed, and the algorithm is suitable for any kind of matrix. A new preference-based multi-objective optimization algorithm [12] was proposed to compete with the gradient ascent approach. The proposed approach use multiple heuristics to process the co-clustering problem, and it also makes a preference selection through the gradient ascent algorithm and the heuristic. A scalable algorithm [18] was designed to co-cluster massive, sparse, and high-dimensional data, and to combine individual clustering results to produce a better final result. The proposed algorithm is particularly suitable for distributed computing environment, which have been revealed in the experiments, and it is implemented on Hadoop platform with MapReduce programming framework in practice. For higher-order data, the authors

***Corresponding author: Yinghui Zhang,** Software Center, Northeastern University, Shenyang 110819, China, e-mail: scuky@163.com

developed [28] a new tensor spectral co-clustering (SCC) method that applies to any non-negative tensor of data. The authors presented a new distributed framework [8] to support efficient implementations of the algorithms with sequential updates, and the framework was evaluated on both a local cluster of machines and the Amazon EC2 cloud.

Co-clustering results are improved by an ensemble learning technique. An ensemble approach [13] is used to improve the performance of these co-clustering methods. The bagged co-clustering method generates a collection of co-clusters by using the bootstrap samples of the original data and aggregates them into new co-clusters. The principle consists in generating a set of co-clusters and aggregating the results. A novel ensemble technique for co-clustering solutions using mutual information [1] is presented. Asteris et al. [3] firstly presented the algorithm with provable approximation guarantees for Max-Agree, which relied on formulating the algorithm as a constrained bilinear maximization over the sets of cluster assignment matrices. Ensemble [17, 25] is a very popular way to improve the accuracy, robustness, and flexibility of learning. A novel robust spectral ensemble clustering [20, 24] approach is proposed for the cluster ensemble, which learns a robust representation for the co-association matrix through a low-rank constraint. Random projection [23] is used in ensemble fuzzy clustering. A co-clustering ensemble method [11] is used to overcome some limitations through repeatedly applying the plaid model with different parameters. Hanczar and Nadif [14] proposed a new method that can improve the accuracies of co-clustering with the ensemble methods, and they [15] also used a bagging approach for gene expression data. An ensemble co-clustering can be formalized by using a binary tri-clustering problem. The author designed a simple and efficient algorithm for the co-clustering problem described above. In order to generate more diverse and high-quality co-clusters to be fused through an ensemble perspective, the author have adopted a well-known multi-modal particle swarm optimization algorithm [21]. An ensemble method for the co-clustering problem [1] that uses optimization techniques to generate consensus is presented. Manifold ensemble learning [19] is used to improve the co-clustering performance, which aims to maximally approximate the intrinsic manifolds of both the feature and sample spaces. An approach [14] is proposed to improve the performance of co-clustering. It is shown that ensemble co-clustering can be considered a problem of binary tri-clustering and the problem can be solved by the proposed algorithm. Except for the co-clustering ensemble algorithms described above, there are also several semi-supervised co-clustering ensemble algorithms. Wang et al. [27] proposed a non-parametric Bayesian approach to co-clustering ensembles. Similar to clustering ensembles, co-clustering ensembles combine several base co-clustering results to obtain a final co-clustering that is a more robust consensus co-clustering. Pio et al. [22] used the co-clustering method to discover the miRNA regulatory networks. Teng and Tan [26] proposed a semi-supervised co-clustering algorithm to find a combinatorial histone code, which is a successful example of co-clustering.

In general, most of the above existing algorithms did not take advantage of ensemble learning and semi-supervised learning. There are two motivations in this paper. First, ensemble learning and semi-supervised learning are integrated to improve the accuracy of co-clustering, which is inspired by the advantage of ensemble learning. Second, the model selection of co-clustering is partially solved by ensemble learning, which is practically used in recommendation systems.

The rest of the paper is organized as follows. In Section 2, the objective function of semi-supervised co-clustering ensemble is proposed in detail. In Section 3, a kernel probabilistic model for semi-supervised co-clustering ensemble (KPMSCE) is designed, and the corresponding algorithm is illustrated in detail. Experimental results are presented in Section 4, and the paper ends with the conclusions in Section 5.

2 Semi-supervised Co-clustering Ensemble

In this section, the pairwise constraints (side information) of co-clustering, which are the extensions of clustering pairwise constraints, are introduced. In general, the pairwise constraints are a popular way for semi-supervised learning. Then, the semi-supervised co-clustering ensemble is illustrated in detail.

2.1 Pairwise Constraints of Co-clustering

The popular way of semi-supervised clustering algorithms is to use the background information of pairwise constraints, such as must-link (ML) and cannot-link (CL) constraints. An ML constraint means that two data points are in the same cluster, while a CL constraint denotes that two data points are in different clusters. However, pairwise constraints will be extended in the problem of co-clustering. Co-cluster ML constraints specify that two entities, or two features, or one entity and one feature must be related, which can be used in co-clustering.

Suppose g_i and g_j are two connected components. Let x_i and x_j be the entities in g_i and g_j, respectively. Let M denote the set of ML constraints. We have

$$(x_i, x_j) \in M, x_i \in g_i, x_j \in g_j.$$

CL constraints denote that two entities, or two features, or one entity and one feature cannot be placed in the same cluster, and CL constraints can also be entailed. Suppose g_i and g_j are two connected components (completely connected subgraphs by ML constraints). x_i and x_j denote the entities in g_i and g_j, respectively. Denote C as the set of CL constraints. Then

$$(x_i, x_j) \in C, x_i \in g_i, x_j \in g_j.$$

Given a data matrix X_m with m rows and n columns. o_i and o_j denote the i^{th} and j^{th} objects (rows) of X_{mn}, while f_i and f_j denote the $ß^{ik}$ and $æ^{\text{th}}$ features (columns) of X_{mn}. Let k_i, k_j, k_i, and k_j be four connected components.

Then, the corresponding pairwise ML constraint sets (including object ML constraint set M_o and feature ML constraint set M_f) are

$$M_o = \{(o_i, o_j) \mid o_i \in k_i; o_j \in k_j\},$$
$$M_f = \{(f_i, f_j) \mid f_i \in k_i; f_j \in k_j\}.$$

Moreover, the pairwise CL constraint sets (including object CL constraint set C_o and feature CL constraint set C_f) are

$$C_o = \{(o_i, o_j) \mid o_i \in k_i; o_j \in k_j; k_i \neq k_j\},$$
$$C_f = \{(f_i, f_j) \mid f_i \in k_i; f_j \in k_j; k_i \neq k_j\}.$$

2.2 Semi-supervised Co-clustering Ensemble Problem Formulation

In this subsection, the semi-supervised co-clustering ensemble objective function is defined. In detail, suppose there is an original data matrix X_{mn} with m rows (i.e. objects) and n columns (i.e. features).

These m objects can be simultaneously grouped into κ row clusters and n columns into ℓ column clusters, so there are $\kappa \times \ell$ co-clusters in total. Moreover, co-clustering can be considered as a set of κ sets of objects $\{\alpha_r \mid r = 1, \ldots, \kappa\}$ and a set of ℓ sets of features $\{\beta_c \mid c = 1, \ldots, \ell\}$, respectively. In general, the procedure can deliver row labels of objects and column labels of features. If there are several base co-clustering algorithms to process the same dataset, sets of row labels and column labels are obtained. Co-clustering ensemble uses a consensus function Γ to combine the set of q row labels $\mu^{(1, \ldots, q)}$ into a single row label μ, and it simultaneously combines the set of q column labels $\nu^{(1, \ldots, q)}$ into a single column label ν.

Commonly, in a dataset, there are $(\mu^{(q)}, \nu^{(q)})$ groupings including $\kappa^{(q)}$ row clusters and $\ell^{(q)}$ column clusters. Γ is defined as a consensus function $\mathbb{N}^{\{m \times t, n \times t\}} \rightarrow \mathbb{N}^{\{m,n\}}$ projecting a set of co-clusterings to an integrated co-clustering:

$$\Gamma : \{(\mu^{(q)}, \nu^{(q)}) \mid q \in \{1, \ldots, t\}\} \rightarrow \{(\mu, \nu), (C^d - M^d)\}. \quad (1)$$

Let the set of groupings $\{(\mu^{(q)}, \nu^{(q)}) \mid q \in \{1, \ldots, t\}\}$ be denoted by Φ. The co-clustering ensemble is used to seek a consensus co-clustering that shares the most information with the original co-clusterings.

Moreover, the side information in the dataset is the two sets of CL C and ML M, and it is called semi-supervised co-cluster ensemble that the side information is used in the combining step of co-clustering. In order to measure the quality of the statistical information that is shared between two co-clusterings, the objective function of the semi-supervised co-clustering ensemble can be defined as follows:

$$(\mu, \nu)^{(\kappa,\ell-\text{opt})} = \arg\max \Sigma_{q=1}^{t} \phi^{(\text{NMI})}\{\widehat{(\mu, \nu)}, (\mu^{(q)}, \nu^{(q)}), (C^d - M^d)\}, \tag{2}$$

where $(\mu, \nu)^{(\kappa,\ell-\text{opt})}$ is the consensus result of co-clustering and it is one of the results that maximize the average mutual information among all individual co-clustering labels $(\mu^{(q)}, \nu^{(q)})$ in Φ. We define a measure between a set of t co-clustering labels, Φ, and a single co-clustering label, $\widehat{(\mu, \nu)}$, as the average normalized mutual information (ANMI) based on this pairwise measure of mutual information, and the definition of ANMI for co-clustering is as follows:

$$\phi^{(\text{ANMI})}(\Phi, \widehat{(\mu, \nu)}) = \frac{1}{t}\sum_{q=1}^{t} \phi^{(\text{NMI})}(\widehat{(\mu, \nu)}, (\mu^{(q)}, \nu^{(q)})). \tag{3}$$

Mutual information is a sound indication of the shared information between a pair of co-clusterings. The normalized mutual information (NMI) was defined as follows:

$$\text{NMI}(X, Y) = \frac{I(X, Y)}{H(X)H(Y)}, \tag{4}$$

where X and Y denote the variables described by the cluster labeling, and $I(X, Y)$ denotes the mutual information between X and Y. $H(X)$ denotes the entropy of X and $H(X) = I(X, Y)$.

In co-clustering, suppose there are two co-clustering labeling variables (X_r, X_c) and (Y_r, Y_c), i.e. (X_r, Y_r), (X_c, Y_c) denote the row cluster labeling variables and column cluster labeling variables, respectively. When we want to obtain the mutual information between the two co-clustering variables, we must measure the mutual information of row cluster labels (X_r, Y_r) and column cluster labels (X_r, X_c), respectively. We define the NMI of co-cluster labeling as follows:

$$\begin{aligned}\text{NMI}((X_r, X_c), (Y_r, Y_c)) &= \text{NMI}(X_r, Y_r) + \text{NMI}(X_c, Y_c) \\ &= \frac{I(X_r, Y_r)}{H(X_r)H(Y_r)} + \frac{I(X_c, Y_c)}{H(X_c)H(Y_c)}.\end{aligned} \tag{5}$$

One can easily find that $\text{NMI}(X_r, X_r) = \text{NMI}(Y_c, Y_c) = 1$. Equation (3) needs to be estimated by using the sampled quantities provided by the co-clusterings. Then, from Eq. (5), the estimation of the NMI $\phi^{(\text{NMI})}$ is

$$\begin{aligned}\phi^{(\text{NMI})}((\mu^i, \nu^i), (\mu^j, \nu^j)) &= \phi^{(\text{NMI})}(\mu^i, \mu^j) + \phi^{(\text{NMI})}(\nu^i, \nu^j) \\ &= \frac{\sum_{\alpha=1}^{\kappa(i)}\sum_{\beta=1}^{\kappa(j)} O_{\alpha,\beta} \log\left(\frac{|O| \cdot O_{\alpha,\beta}}{O_\alpha^i O_\beta^j}\right)}{\sqrt{\left(\sum_{\alpha=1}^{\kappa(i)} O_\alpha^i \log\frac{O_\alpha^i}{|O|}\right)\left(\sum_{\beta=1}^{\kappa(j)} O_\beta^j \log\frac{O_\beta^j}{|O|}\right)}} + \frac{\sum_{\alpha=1}^{\ell(i)}\sum_{\beta=1}^{\ell(j)} F_{\alpha,\beta} \log\left(\frac{|F| \cdot F_{\alpha,\beta}}{F_\alpha^i F_\beta^j}\right)}{\sqrt{\left(\sum_{\alpha=1}^{\ell(i)} F_\alpha^i \log\frac{F_\alpha^i}{|F|}\right)\left(\sum_{\beta=1}^{\ell(j)} F_\beta^j \log\frac{F_\beta^j}{|F|}\right)}},\end{aligned} \tag{6}$$

where $|O|$ and $|F|$ denote the number of objects and features in a co-cluster, respectively; (O_α^i, F_α^i) denote the number of objects and features in co-cluster Co_α according to (μ^i, ν^i); and (O_β^j, F_β^j) denote the number of objects and features in co-cluster Co_β according to (μ^j, ν^j). $O_{\alpha,\beta}$ and $F_{\alpha,\beta}$ denote the number of objects and features, respectively, in co-cluster Co_α according to (μ^i, ν^i) as well as in group Co_β according to (μ^j, ν^j).

3 Semi-supervised Co-clustering Ensemble Based on Kernel Probabilistic Model

In this section, a generative model for semi-supervised co-clustering ensemble based on the kernel probabilistic theory is proposed, and the gradient descent method is used to infer the model. At last, the corresponding algorithm is illustrated step by step.

3.1 Kernel Probabilistic Model for Semi-supervised Co-clustering Ensemble

The model in KPMSCE, a zero mean Gaussian process of $U_{:,d}$ and $V_{:,d}$, is regarded as the prior distribution for the feature of a data set. For a universal situation, a generalization of the multivariate Gaussian distribution can be used for this process in the model. In general, a mean value and a covariance matrix can determine a multivariate Gaussian, and in the same situation, a mean function $m(x)$ and a covariance function $k(x;x^T)$ can also determine the Gaussian process.

For the semi-supervised co-cluster ensemble problem, x is an index of matrix rows or columns in different ways. If $m(x)$ equals 0 and the corresponding kernel function is $k(x;x^T)$, the function can represent the covariance and the corresponding pair of objects or features. $K_U \in R^{N\times N}$ is set to be a full covariance matrix for objects, and it can be a prior that can force the factorization to capture the covariance among rows. Meanwhile, $K_V \in R^{M\times M}$ is set to be a full covariance matrix for features, and it can be a prior that can force the factorization to capture the covariance among columns.

If K_U and K_V are the priors and they are assumed to be known, the generative process steps for KPMSCE are as follows:

1. Sample $U_{:,d}$ according to $GP(0, K_U)$, $[d]_1^D$.
2. Sample $V_{:,d}$ according to $GP(0, K_V)$, $[d]_1^D$.
3. For each object $R_{n,m}$, sample $R_{n,m}$ according to $N(U_n,:V_m^T,:,\sigma^2)$, where σ is a constant.

If U and V are known, the likelihood over the visible objects in the target field R is

$$p(R|U,V,\sigma^2)=\prod_{n=1}^{N}\prod_{m=1}^{M}[N(R_{n,m}|U_n,V_m^T,\sigma^2)]\delta_{n,m}, \tag{7}$$

and U and V are given by

$$p(U|K_U)=\prod_{d=1}^{D}GP(U_{:,d}|0,K_U), \tag{8}$$

$$p(V|K_V)=\prod_{d=1}^{D}GP(V_{:,d}|0,K_V). \tag{9}$$

For simplicity, we denote K_U^{-1} by S_U and K_V^{-1} by S_V. The log-posterior over U and V can be calculated by

$$\begin{aligned}\log p(U,V|R,\sigma^2,K_U,K_V)=&-\frac{1}{2\sigma^2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{n,m}(R_{n,m}-U_n,V_m^T)^2\\&-\frac{1}{2}\sum_{d=1}^{D}U_{:,d}^T S_U U_{:,d}-\frac{1}{2}\sum_{d=1}^{D}V_{:,d}^T S_V V_{:,d}-A\log\sigma^2-\frac{D}{2}(\log|K_U|+\log|K_V|)+C,\end{aligned} \tag{10}$$

where A is the total number of objects and $|K|$ is the determinant of K. The proposed model is a generative model that is used to simulate how to sample the results of base co-clustering results. Then, if we extract the latent labels in the graphical model, at last the semi-supervised co-cluster ensemble results are obtained.

3.2 Inference of KPMSCE Based on Gradient Descent

There are several latent variables to be inferred in this model. Expectation maximization and maximum a posteriori can be used to estimate these latent variables. In this paper, a maximum a posteriori is applied to estimate the latent matrices U and V, and these matrices can maximize the posteriors of the model. In other words, the following objective function can be minimized:

$$E = \frac{1}{2\sigma^2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{n,m}(R_{n,m} - U_n, V_m^T)^2 + \frac{1}{2}\sum_{d=1}^{D}U_{:,d}^T S_U U_{:,d} + \frac{1}{2}\sum_{d=1}^{D}V_{:,d}^T S_V V_{:,d}. \tag{11}$$

In general, gradient descent can be used for minimizing the function E. The gradient of objects (rows) is as follows:

$$\frac{\partial E}{\partial U_{n,d}} = -\frac{1}{\sigma^2}\sum_{m=1}^{M}\delta_{n,m}(R_{n,m} - U_n, V_m^T)V_{d,m} + e_{(n)}^T S_U U_{:,d}, \tag{12}$$

where $e(n)$ is an token vector with the corresponding bit being 1 and others being 0. Then, the gradient of features (columns) is defined as

$$\frac{\partial E}{\partial V_{m,d}} = -\frac{1}{\sigma^2}\sum_{m=1}^{M}\delta_{n,m}(R_{n,m} - U_n, V_m^T)U_{d,m} + e_{(m)}^T S_V V_{:,d}, \tag{13}$$

where $e(m)$ is also an token vector with the corresponding bit being 1 and others being 0. Given the initial guess of the priors, U is updated by

$$U_{n,d}^{t+1} = U_{n,d}^t - \eta\frac{\partial E}{\partial U_{n,d}}, \tag{14}$$

where η is the learning rate for flexibility. It can be settled from 0 to 1. V is updated by

$$V_{n,d}^{t+1} = V_{n,d}^t - \eta\frac{\partial E}{\partial V_{n,d}}. \tag{15}$$

According to the updating functions, U and V are updated alternatively until convergence. When K_U and K_V remain stable throughout all iterations, S_U and S_V are computed only once. In the extreme case, the whole objects or feature are missed but the appropriate side information is known. The update rules with missing values are the following equations:

$$U_{n,d}^{(t+1)} = U_{n,d}^{(t)} - \eta e_{(n)}^T S_U U_{:,d} = U_{n,d}^{(t)} - \eta\sum_{n'=1}^{N}S_U(n,n')U_{n',d}, \tag{16}$$

and

$$V_{m,d}^{(t+1)} = V_{m,d}^{(t)} - \eta e_{(m)}^T S_V V_{:,d} = V_{m,d}^{(t)} - \eta\sum_{m'=1}^{N}S_V(m,m')V_{m',d}. \tag{17}$$

In this case, $U_{n,:}$ is updated according to the weighted average of the current U over all rows, whether the rows are missing or not. The weights $S_U(n, n')$ show the correlation between the current n and all the rest. $V_{m,:}$ is updated according to the weighted average of the current V over all rows, whether the columns are missing or not. The weights $S_V(m, m')$ show the correlation between the current m and all the rest.

3.3 Algorithm

In this subsection, the KPMSCE algorithm is described. The diversity of base co-clustering results is an important reason to improve the result of semi-supervised co-cluster ensemble, so KPMSCE is used to semi-supervise some diversity base co-clustering results to integrate and obtain the final semi-supervised co-clustering ensemble result. According to the above model and inference, a kernel probabilistic algorithm for semi-supervised co-clustering ensemble algorithm is designed. The algorithm procedure is described step by step below.

Algorithm: KPMSCE Algorithm.
Input: Pairwise constraint set $P(i, j)$, original data matrix X_{mn}, number of row clusters κ, and number of column clusters ℓ (i.e. $\kappa \times \ell$ co-clusters in total).
Output: The final consensus co-clustering result.

1. Divide X_{mn} into κ row clusters and ℓ column clusters by the co-clustering algorithms, and the base co-clustering labels are obtained.
2. Compute the NMI among the base co-clusters and obtain a new data matrix.
3. Calculate the likelihood for each column according to the equation

$$P(R_{n<m>,m} \mid U, V) = N(R_{n<m>,m} \mid (U_{n<m>,:} V_m^T, :), \sigma^2 I).$$

4. Marginalize the probability over V, obtaining $P(R \mid U) = \prod_{m=1}^{M} p(R_{n<m>,m} \mid U)$.
5. Compute the objective function, $E = \sum_{m=1}^{M} (R_{n<m>,m}^T C^{-1} R_{n<m>,m} + \log(C)) + \sum_{d=1}^{D} U_{:,d}^T S_U U_{:,d}$, and we can see that V is deleted in the objective function, so the gradient descent can be obtained on U, which is updated at each iteration according to the inversion of C.
6. The maximum likelihood estimation is computed by $\hat{R}_{n,m} = \hat{U}_{n,:} \hat{V}_{m,:}^T$.
7. Obtain the column and row cluster ensemble according to maximum likelihood.
8. Integrate the final row and column clusters.

4 Empirical Study

In this section, 10 datasets are used in the experiments. In particular, eight datasets are from the UCI machine learning repository, and a dataset is from the KDD Cup. The last dataset, called yeast cell data, has been analyzed by using many clustering and co-clustering algorithms. For all reported results, there are two steps to obtain the final co-clustering ensemble results in the experiments. First, a set of base co-clustering labels is obtained by running the base co-clustering algorithms. Second, KPMSCE is applied to the base co-clustering labels to generate the final consensus co-clustering.

The standard deviation of the co-clusters is applied as the criterion. For a machine learning method, a final co-clustering ensemble result has two equally important measures of accuracy. The result is not only good for clusters of objects but also for clusters of features. The quality of the co-cluster is the final comprehensive assessment measure that takes into account both co-clustering aspects. *RSD* is defined as the standard deviation of all rows in a co-cluster, *CSD* as the standard deviation of all columns in a co-cluster, and *CoSD* as the standard deviation of all entries in the co-cluster. The smaller the *RSD*, *CSD*, and *CoSD*, the better the quality of row clustering, column clustering, and co-clustering, respectively.

Because all datasets have their labels, micro-precision (MP) is used to measure the accuracy of the cluster with respect to the true labels. MP is defined as $\mathrm{MP} = \sum_{i=1}^{k} a_i / N$, where k is the number of clusters, N is the number of objects, and a_i denotes the number of objects in the cluster i that are correctly assigned to the corresponding class [29]. Moreover, AMP means average MP.

Table 1: *RSD*, *CSD*, and *CoSD* of Each Data Obtained by Base Co-clustering Algorithms and KPMSCE.

Dataset	*RSD*				*CSD*				*CoSD*			
	SCC	ITCC	BCC	KPMSCE	SCC	ITCC	BCC	KPMSCE	SCC	ITCC	BCC	KPMSCE
sonar	127.80	125.26	**113.87**	124.73	20.92	21.49	22.36	**20.47**	1.32	1.30	**1.23**	1.28
spectheart	246.40	255.15	254.8295	**242.53**	20.16	**20.10**	20.31	20.17	2.12	2.15	2.13	**2.11**
breast	2178.17	2214.55	2161.27	**2126.92**	47.06	47.04	47.96	**46.62**	11.87	11.32	11.35	**11.12**
semeion	7649.95	7814.65	7543.46	**7442.21**	1178.87	1175.61	1205.44	**1171.19**	**54.17**	56.73	56.30	54.66
hepatitis	12,746.32	11,137.8	**3459.69**	8338.81	359.14	**339.87**	362.61	340.78	182.77	166.92	190.58	**124.39**
cred	93,168.57	**5643.97**	5652.27	50,286.18	414.66	**414.05**	417.38	423.19	312.04	351.50	356.21	186.26
yeast	231,596.70	224,648.70	223,290.60	**222,387.00**	26,132.06	24,633.90	25,933.71	**24,354.13**	8389.86	8053.27	8106.78	**7444.40**
Kdd99sub	625,626.50	544,649.20	**208,676.10**	417,943.00	16,336.52	19,701.23	**8433.84**	15,932.78	7426.51	4312.35	**2811.02**	3951.52
hvwnt	927,011.00	966,412.60	**752,109.30**	927,012.40	3,954,112.0	3,964,344.0	**3,170,334.0**	3,954,368.0	79,431.88	79,561.0	**63,554.5**	79,424.0
secom	2,446,928.0	4,492,293.0	3,799,032.0	**2,104,492.0**	75,289.87	74,446.81	75,154.10	**74,193.08**	3227.71	5968.66	6881.71	**2875.36**

Bold values denote the best results corresponding to criteria.

Table 2: *RSD*, *CSD*, and *CoSD* obtained by KPMSCE and RMCCE.

Dataset	sonar	spectheart	breast	semeion	hepatitis	cred	yeast	Kdd99sub	hvwnt	secom
RSD										
RMCCE	**114.85**	305.26	**2221.23**	8676.22	12,782.15	93,588.21	236,639.07	**208,682.96**	957,721.38	5,349,347.15
KPMSCE	124.74	**237.84**	2240.81	**7604.99**	**8338.88**	**50,286.18**	**231,664.26**	417,943.33	**927,012.43**	**1,926,235.25**
CSD										
RMCCE	21.93	20.35	47.56	1208.49	401.30	423.75	27,394.69	20,470.31	**3,602,929.88**	**73,621.96**
KPMSCE	**20.41**	**20.18**	**46.66**	**1171.46**	**340.78**	**423.19**	**24,357.67**	**17,444.44**	3,953,442.72	74,467.02
CoSD										
RMCCE	**1.18**	2.41	11.38	61.11	188.90	322.33	8601.97	4739.50	**72,101.06**	7055.71
KPMSCE	1.28	**2.10**	**11.23**	**54.72**	**124.39**	**186.26**	**7912.92**	**4493.20**	79,424.02	**2873.34**

Bold values denote the best results corresponding to criteria.

4.1 Comparison of Experimental Results Among SCC, Information Theoretic Co-clustering, Bregman Co-clustering, and KPMSCE

To illustrate the performance of KPMSCE, the results obtained by KPMSCE are compared with the co-clustering results generated by SCC [9], information theoretic co-clustering (ITCC) [10], and Bregman co-clustering (BCC) [2]. The experimental procedure is described as follows. First, the base co-clusterings are obtained by running each co-clustering algorithm three times on each dataset, i.e. nine co-clusterings of each dataset are obtained. Then, the final consensus co-clustering is obtained by combining the base co-clusterings via KPMSCE. The experimental results are shown in Table 1.

Among all the algorithms on the 10 datasets, it was six times that KPMSCE achieved the best results, while the other three algorithms only achieved four best results. Table 1 shows that the performance of KPMSCE is better than the other three algorithms most of the times. All results showed that the performance of co-clustering can be enhanced by the ensemble method. Table 1 also shows that the row clustering performance of BCC is better compared to the other two co-clustering algorithms because the number of the best row clustering results obtained by BCC is bigger than that of the other co-clustering algorithms. Similarly, we can find that the column clustering performance of ITCC is the best among the three base co-clustering algorithms.

4.2 Comparison of Experimental Results Between KPMSCE and Relational Multi-manifold Co-clustering Ensemble

To demonstrate how the method works for the co-clustering problem and improves the co-clustering performance, it is compared with the co-clustering ensemble method named relational multi-manifold co-clustering ensemble (RMCCE) [19]. The base co-clustering labels are obtained by running each co-clustering algorithm five times on each dataset, and the results are shown in Table 2. In Table 2, it is clear that KPMSCE outperforms the RMCCE most of the times. In other words, the co-clustering ensemble method actually gives better results than RMCCE.

4.3 AMP Results

In this subsection, SCC, ITCC, BCC, KPMSCE, and RMCCE are used for this experiment on all datasets. There are two steps in the proposed experiment. First, SCC, ITCC, and BCC with random initializations are used

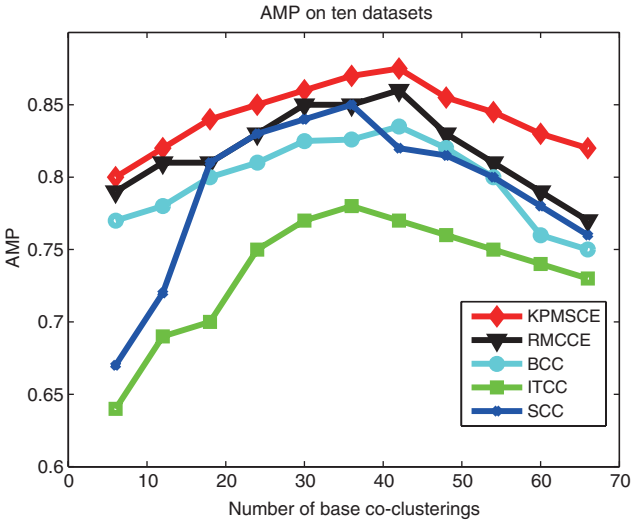

Figure 1: AMP Results of Algorithms on 10 Datasets.

many times for the base co-clustering, and the average AMPs are recorded on different numbers of base co-clusterings. Second, all base co-clustering results are drawn as input data of KPMSCE and RMCCE for the co-clustering ensemble, and the ensemble results are recorded. The results are reported in Figure 1. The x-axis shows the number of base co-clusterings, and the y-axis shows the AMP results on different numbers of base co-clusterings. We can see that KPMSCE obtains the best AMP result, and RMCCE obtains the second best. The results show that ensemble learning can improve the performance of co-clustering. Moreover, semi-supervised learning can positively leverage the base co-clustering and co-clustering ensemble.

5 Conclusions

In this paper, the semi-supervised co-cluster ensemble was illustrated in detail based on semi-supervised learning and ensemble learning. Semi-supervised co-cluster ensembles provide a framework for combining multiple base co-clusterings and the side information of a dataset to generate a stable and robust consensus co-clustering. Moreover, the objective function of the semi-supervised co-cluster ensemble was formulated in detail. Then, KPMSCE was presented, and the inference-oriented KPMSCE was illustrated in detail. Furthermore, the corresponding algorithm was designed. In addition, different algorithms and the proposed algorithm were used for experiments on a real dataset. The experimental results demonstrated that the proposed algorithm can significantly outperform the compared algorithms in terms of several indices.

Future work will focus on the diversity of the base co-clustering labels for the co-clustering ensemble.

Bibliography

[1] G. Aggarwal and N. Gupta, BEMI bicluster ensemble using mutual information, in: *2013 12th International Conference on Machine Learning and Applications (ICMLA)*, 1, pp. 321–324, IEEE, 2013.
[2] B. Arindam, A generalized maximum entropy approach to Bregman co-clustering and matrix approximation, in: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 509–514, 2004.
[3] M. Asteris, A. Kyrillidis, D. Papailiopoulos and A. G. Dimakis, Bipartite correlation clustering – maximizing agreements, in: *Proceedings of the 19th International Conference on Artificial Intelligence and statistics*, pp. 121–129, 2016.
[4] A. Beutel, A. Ahmed and A. J. Smola, ACCAMS: additive co-clustering to approximate matrices succinctly, in: *International Conference on World Wide Web*, pp. 119–129, 2015.
[5] J. Cheng, Z. -S. Tong and L. Zhang, Scaling behavior of nucleotide cluster in DNA sequences, *J. Zhejiang Univ. Sci. B* **8** (2007), 359–364.
[6] J. Cheng and L. -x. Zhang, Statistical properties of nucleotide clusters in DNA sequences, *J. Zhejiang Univ. Sci. B* **6** (2005), 408–412.
[7] X. Cheng, S. Su, L. Gao and J. Yin, Co-ClusterD: a distributed framework for data co-clustering with sequential updates, *IEEE Trans. Knowl. Data Eng.* **27** (2015), 3231–3244.
[8] Y. Cheng and G. M. Church, Biclustering of expression data, in: *International Conference on Intelligent Systems for Molecular Biology*, vol. 8, pp. 93–103, 2000.
[9] I. S. Dhillon, Co-clustering documents and words using bipartite spectral graph partitioning, in: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 269–274, 2001.
[10] I. S. Dhillon, S. Mallela and D. S. Modha, Information-theoretic co-clustering, in: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 89–98, ACM, 2003.
[11] P. Georg, *Ensemble Methods for Plaid Bicluster Algorithm*, 2010.
[12] F. Gullo, A. K. M. K. A. Talukder, S. Luke, C. Domeniconi and A. Tagarelli, Multiobjective optimization of co-clustering ensembles, in: *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion*, pp. 1495–1496, 2012.
[13] B. Hanczar and M. Nadif, Bagged biclustering for microarray data, in: *ECAI*, pp. 1131–1132, 2010.
[14] B. Hanczar and M. Nadif, Using the bagging approach for biclustering of gene expression data, *Neurocomputing* **74** (2011), 1595–1605.
[15] B. Hanczar and M. Nadif, Ensemble methods for biclustering tasks, *Pattern Recognit.* **45** (2012), 3938–3949.
[16] J. A. Hartigan, Direct clustering of a data matrix, *J. Am. Stat. Assoc.* **67** (1972), 123–129.
[17] D. Huang, C. D. Wang and J. H. Lai, Locally weighted ensemble clustering, *IEEE Trans. Cybern.* **PP** (2017), 1–14.

[18] Q. Huang, X. Chen, J. Huang, S. Feng and J. Fan, Scalable ensemble information-theoretic co-clustering for massive data, in: *Proceedings of the International Multiconference of Engineers and Computer Scientists*, vol. 1, 2012.
[19] P. Li, J. Bu, C. Chen and Z. He, Relational co-clustering via manifold ensemble learning, in: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 1687–1691, ACM, 2012.
[20] H. Liu, J. Wu, T. Liu, D. Tao and Y. Fu, Spectral ensemble clustering via weighted K-means: theoretical and practical evidence, *IEEE Trans. Knowl. Data Eng.* **29** (2017), 1129–1143.
[21] L. Menezes and A. L. V. Coelho, On ensembles of biclusters generated by NichePSO, in: *2011 IEEE Congress on Evolutionary Computation (CEC)*, pp. 601–607, IEEE, 2011.
[22] G. Pio, D. Malerba, D. D'Elia and M. Ceci, Integrating microRNA target predictions for the discovery of gene regulatory networks: a semi-supervised ensemble learning approach, *BMC Bioinformatics* **15** (2014), S4.
[23] P. Rathore, J. C. Bezdek, S. M. Erfani, S. Rajasegarar and M. Palaniswami, Ensemble fuzzy clustering using cumulative aggregation on random projections, *IEEE Trans. Fuzzy Syst.* **PP** (2017), 1–1.
[24] Z. Tao, H. Liu and Y. Fu, Simultaneous clustering and ensemble, in: *AAAI*, 2017.
[25] Z. Tao, H. Liu, S. Li and Y. Fu, *Robust spectral ensemble clustering*, pp. 367–376, 2016.
[26] L. Teng and K. Tan, Finding combinatorial histone code by semi-supervised biclustering, *BMC Genomics* **13** (2012), 301.
[27] P. Wang, K. B. Laskey, C. Domeniconi and M. I. Jordan, Nonparametric Bayesian co-clustering ensembles, in: *SDM*, pp. 331–342, SIAM, 2011.
[28] T. Wu, A. R. Benson and D. F. Gleich, *General tensor spectral co-clustering for higher-order data*, 2016.
[29] Z. Zhou and W. Tang, Clusterer ensemble, *Knowl. Based Syst.* **19** (2006), 77–83.