IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 18, 2019, accepted February 11, 2019, date of publication March 13, 2019, date of current version March 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904314

An Overview of Co-Clustering via Matrix Factorization

RENJIE LIN, SHIPING WANG, (Member, IEEE), AND WENZHONG GUO, (Member, IEEE)
College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou 350116, China
Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China

Corresponding author: Wenzhong Guo (guowenzhong@fzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502104 and Grant 61672159, in part by the Fujian Collaborative Innovation Center for Big Data Application in Governments, and in part by the Technology Innovation Platform Project of Fujian Province under Grant 2014H2005.

ABSTRACT Co-clustering algorithms have been widely used for text clustering and gene expression through matrix factorization. In recent years, diverse co-clustering algorithms which group data points and features synchronously have shown their advantages over traditional one-side clustering. In order to solve the co-clustering problems, most existing methods relaxed constraints via matrix factorization. In this paper, we provide a detailed understanding of six co-clustering algorithms with different performance and robustness. We conduct comprehensive experiments in eight real-world datasets to compare and evaluate these co-clustering methods based on four evaluation metrics including clustering accuracy, normalized mutual information, adjusted rand index, and purity. Our findings demonstrate the strengths and weaknesses of these methods and provide insights to motivate further exploration of co-clustering methods and matrix factorization.

INDEX TERMS Machine learning, co-clustering, graph regularization, clustering, matrix factorization.

I. INTRODUCTION

Clustering has long been a fundamental topic in unsupervised machine learning. It focuses on partitioning data points into groups based on their similarities. In order to solve clustering problems, diverse algorithms have been proposed in recent years [1]. Clustering algorithms such as k-means [2], spectral clustering [3], [4], normalized cut [5], min-max cut [6] and non-negative matrix factorization (NMF) [7], have been successfully applied to data mining [8], [9] and computer vision [5], [10]–[12].

Traditional clustering algorithms are designed for one-side clustering and proposed to cluster samples based on the similarities along the feature side and vice versa [13]. However, in the one-side clustering mechanism, there is a lack of consideration about the duality between samples and features. Thus co-clustering algorithms have been proposed and demonstrated the superiority to traditional one-side clustering. For instance, Van Pham *et al.* [14] proposed a new cluster tendency assessment method for fuzzy co-clustering, Hu *et al.* [15] proposed a unsupervised audiovisual learning model, named as deep co-clustering, Jacques and Biernacki [16] proposed a model-based co-clustering algorithm for ordinal data and Gu and Zhou [17] proposed a Dual Regularized Co-Clustering algorithm.

Non-negative matrix factorization [18], [19] has captured enormous interest in machine learning and computer vision. NMF gains advantages in numerous problem formulations optima for clustering algorithms. It is noteworthy that k-means and spectral clustering are able to be expressed as certain canonical forms of non-negative matrix factorization [20], [21].

It is examined that most co-clustering algorithms are designed via matrix factorization and evaluated with diverse perspectives and datasets. The focus of this paper is to revisit existing matrix factorization based co-clustering algorithms and compare these algorithms in co-clustering problem based on eight real-world datasets. We provide insights and evaluate the performance of these algorithms with a set of metrics.

The contributions of this paper are:

- Provide insights about the taxonomy and differences of the various co-clustering algorithms. This is beneficial for understanding the problem formulation and optimization routine and comparing the advantages and disadvantages of each method.

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqiang Wang.

- Demonstrate the visualization of described algorithms in a set of point data constructed in two moons pattern. The results of these algorithms are compared by different clustering performance. Then we evaluate these algorithms with a set of metrics including clustering accuracy (ACC), normalized mutual information (NMI), adjusted rand index (ARI) and purity which are of benefit to comprehend the performance of these algorithms.

The rest of this paper is arranged as follows. In Section II, we provide understanding with a set of co-clustering via matrix factorization. In Section III, datasets, evaluation metrics and experimental results and analyses are provided. Finally, this paper is concluded in Section IV.

II. CO-CLUSTERING ALGORITHMS

Compared with the traditional one-side clustering, co-clustering algorithms which can categorize data points and features synchronously have more powerful performance [22]. Co-clustering could be used for an extensive range of applications [23]. For instance, Rege *et al.* [24] used document-word co-clustering for clustering similar documents and topics. Chen *et al.* [25] and Felzenszwalb and Huttenlocher [26] made use of image co-clustering for image processing. Meanwhile, co-clustering has also been applied to identification of interaction networks [27], [28].

Therefore, this paper provides detailed understanding about six matrix factorization based co-clustering algorithms and analyzes their performance. It is worth noting that we provide insights of these algorithms from an algorithmic level and compare them through comprehensive experiments. Meanwhile, these algorithms introduced diverse novelties and insights, which is useful to beginners who are new to co-clustering.

For the co-clustering problem formulation, given a dataset $\boldsymbol{X} = \{x_1, \cdots, x_n\} = \{f_1, \cdots, f_d\} \in \mathbb{R}^{d\times n}$, group the data points $\{x_1, \cdots, x_n\}$ into c clusters $\{C_i\}_{i=1}^{c}$, while the features $\{f_1, \cdots, f_d\}$ into m clusters $\left\{C_j'\right\}_{j=1}^{m}$. A partition matrix $\boldsymbol{F} \in \{0,1\}^{n\times c}$ is used to represent the clustering result of data points, while $\boldsymbol{G} \in \{0,1\}^{d\times m}$ corresponds to the result of features. And $\|\circ\|_F$ represents the Frobenius norm.

A. NON-NEGATIVE MATRIX FACTORIZATION

Non-negative matrix factorization (NMF) is a useful decomposition for multivariate data [7], [29]. Nonnegativity is a helpful constraint for matrix factorization and can be used for learning a portion of representation of the data [29], [30]. Several new variations on the theme of NMF were proposed, such as semi-NMF [31] and SNMF [32]. This algorithm aims to solve the following problem:

Given a non-negative matrix $\boldsymbol{V}$, find non-negative matrix factors $\boldsymbol{W}$ and $\boldsymbol{H}$ such that:

$$\boldsymbol{V} \approx \boldsymbol{WH} \tag{1}$$

where the dimension of matrix $\boldsymbol{V}$ is $n \times m$. This matrix is approximately resolved into factors: an $n \times r$ matrix $\boldsymbol{W}$ and an $r \times m$ matrix $\boldsymbol{H}$.

NMF does not aim to find an exactly factorization $\boldsymbol{V} \approx \boldsymbol{WH}$, but aims to make $\boldsymbol{V}$ and $\boldsymbol{WH}$ as approximate as close as possible. Thus a cost function $J(\boldsymbol{V}, \boldsymbol{W}, \boldsymbol{H})$ is necessary to quantify the quality of the approximation. If J is smaller, $\boldsymbol{V}$ will be more approximate to $\boldsymbol{WH}$. NMF consider the features (such as continuity or concavity) of J to use suitable optimization methods for working out the $\boldsymbol{W}$ and $\boldsymbol{H}$. This cost function could be diverse and two useful measures are shown below:

- $J = \|\boldsymbol{V} - \boldsymbol{WH}\|_F^2$ is the square of the Euclidean distance between $\boldsymbol{A}$ and $\boldsymbol{B}$ [33].

$$\|\boldsymbol{A} - \boldsymbol{B}\|_F^2 = \sum_{ij} \left(\boldsymbol{A}_{ij} - \boldsymbol{B}_{ij}\right)^2 \tag{2}$$

  This is lower bounded by 0 and clearly vanishes if and only if $\boldsymbol{A} = \boldsymbol{B}$.
- $J = D(\boldsymbol{V} \parallel \boldsymbol{WH})$ is simply the Kullback-Leibler divergence between $\boldsymbol{A}$ and $\boldsymbol{B}$.

$$D(\boldsymbol{A} \parallel \boldsymbol{B}) = \sum_{ij} \left(\boldsymbol{A}_{ij} \log \frac{\boldsymbol{A}_{ij}}{\boldsymbol{B}_{ij}} - \boldsymbol{A}_{ij} + \boldsymbol{B}_{ij}\right) \tag{3}$$

  It reduces to the relative entropy. If $\sum_{ij} \boldsymbol{A}_{ij} = \sum_{ij} \boldsymbol{B}_{ij} = 1$, $\boldsymbol{A}$ and $\boldsymbol{B}$ could be regarded as normalized probability distributions.

Both measures work out the optimal solution through Lagrangian multiplier method and Karush Kuhn Tucker (KKT). When $\boldsymbol{W}$ and $\boldsymbol{H}$ are stable point, the iteration is convergent.

In this paper, we use the performance of NMF to cluster data as the baseline. The clustering application of NMF could be regarded as: a dataset has m examples and each example has n features, so that it constitutes matrix $\boldsymbol{X}$. Then we need to use the NMF to find out $\boldsymbol{W}$ and $\boldsymbol{H}$. The whole process aims to transform the clustering problem of matrix $\boldsymbol{X}$ to clustering problem of matrix $\boldsymbol{H}$ through dimensionality reduction. While NMF is able to be utilized to solve clustering problems, it needs a post-processing step to output the clustering result.

B. DUAL REGULARIZED CO-CLUSTERING

Dual regularized co-clustering (DRCC) algorithm [17] is based on semi-non-negative matrix tri-factorization and this algorithm inherits the strengths of ONMTF [34]. Gu and Zhou [17] considered that data points and features should be both sampled from some manifolds, then they embedded the geometric structure of data manifold and feature manifold. The processes of co-clustering are formulated as semi-non-negative matrix tri-factorization through these two graph regularizers. It is worth noting that cluster labels of data points/features should be smooth concerning to the intrinsic data/feature manifold. In summary, DRCC takes into consideration the geometry of data points and features, thus it could perform well for clustering data on manifold. Meanwhile, DRCC could be optimized by iterative multiplicative updating algorithm and it is convergent in theory.

Gu and Zhou [17] supported that existing co-clustering algorithms [34]–[36] do not take into account the geometric structure when clustering data on manifold. Therefore, DDRC focuses on constructing data graph and feature graph to explore the geometric structure of data/feature manifold.

- Data Graph
  According to cluster assumption, Gu and Zhou [17] supported that if data point x_i is close to x_j, then its cluster labels x_i' should be close to the x_j'. The degree of x_i' close to x_j' could be expressed by $\boldsymbol{W}_{ij}^F$ which is shown as follows:

$$\boldsymbol{W}_{ij}^F = \begin{cases} 1, & \text{if } x_j \in \mathcal{N}(x_i) \text{ or } x_i \in \mathcal{N}(x_j) \\ 0, & \text{otherwise} \end{cases}$$

  where $\mathcal{N}(x_i)$ denotes the k-nearest neighbor set of x_i. Heat kernel [37] also can be used for measuring this affinity. The formulation is shown below:

$$\begin{aligned} \frac{1}{2}\sum_{i,j}\left\|x_i' - x_j'\right\|_F^2 \boldsymbol{W}_{ij}^F &= \sum_{i,j} x_i' \boldsymbol{W}_{ij}^F x_i'^T - \sum_{i,j} x_i' \boldsymbol{W}_{ij}^F x_j'^T \\ &= \sum_{i} x_i' \boldsymbol{D}_{ii}^F x_i'^T - \sum_{i,j} x_i' \boldsymbol{W}_{ij}^F x_j'^T \\ &= tr\left(\boldsymbol{F}^T\left(\boldsymbol{D}^F - \boldsymbol{W}^F\right)\boldsymbol{F}\right) \\ &= tr\left(\boldsymbol{F}^T \boldsymbol{L}_F \boldsymbol{F}\right) \end{aligned} \tag{4}$$

  where $\boldsymbol{D}_{ii}^F = \sum_j \boldsymbol{W}_{ij}^F$ is the diagonal degree matrix and $\boldsymbol{L}_F = \boldsymbol{D}^F - \boldsymbol{W}^F$ is the graph Laplacian [38] of the data graph.
- Feature Graph
  Similar with the construction of the data graph. If feature f_i is close to f_j, its cluster label f_i' should be close to the f_j'. The feature affinity matrix $\boldsymbol{W}^G$ is shown as follows:

$$\boldsymbol{W}_{ij}^G = \begin{cases} 1, & \text{if } f_j \in \mathcal{N}(f_i) \text{ or } f_i \in \mathcal{N}(f_j) \\ 0, & \text{otherwise} \end{cases}$$

  where $\mathcal{N}(f_i)$ denotes the k-nearest neighbor set of f_i. The formulation is demonstrated below:

$$\begin{aligned} \frac{1}{2}\sum_{i,j}\left\|f_i' - f_j'\right\|_F^2 \boldsymbol{W}_{ij}^G &= tr\left(\boldsymbol{G}^T\left(\boldsymbol{G}^G - \boldsymbol{W}^G\right)\boldsymbol{G}\right) \\ &= tr\left(\boldsymbol{G}^T \boldsymbol{L}_G \boldsymbol{G}\right) \end{aligned} \tag{5}$$

  where $\boldsymbol{D}_{ii}^G = \sum_j \boldsymbol{W}_{ij}^G$ is the diagonal degree matrix and $\boldsymbol{L}_G = \boldsymbol{D}^G - \boldsymbol{W}^G$ is the graph Laplacian [38] of the feature graph.

According to the data and feature graph regularizers, the objective function of co-clustering J_{DRCC} is shown as follows:

$$\left\|\boldsymbol{X} - \boldsymbol{G}\boldsymbol{S}\boldsymbol{F}^T\right\|_F^2 + \lambda tr\left(\boldsymbol{F}^T \boldsymbol{L}_F \boldsymbol{F}\right) + \mu tr\left(\boldsymbol{G}^T \boldsymbol{L}_G \boldsymbol{G}\right) \tag{6}$$

where $\lambda, \mu \geq 0$ are the regularization parameters which are used to balance the reconstruction error and the label smoothness. And $\boldsymbol{S}$ is a matrix which could be any signs. Then [17] releases $\boldsymbol{F}$ and $\boldsymbol{G}$ into continuous non-negative domain for reducing difficulty of calculation. Thus DRCC in Equation (6) converts into the objective function below:

$$\begin{aligned} &\left\|\boldsymbol{X} - \boldsymbol{G}\boldsymbol{S}\boldsymbol{F}^T\right\|_F^2 + \lambda tr\left(\boldsymbol{F}^T \boldsymbol{L}_F \boldsymbol{F}\right) + \mu tr\left(\boldsymbol{G}^T \boldsymbol{L}_G \boldsymbol{G}\right) \\ &\text{s.t. } \boldsymbol{G} \geq 0, \boldsymbol{F} \geq 0 \end{aligned} \tag{7}$$

Equation (7) could be regarded as Dual Regularized Semi-Non-negative Matrix Tri-Factorization (DRSNMTF). In order to solve Equation (7), it could repeat the processes of fixing most of variable except one variable until convergence. The summary processes are demonstrated in Algorithm 1.

Algorithm 1 Dual Regularized Co-Clustering

Input: Data matrix $\boldsymbol{X} \in \mathbb{R}^{d\times n}$, the number of data clusters c, the number of feature clusters m, regularization parameters λ, μ, maximum number of iterations T.

Output: Partitions $\boldsymbol{F} \in \mathbb{R}^{n\times c}$, $\boldsymbol{G} \in \mathbb{R}^{d\times m}$.

1: Initialize $\boldsymbol{F}$ and $\boldsymbol{G}$ using k-means.
2: **while** not convergent **and** $t \leq T$ **do**
3: Compute $\boldsymbol{S} = (\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T\boldsymbol{X}\boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-1}$.
4: Update $\boldsymbol{F}_{ij} \leftarrow \boldsymbol{F}_{ij}\sqrt{\frac{[\lambda \boldsymbol{L}_F^- \boldsymbol{F} + \boldsymbol{A}^+ + \boldsymbol{F}\boldsymbol{B}^-]_{ij}}{[\lambda \boldsymbol{L}_F^+ \boldsymbol{F} + \boldsymbol{A}^- + \boldsymbol{F}\boldsymbol{B}^+]_{ij}}}$.
5: Update $\boldsymbol{G}_{ij} \leftarrow \boldsymbol{G}_{ij}\sqrt{\frac{[\lambda \boldsymbol{L}_G^- \boldsymbol{G} + \boldsymbol{P}^+ + \boldsymbol{G}\boldsymbol{Q}^-]_{ij}}{[\lambda \boldsymbol{L}_G^+ \boldsymbol{G} + \boldsymbol{P}^- + \boldsymbol{G}\boldsymbol{Q}^+]_{ij}}}$.
6: **end while**
7: **return** $\boldsymbol{F}$ and $\boldsymbol{G}$.

where $\boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{G}\boldsymbol{S}$, $\boldsymbol{B} = \boldsymbol{S}^T\boldsymbol{G}^T\boldsymbol{G}\boldsymbol{S}$ and $\boldsymbol{A} = \boldsymbol{A}^+ - \boldsymbol{A}^-$, $\boldsymbol{B} = \boldsymbol{B}^+ - \boldsymbol{B}^-$, $\boldsymbol{A}_{ij}^+ = \left(\left|\boldsymbol{A}_{ij}\right| + \boldsymbol{A}_{ij}\right)/2$, $\boldsymbol{A}_{ij}^- = \left(\left|\boldsymbol{A}_{ij}\right| - \boldsymbol{A}_{ij}\right)/2$ and $\boldsymbol{P} = \boldsymbol{X}\boldsymbol{F}\boldsymbol{S}^T$, $\boldsymbol{Q} = \boldsymbol{S}\boldsymbol{F}^T\boldsymbol{F}\boldsymbol{S}^T$.

DRCC is difficult to be applied to large-scale data in real world applications because intensive matrix multiplications involved in each iteration step, which takes up a lot of computation time for solving the clustering problems.

C. GRAPH DUAL REGULARIZATION NON-NEGATIVE MATRIX TRI-FACTORIZATION

Graph dual regularization non-negative matrix tri-factorization (DNMTF) [39] makes use of non-negative matrix tri-factorization to solve co-clustering problems and proposes an optimization scheme which grounds on iterative updating rules. The objective function of this algorithm simultaneously group the graph regularizers of data manifold and feature manifold. Similar to DRCC [17], DNMTF constructs data graph and feature graph to effectively model the geometric structures of data and feature manifold. Shang *et al.* [39] constructed a k-nearest neighbor data graph: $\{x_1, \cdots, x_n\}$ firstly. Then as mentioned in [40], DNMTF uses the $\{0, 1\}$ weighting scheme for constructing neighbor graph above. The data weight matrix is shown as follows:

$$\boldsymbol{W}_{ij}^V = \begin{cases} 1, & \text{if } x_j \in \mathcal{N}(x_i) \\ 0, & \text{otherwise} \end{cases}$$

where $i, j = \{1, \cdots, n\}$, and $\mathcal{N}(x_i)$ refers to the set of k-nearest neighbors of x_i. The graph Laplacian of the data graph is defined as $\boldsymbol{L}_V = \boldsymbol{D}^V - \boldsymbol{W}^V$, where $\boldsymbol{D}^V_{ii} = \sum_j \boldsymbol{W}^V_{ij}$.

Here also constructs a k-nearest neighbor feature graph: $\{f_1, \cdots, f_d\}$, The feature weight matrix is shown as follows:

$$\boldsymbol{W}^U_{ij} = \begin{cases} 1, & \text{if } f_j \in \mathcal{N}(f_i) \\ 0, & \text{otherwise} \end{cases}$$

where $i, j = \{1, \cdots, d\}$, the graph Laplacian of the feature graph: $\boldsymbol{L}_U = \boldsymbol{D}^U - \boldsymbol{W}^U$.

Based on two graph regularizers of data manifold and feature manifold, the objective function is formulated below:

$$\begin{aligned} J_{DNMTF} = &\left\| \boldsymbol{X} - \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \right\|_F^2 + \lambda \mathrm{tr}\left(\boldsymbol{V}^T \boldsymbol{L}_V \boldsymbol{V}\right) \\ &+ \mu \mathrm{tr}\left(\boldsymbol{U}^T \boldsymbol{L}_U \boldsymbol{U}\right), \quad \text{s.t. } \boldsymbol{U} \geq 0, \quad \boldsymbol{S} \geq 0, \ \boldsymbol{V} \geq 0 \end{aligned} \tag{8}$$

where $\lambda, \mu \geq 0$ are the regularization parameters which are used to balance the reconstruction error and graph regularizers.

Shang *et al.* [39] optimized this objective function with respect to one variable while fixing other variables, so that Equation (8) could be rewritten as follows:

$$\begin{aligned} J_{DNMTF} &= \mathrm{tr}\left(\left(\boldsymbol{X} - \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\right)\left(\boldsymbol{X} - \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\right)^T\right) \\ &\quad + \lambda \mathrm{tr}\left(\boldsymbol{V}^T \boldsymbol{L}_V \boldsymbol{V}\right) + \mu \mathrm{tr}\left(\boldsymbol{U}^T \boldsymbol{L}_U \boldsymbol{U}\right) \\ &= \mathrm{tr}\left(\boldsymbol{X}\boldsymbol{X}^T\right) - 2\boldsymbol{tr}\left(\boldsymbol{X}\boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}^T\right) + \mathrm{tr}\left(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}^T\right) \\ &\quad + \lambda \mathrm{tr}\left(\boldsymbol{V}^T \boldsymbol{L}_V \boldsymbol{V}\right) + \mu \mathrm{tr}\left(\boldsymbol{U}^T \boldsymbol{L}_U \boldsymbol{U}\right) \end{aligned} \tag{9}$$

where the constraints $\boldsymbol{U}_{ij}, \boldsymbol{V}_{kj} \geq 0$, and this function could be handled through Lagrange multiplier.

The updating formulas are shown as follows:

$$\boldsymbol{S}_{jl} \leftarrow \boldsymbol{S}_{jl} \frac{[\boldsymbol{U}^T \boldsymbol{X} \boldsymbol{V}]_{jl}}{[\boldsymbol{U}^T \boldsymbol{U} \boldsymbol{S} \boldsymbol{V}^T \boldsymbol{V}]_{jl}} \tag{10}$$

$$\boldsymbol{U}_{ij} \leftarrow \boldsymbol{U}_{ij} \frac{[\boldsymbol{X}\boldsymbol{V}\boldsymbol{S}^T + \mu \boldsymbol{W}^U \boldsymbol{U}]_{ij}}{[\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{S}^T + \mu \boldsymbol{D}^U \boldsymbol{U}]_{ij}} \tag{11}$$

$$\boldsymbol{V}_{kj} \leftarrow \boldsymbol{V}_{kj} \frac{[\boldsymbol{X}^T \boldsymbol{U}\boldsymbol{S} + \lambda \boldsymbol{W}^V \boldsymbol{V}]_{kj}}{[\boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{S} + \lambda \boldsymbol{D}^V \boldsymbol{V}]_{kj}} \tag{12}$$

Shang *et al.* [39] proved the convergence of the updating rules in Equations (10), (11) and (12). For $\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{S} \geq 0$, they proved that the objective function in Equation (8) is non-increasing under the updating rules.

Following [34] and [40], the multiplicative updating rules in Equations (10), (11) and (12) are special cases of gradient decent which could select an automatic step parameter. The iterative multiplicative updating rules converge to a local optimum can be guaranteed. It is worth noting that DNMTF relaxes the orthogonality constraint as nonnegativity, which may deteriorates robustness and performance.

D. PENALIZED NON-NEGATIVE MATRIX TRI-FACTORIZATION

Penalized non-negative matrix tri-factorization (PNMT) [41] presents its advantages in introducing three penalty terms to guarantee the near orthogonality of the clustering indicator matrices, on account of most existing algorithms relaxed the orthogonality constraint as nonnegativity, which may decrease the performance and robustness for NP-completeness of the co-clustering problems.

In detail, co-clustering is formulated as matrix tri-factorization with dual orthogonality constraints, and two indicator matrices are used to present clustering results in two approaches. These two matrices are difficult to optimize because of the orthogonality and nonnegativity of constraints. Pompili *et al.* [42] supported that it is a tough task to design efficient co-clustering algorithms because of the orthogonality constraints. Wang and Huang [41] made use of the penalty terms to approximately solve high-order orthogonality constraints, so that co-clustering is formulated as quadratic non-negative matrix factorization and could be efficient iterate.

According to the problem formulation, for the given the dataset where with d features and n samples could propose the data matrix $\boldsymbol{X} \in \mathbb{R}^{d \times n}$. $\boldsymbol{G} \in \mathbb{R}^{n \times c}$ is used to transform the discrete k-means clustering results, where $\boldsymbol{G}_{ij} = \frac{1}{\sqrt{|C_j|}}$ if sample $\boldsymbol{x}_i$ belongs to cluster C_i and $\boldsymbol{G}_{ij} = 0$ [43]. It is not difficult to find that $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}$ if $\boldsymbol{I} \in \mathbb{R}^{c \times c}$ is a unit matrix. Thus the clustering problem could be approximately formulated as following problem according to the clustering matrix.

$$\begin{aligned} &\min_{\boldsymbol{S},\boldsymbol{G}} \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{S}\boldsymbol{G}^T \right\|_F^2 \\ &\text{s.t. } \boldsymbol{G} \geq 0, \quad \boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I} \end{aligned} \tag{13}$$

where $\boldsymbol{S} \in \mathbb{R}^{d \times c}$ is a coefficient matrix, $\boldsymbol{G} \in \mathbb{R}^{n \times c}$ is a clustering matrix. Approximately, co-clustering problem could be formulated as follows:

$$\begin{aligned} &\min_{\boldsymbol{F},\boldsymbol{S},\boldsymbol{G}} \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{F}\boldsymbol{S}\boldsymbol{G}^T \right\|_F^2 \\ &\text{s.t. } \boldsymbol{F} \geq 0, \quad \boldsymbol{G} \geq 0, \ \boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}, \ \boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I} \end{aligned} \tag{14}$$

where $\boldsymbol{F} \in \mathbb{R}^{d \times c_1}$, $\boldsymbol{G} \in \mathbb{R}^{n \times c_2}$ are indicator matrices, $\boldsymbol{S} \in \mathbb{R}^{c_1 \times c_2}$ is a coefficient matrix. And any matrix $\boldsymbol{X} = (\boldsymbol{X}_{ij})_{d \times n} \in \mathbb{R}^{d \times n}$,

$$\|\boldsymbol{X}\|_F = \left(\sum_{j=1}^{n} \sum_{i=1}^{d} \boldsymbol{X}_{ij}^2 \right)^{\frac{1}{2}}. \tag{15}$$

PNMT introduces three penalties to take the place of $\boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}$ and $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}$, so that the co-clustering problem is transformed into the following function:

$$\begin{aligned} &\min_{\boldsymbol{F},\boldsymbol{S},\boldsymbol{G}} \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{F}\boldsymbol{S}\boldsymbol{G}^T \right\|_F^2 + \frac{\alpha}{2} \mathrm{tr}\left(\boldsymbol{F}\Phi\boldsymbol{F}^T\right) \\ &\qquad + \frac{\beta}{2} \mathrm{tr}\left(\boldsymbol{G}\Psi\boldsymbol{G}^T\right) + \frac{\gamma}{2} \mathrm{tr}\left(\boldsymbol{S}^T\boldsymbol{S}\right) \\ &\text{s.t. } \boldsymbol{F} \geq 0, \quad \boldsymbol{G} \geq 0 \end{aligned} \tag{16}$$

where $\Phi \in \mathbb{R}^{c_1 \times c_1}$, $\Psi \in \mathbb{R}^{c_2 \times c_2}$ are two penalized matrices to guarantee the near orthogonality of $\boldsymbol{F}$ and $\boldsymbol{G}$, $\mathrm{tr}(\boldsymbol{S}^T\boldsymbol{S})$ is used for keeping $\boldsymbol{F}$ and $\boldsymbol{G}$ big enough and α, β, γ are the weights of three terms.

Wang and Huang [41] proposed the penalized matrix as follows:

$$\Phi = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix}. \tag{17}$$

It can be observed that

$$\mathrm{tr}\left(\boldsymbol{F}\Phi\boldsymbol{F}^T\right) = \mathrm{tr}\left(\boldsymbol{F}^T\boldsymbol{F}\Phi\right) = \sum_{i\neq j} \boldsymbol{F}_i^T\boldsymbol{F}_j \tag{18}$$

which could be minimized by an orthogonal vector group $\{F_1, \cdots, F_{c_1}\}$.

Wang and Huang [41] addressed the penalized non-negative matrix factorization through the Lagrange multiplier method and Karush-Kuhn-Tucker (KKT) conditions. The outline of PNMT algorithm is illustrated as follows.

Algorithm 2 Penalized Non-Negative Matrix Tri-Factorization for Co-Clustering (PNMT)

Input: Data matrix $\boldsymbol{X} \in \mathbb{R}^{d\times n}$, the numbers of the clusters c_1, c_2, and parameters Φ, Ψ, α, β, γ.

Output: Two clustering labels $\{C_i\}_{i=1}^{c_1}$ and $\{C'_j\}_{j=1}^{c_2}$.

1: Initialize $\boldsymbol{F}$, $\boldsymbol{S}$ and $\boldsymbol{G}$.
2: **while** not convergent **do**
3: Fix $\boldsymbol{S}$, $\boldsymbol{G}$ update $\boldsymbol{F}$ by
$\boldsymbol{F}_{ij} \leftarrow \boldsymbol{F}_{ij}[\frac{(\boldsymbol{XGS}^T)_{ij}}{(\boldsymbol{FSG}^T\boldsymbol{GS}^T+\alpha\boldsymbol{F}\Phi)_{ij}}]^{\frac{1}{2}}$.
4: Fix $\boldsymbol{F}$, $\boldsymbol{G}$ update $\boldsymbol{S}$ by
$\boldsymbol{S}_{ij} \leftarrow \boldsymbol{S}_{ij}[\frac{(\boldsymbol{F}^T\boldsymbol{XG})_{ij}}{(\boldsymbol{F}^T\boldsymbol{FSG}^T\boldsymbol{G}+\gamma\boldsymbol{S})_{ij}}]^{\frac{1}{2}}$.
5: Fix $\boldsymbol{F}$, $\boldsymbol{S}$ update $\boldsymbol{G}$ by
$\boldsymbol{G}_{ij} \leftarrow \boldsymbol{S}_{ij}[\frac{(\boldsymbol{X}^T\boldsymbol{FS})_{ij}}{(\boldsymbol{GS}^T\boldsymbol{F}^T\boldsymbol{FS}+\beta\boldsymbol{G}\Psi)_{ij}}]^{\frac{1}{2}}$.
6: **end while**
7: Denote the feature space $\{x_{1\cdot}, \cdots, x_{d\cdot}\}$ and the sample space $\{x_{\cdot 1}, \cdots, x_{\cdot n}\}$. Then $x_{i\cdot} \in C_j$ if $\boldsymbol{F}_{ij} = \max_k \boldsymbol{F}_{ik}$, and $x_{\cdot j} \in C'_j$ if $\boldsymbol{G}_{ij} = \max_k \boldsymbol{G}_{ik}$.

E. STRUCTURED OPTIMAL BIPARTITE GRAPH

Structured optimal bipartite graph (SOBG) [44] is a novel co-clustering algorithm to learn a bipartite graph with exactly k (the number of clusters) connected components. Compared to the most existing graph based co-clustering algorithms, they usually describe the feature-sample relations by constructing a bipartite graph and conduct clustering on the graph achieved from the original data matrix which may lead to ambiguous cluster structure. These existing algorithms require a post-processing step such as k-means clustering to obtain the final results, while SOBG was proposed to address this problem. SOBG could be used to learn new bipartite graph which is approximate to the original graph while keeps an explicit cluster structure. This insights realized by imposing constraints on the rank of its Laplacian or normalized Laplacian matrix.

According to the problem formulation, firstly view $\boldsymbol{X}$ as the weight matrix of a bipartite graph (see Figure 1). The green nodes are the d rows of $\boldsymbol{X}$ and the red nodes are the n columns of $\boldsymbol{X}$, and the weight of the i-th blue node to the j-th red node is b_{ij}. An affinity matrix $\boldsymbol{A} \in \mathbb{R}^{n_1\times n_1}$ is shown below:

$$\boldsymbol{A} = \begin{bmatrix} 0 & \boldsymbol{X} \\ \boldsymbol{X}^T & 0 \end{bmatrix} \tag{19}$$

Then a new graph similarity matrix $\boldsymbol{S} \in \mathbb{R}^{n_1\times n_1}$ or $\boldsymbol{P} \in \mathbb{R}^{d\times n}$ were learned as:

$$\boldsymbol{S} = \begin{bmatrix} 0 & \boldsymbol{P} \\ \boldsymbol{P}^T & 0 \end{bmatrix} \tag{20}$$

such that this new graph is more appropriate for clustering problem and could provide clear clustering structure. Here, matrix $\boldsymbol{S}$ that has exact k connected components (Figure 1) can help to obtain the final clustering result directly, without running discretization procedures (such as k-means) as traditional clustering algorithms.

It is noted that the structured optimal graph similarity matrix $\boldsymbol{S}$ should close to the affinity matrix $\boldsymbol{A}$ as possible, so it could be transformed to the following problem:

$$\min_{\boldsymbol{P}\geq 0, \boldsymbol{P}1=1, \boldsymbol{S}\in\Omega} \|\boldsymbol{S}-\boldsymbol{A}\|_F^2 \tag{21}$$

according to Equations (19), (20) and (21), this problem could be rewritten as:

$$\min_{\boldsymbol{P}\geq 0, \boldsymbol{P}1=1, \boldsymbol{S}\in\Omega} \|\boldsymbol{P}-\boldsymbol{X}\|_F^2 \tag{22}$$

while it is difficult to solve the constraint $\boldsymbol{S} \in \Omega$, Nie *et al.* [44] proposed a novel and efficient algorithm for solving this problem.

If the similarity matrix $\boldsymbol{S}$ is a non-negative matrix, then the Laplacian matrix $\boldsymbol{L}_S = \boldsymbol{D}_S - \boldsymbol{S}$ associated with $\boldsymbol{S}$ [38], [45], [46].

Nie *et al.* [44] proved that the multiplicity k of the eigenvalue 0 of the Laplacian matrix $\boldsymbol{L}_S$ is equal to the number of connected components in the graph associated with $\boldsymbol{S}$, so that $\boldsymbol{S} \in \Omega$ could be solved if $\mathrm{rank}(\boldsymbol{L}_S) = n_1 - k$:

$$\min_{\boldsymbol{P}\geq 0, \boldsymbol{P}1=1, \mathrm{rank}(\boldsymbol{L}_S)=n_1-k} \|\boldsymbol{P}-\boldsymbol{X}\|_F^2 \tag{23}$$

then suppose $\sigma_i(\boldsymbol{L}_S)$: the i-th smallest eigenvalue of $\boldsymbol{L}_S$ and $\sigma_i(\boldsymbol{L}_S) \geq 0$. The problem above is transformed to the following problem for a large enough λ:

$$\min_{\boldsymbol{P}\geq 0, \boldsymbol{P}1=1} \|\boldsymbol{P}-\boldsymbol{X}\|_F^2 + \lambda \sum_{i=1}^{k} \sigma_i(\boldsymbol{L}_S) \tag{24}$$

According to the Ky Fan's Theorem [47]:

$$\sum_{i=1}^{k} \sigma_i(\boldsymbol{L}_S) = \min_{\boldsymbol{F}\in\mathbb{R}^{n_1\times k}, \boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}} \mathrm{tr}(\boldsymbol{F}^T\boldsymbol{L}_S\boldsymbol{F}) \tag{25}$$

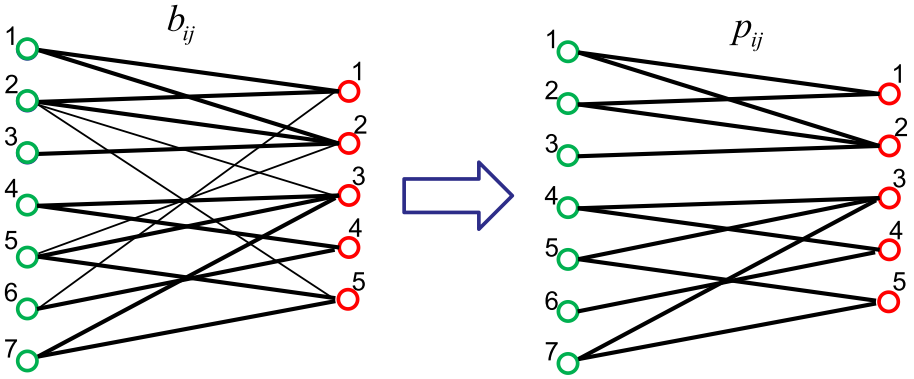


FIGURE 1. Illustration of the structured optimal bipartite graph, the blue green on the left represent features while red nodes on the right show samples. The affinity between the features and samples is denoted by the weight of the corresponding edge.

Therefore, the problem is transformed into the following problem which is much easier to handle:

$$\min_{P,F} \ \|\boldsymbol{P}-\boldsymbol{X}\|_F^2+\lambda \mathrm{tr}(\boldsymbol{F}^T\boldsymbol{L}_S\boldsymbol{F})$$
$$\text{s.t. } \boldsymbol{P}\geq 0, \quad \boldsymbol{P}1=1,\ \boldsymbol{F}\in\mathbb{R}^{n_1\times k},\ \boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I} \quad (26)$$

The algorithm to solve the problem (26) is summarized in Algorithm 3.

Algorithm 3 Algorithm to Solve the Problem (26)

Input: $\boldsymbol{X}\in\mathbb{R}^{d\times n}$, cluster number k, a large enough λ.
Output: $\boldsymbol{P}\in\mathbb{R}^{d\times n}$ and thus $\boldsymbol{S}\in\mathbb{R}^{n_1\times n_1}$ defined in Equation (20) with exact k connected components.

1: Initialize $\boldsymbol{F}\in\mathbb{R}^{n_1\times k}$, which is formed by the k eigenvectors of $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{A}$ corresponding to the k smallest eigenvalues, A is defined in Equation (19).
2: **while** not convergent **do**
3: For each i, update the i-th row of $\boldsymbol{P}$, where the j-th element of v_i is $v_{ij}=\|f_i-f_j\|_2^2$.
4: Update $\boldsymbol{F}$, which is formed by the k eigenvectors of $\boldsymbol{L}_S=\boldsymbol{D}_S-\boldsymbol{S}$ corresponding to the k smallest eigenvalues.
5: **end while**
6: **return** $\boldsymbol{P}$ and $\boldsymbol{S}$.

If the similarity matrix $\boldsymbol{S}$ is non-negative, the normalized Laplacian matrix $\tilde{\boldsymbol{L}_S}=\boldsymbol{I}-\boldsymbol{D}_S^{\frac{1}{2}}\boldsymbol{S}\boldsymbol{D}_S^{-\frac{1}{2}}$ associated with $\boldsymbol{S}$ [38], [45].

Nie *et al.* [44] proved that the multiplicity k of the eigenvalue 0 of the normalized Laplacian matrix $\tilde{\boldsymbol{L}_S}$ is equal to the number of connected components in the graph associated with $\boldsymbol{S}$, so that $\boldsymbol{S}\in\Omega$ could be solved if $\mathrm{rank}(\tilde{\boldsymbol{L}_S})=n_1-k$:

$$\min_{\boldsymbol{P}\geq 0,\boldsymbol{P}1=1,\mathrm{rank}(\tilde{\boldsymbol{L}_S})=n_1-k} \|\boldsymbol{P}-\boldsymbol{X}\|_F^2 \quad (27)$$

Similarly, the problem (27) is equivalent to the following problem with λ:

$$\min_{\boldsymbol{P},\boldsymbol{F}} \ \|\boldsymbol{P}-\boldsymbol{X}\|_F^2+\lambda \mathrm{tr}(\boldsymbol{F}^T\tilde{\boldsymbol{L}_S}\boldsymbol{F})$$
$$\text{s.t. } \boldsymbol{P}\geq 0, \quad \boldsymbol{P}1=1,\ \boldsymbol{F}\in\mathbb{R}^{n_1\times k},\ \boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I} \quad (28)$$

The algorithm to solve the problem (28) is summarized in Algorithm 4. It could only update the m nearest similarities for each data point and thus the complexity of updating $\boldsymbol{P}$ or $\boldsymbol{F}$ can be decreased dramatically.

Algorithm 4 Algorithm to Solve the Problem (28)

Input: $\boldsymbol{X}\in\mathbb{R}^{d\times n}$, cluster number k, a large enough λ.
Output: $\boldsymbol{P}\in\mathbb{R}^{d\times n}$ and thus $\boldsymbol{S}\in\mathbb{R}^{n_1\times n_1}$ defined in Equation (20) with exact k connected components.

1: Initialize $\boldsymbol{F}\in\mathbb{R}^{n_1\times k}$, which is formed by the k eigenvectors of $\tilde{\boldsymbol{L}_S}=\boldsymbol{I}-\boldsymbol{D}_S^{\frac{1}{2}}\boldsymbol{S}\boldsymbol{D}_S^{-\frac{1}{2}}$ corresponding to the k smallest eigenvalues, $\boldsymbol{A}$ is defined in Equation (19).
2: **while** not convergent **do**
3: For each i, update the i-th row of $\boldsymbol{P}$, where the j-th element of v_i is $v_{ij}=\left\|\frac{f_i}{\sqrt{d_i}}-\frac{f_j}{\sqrt{d_j}}\right\|_2^2$.
4: Update $\boldsymbol{F}=\begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{V}\end{bmatrix}$, where U and V are the leading k left and right singular vectors of $\tilde{\boldsymbol{S}}=\boldsymbol{D}_{S_u}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{D}_{S_v}^{-\frac{1}{2}}$ respectively and $\boldsymbol{D}_S=\begin{bmatrix}\boldsymbol{D}_{S_u} & \\ & \boldsymbol{D}_{S_v}\end{bmatrix}$.
5: **end while**
6: **return** $\boldsymbol{P}$ and $\boldsymbol{S}$.

F. FAST NON-NEGATIVE MATRIX TRI-FACTORIZATION

Fast non-negative matrix tri-factorization (FNMTF) [48] algorithm could conduct co-clustering on macroscale data efficiently. FNMTF constraints the factor matrices of NMTF with cluster indicator matrices, this process is a special type of non-negative matrices. Due to this advancement, the clustering results are easily deposited in the resulted factor matrices. Moreover, the optimization problems could be solved with much less matrix multiplications which are benefit from the property of indicator matrices. In summary, this algorithm is superior to other algorithms in computing efficiency and the scale of data.

The process of problem formalization is similar to the DRCC [17], It is noted that $\boldsymbol{F}$, $\boldsymbol{G}$ is cluster indicator matrices, each row of them has one and only one element equal to 1 to indicate the cluster membership, while the rest are 0. Here, given a set of all cluster indicator matrices: Ψ.

NMTF constraints the factor matrices of NMTF with cluster indicator matrices and minimize the objective function shown below:

$$J_{FNMTF} = \left\| \boldsymbol{X} - \boldsymbol{F}\boldsymbol{S}\boldsymbol{G}^T \right\|_F^2 \quad \text{s.t. } \boldsymbol{F} \in \Psi^{d\times m}, \boldsymbol{G} \in \Psi^{n\times c}. \tag{29}$$

It is worth noting that the orthonormal constraints on $\boldsymbol{F}$ and $\boldsymbol{G}$ are disappeared in this objective.

In the optimization procedures, Wang *et al.* [48] fix $\boldsymbol{F}$ and $\boldsymbol{G}$ and setting the derivative $\boldsymbol{S}$ as 0 to solve the three variables in Equation (29), thus:

$$\boldsymbol{S} = \left(\boldsymbol{F}^T\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{X}\boldsymbol{G}\left(\boldsymbol{G}^T\boldsymbol{G}\right)^{-1}. \tag{30}$$

Secondly, Wang *et al.* [48] fix $\boldsymbol{F}$ and $\boldsymbol{S}$ to obtain $\boldsymbol{G}$ to rewrite the problem with each $i(1 \le i \le n)$:

$$\min_{\boldsymbol{G}\in\Psi} \left\| x_i - \boldsymbol{F}\boldsymbol{S}_{g_i}^T \right\|_F^2. \tag{31}$$

Because $g_i\,(1 \le i \le n) \in \Psi^{1\times c}$ is a cluster indicator vector, so that:

$$g_{ij} = \begin{cases} 1 & j = \arg\min_k \left\| x_i - \tilde{f}_k \right\|_F^2, \\ 0 & otherwise, \end{cases} \tag{32}$$

where $\tilde{\boldsymbol{F}} = \boldsymbol{F}\boldsymbol{S}$ and $\tilde{f}_k$ is the k-th columm of $\tilde{\boldsymbol{F}}$. This equation could enumerate the c vector norms and find out the maximum one, without involving any matrix multiplication.

Finally, they fix $\boldsymbol{G}$ and $\boldsymbol{S}$ to obtain $\boldsymbol{F}$ to rewrite the problem with each $i(1 \le i \le n)$:

$$\min_{\boldsymbol{F}\in\Psi} \left\| x_i - f_j\boldsymbol{S}\boldsymbol{G}^T \right\|_F^2. \tag{33}$$

Because $f_j\,(1 \le i \le d) \in \Psi^{1\times m}$ is a cluster indicator vector, so that:

$$f_{ij} = \begin{cases} 1 & i = \arg\min_l \left\| x_j - \tilde{g}_l \right\|_F^2, \\ 0 & otherwise, \end{cases} \tag{34}$$

where $\tilde{\boldsymbol{G}}^T = \boldsymbol{S}\boldsymbol{G}^T$ and $\tilde{g}_l$ is the l-th row of $\tilde{\boldsymbol{G}}^T$.

The procedures to solve Equation (29) are summarized in Algorithm 5.

G. BILATERAL K-MEANS ALGORITHM

Bilateral k-means algorithm (BKM) [49] algorithm is different from traditional k-means algorithms, it has two indicator matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ and a diagonal matrix $\boldsymbol{S}$ to be handled, which represents the cluster memberships of data and features and the co-cluster centers, respectively.

BKM relaxes the minimum normalized cuts problem to a special NMF with indicator matrices constraints problem.

Algorithm 5 Fast Non-Negative Matrix Tri-Factorization

Input: Data matrix $\boldsymbol{X} \in \mathbb{R}^{d\times n}$.

Output: Indicator matrices $\boldsymbol{G} \in \Psi^{n\times c}$ for data point clustering and $\boldsymbol{F} \in \Psi^{d\times m}$ for feature clustering.

1: Initialize $\boldsymbol{G}$ and $\boldsymbol{F}$ with arbitrary class indicator matrices;
2: **while** not convergent **do**
3: calculate $\boldsymbol{S}$ by Equation (30);
4: calculate $\boldsymbol{G}$ by Equation (32);
5: calculate $\boldsymbol{F}$ by Equation (34);
6: **end while**
7: **return** $\boldsymbol{G}$ and $\boldsymbol{F}$.

These indicators maintain the clustering results. Whereas, it needs a post-processing steps to output the final result. Han *et al.* [49] handled the optimization problem of BKM through decomposing this problem into three subproblems and solved in an alternative way.

The process of problem formalization is similar to the FNMTF [48]. If i-th feature $x_{i\text{ą}d}$ belongs to cluster C'_j, $g_{ij} = 1$, and If i-th sample $x_{\text{ą}di}$ belongs to cluster C_j, $f_{ij} = 1$. Thus, $\boldsymbol{G}$ and $\boldsymbol{F}$ represent to indicator matrices and $\boldsymbol{G} \in \Phi^{d\times c}$ and $\boldsymbol{F} \in \Phi^{n\times c}$.

BKM replaces the bipartitioning normalized cuts in BSGP [35] by multipartitioning normalized cuts, the objective function is indicated as follows:

$$\min_Y \sum_{k=1}^{c} \frac{\boldsymbol{y}_k^T\boldsymbol{L}\boldsymbol{y}_k}{\boldsymbol{y}_k^T\boldsymbol{D}\boldsymbol{y}_k} \quad \text{s.t. } \boldsymbol{Y} \in \Phi^{(d+n)\times m} \tag{35}$$

where $\boldsymbol{D}$ is the diagonal 'degree' matrix with $\boldsymbol{D}_{ii} = \sum_k \boldsymbol{A}_{ik}$, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, $\boldsymbol{y} = [\boldsymbol{G}^T, \boldsymbol{F}^T]^T$.

Because there is only one non-zero element in each row of $\boldsymbol{Y}$, and $\boldsymbol{D}$ is a diagonal matrix, the matrix $\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}$ is a diagonal matrix with (k,k)-element equal to $\boldsymbol{y}_k^T\boldsymbol{D}\boldsymbol{y}_k$. Thus, Equation (35) could be rewritten as below:

$$\min_Y \operatorname{tr}\left(\boldsymbol{Y}^T\boldsymbol{L}\boldsymbol{Y}\left(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}\right)^{-1}\right) \quad \text{s.t. } \boldsymbol{Y} \in \Phi^{(m+n)\times c} \tag{36}$$

Substituting $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ could be used in Equation (36) as:

$$\operatorname{tr}\left(\boldsymbol{Y}^T\boldsymbol{L}\boldsymbol{Y}\left(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}\right)^{-1}\right) = \operatorname{tr}\left(\boldsymbol{I} - \boldsymbol{Y}^T\boldsymbol{A}\boldsymbol{Y}\left(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}\right)^{-1}\right) \tag{37}$$

where $\boldsymbol{I}$ is an identity matrix.

Because the indicator matrix $\boldsymbol{Y}$ can be rewritten as $\boldsymbol{Y}^T = [\boldsymbol{G}^T, \boldsymbol{F}^T]$, the objective function could be transformed as follows:

$$\min_{\boldsymbol{F},\boldsymbol{G}} \operatorname{tr}\left(-\boldsymbol{F}^T\boldsymbol{X}\boldsymbol{F}\left(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}\right)^{-1}\right) \quad \text{s.t. } \boldsymbol{G} \in \Phi^{d\times c}, \quad \boldsymbol{F} \in \Phi^{n\times c}. \tag{38}$$

The objective in Equation (38) is a NP-complete problem [50]. Han *et al.* [49] relax this optimization problem

into a matrix decomposition problem, append two terms: $\mathrm{tr}\left(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}^{-1}\boldsymbol{G}^T\boldsymbol{G}\left(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}\right)^{-1}\boldsymbol{F}^T\boldsymbol{F}\right)$ and $\mathrm{tr}\left(\boldsymbol{X}^T\boldsymbol{X}\right)$. The optimization problem is indicated as follows:

$$\begin{aligned}&\min_{\boldsymbol{F},\boldsymbol{G}}\left\|\boldsymbol{X}-\boldsymbol{G}\left(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}\right)^{-1}\boldsymbol{F}^T\right\|_F^2\\&\text{s.t. }\boldsymbol{G}\in\Phi^{d\times c},\quad \boldsymbol{F}\in\Phi^{n\times c}.\end{aligned}\tag{39}$$

Finally, due to $\left(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}\right)^{-1}$ is a diagonal matrix could be replaced by a matrix $\boldsymbol{S}$ which could be considered as a parameter. The optimization problem of bilateral k-means algorithm is demonstrated below:

$$\begin{aligned}&\min_{\boldsymbol{F},\boldsymbol{G},\boldsymbol{S}}\left\|\boldsymbol{X}-\boldsymbol{G}\boldsymbol{S}\boldsymbol{F}^T\right\|_F^2\\&\text{s.t. }\boldsymbol{G}\in\Phi^{d\times c},\quad \boldsymbol{F}\in\Phi^{n\times c},\ \boldsymbol{S}\in diag.\end{aligned}\tag{40}$$

where *diag* represents the set of diagonal matrices.

The procedures of solving the model of BKM Equation (40) are summarized as follows:

Algorithm 6 Algorithm to Solve the Problem (40)

Input: Data matrix $\boldsymbol{X}\in\mathbb{R}^{d\times n}$.

Output: Indicator matrices $\boldsymbol{F}$ for sample clustering and $\boldsymbol{G}$ for feature clustering.

1: Initialize $\boldsymbol{G}$ $\boldsymbol{F}$ with arbitrary class indicator matrices.
2: **while** not convergent **do**
3: Calculating $\boldsymbol{S}$ by $\boldsymbol{s}=\boldsymbol{H}^{-1}\boldsymbol{r}$, here $\boldsymbol{s}$ is used to denote $f(\boldsymbol{S})$, and $\boldsymbol{r}$ to denote $f(\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{Q})$, $\boldsymbol{H}$ is a diagonal matrix, and $\boldsymbol{H}^{-1}$ is easily to be solved.
4: Calculating $\boldsymbol{G}$ by
$$p_{ij}=\begin{cases}1, & j=\arg\min_k\left\|x_{j\cdot}-l_{k\cdot}\right\|^2\\0, & \text{otherwise}\end{cases}$$
where $\boldsymbol{L}=\boldsymbol{S}\boldsymbol{G}^T$, $l_{k\cdot}$ is the k-th row of $\boldsymbol{L}$.
5: Calculating $\boldsymbol{F}$ by
$$q_{ij}=\begin{cases}1, & j=\arg\min_k\left\|x_{\cdot i}-r_{\cdot k}\right\|^2\\0, & \text{otherwise}\end{cases}$$
where $\boldsymbol{R}=\boldsymbol{F}\boldsymbol{S}$, $r_{\cdot k}$ is the k-th column of $\boldsymbol{R}$.
6: **end while**
7: **return** $\boldsymbol{G}$ and $\boldsymbol{F}$.

III. EXPERIMENTAL ANALYSIS

In this section, comprehensive experiments are conducted to evaluate the performance of the described algorithms. we concentrate on the evaluation metrics and datasets firstly. Then we demonstrate the visualization based on a set of points constructed in two moons pattern. Finally, the co-clustering effectiveness and efficiency of the described algorithms would be given and analyzed.

A. DATASETS

In our experiments, we choose eight machine learning datasets to evaluate the performance of the described algorithms. These datasets derive from diverse fields, which facilitate to make the experiments more comprehensive.

Coil20 dataset. This dataset contains 32 × 32 gray scale images of 20 objects and each object includes 72 images.

CalTech 101 silhouettes dataset. This dataset is based on the CalTech 101 image annotations and each of the 101 classes has at most 100 training instances. The minimum quantity of training instances is around 20 per class.

MNIST database. The MNIST database of handwritten digits from Yann LeCun's page has a training set of 60,000 examples and a 10,000 examples test set.

ISOLET database. This dataset contains 150 subjects who spoke the name of each letter of the alphabet twice. The speakers are grouped into sets of 30 speakers.

Here, we also use the USPS handwritten image database and two text datasets such as BASEHOCK dataset and PCMAC dataset. Meanwhile UCI Statlog dataset is also used to evaluate these algorithms. The detailed information of datasets is summarized in Table 1.

TABLE 1. Description of real world datasets.

Datasets	# sample	# feature	# classes
BASEHOCK	1993	4862	2
Caltech101-silhouettes	8641	256	101
Coil20	1440	1024	20
ISOLET	1560	617	26
MNIST	4000	784	10
PCMAC	1943	3289	2
Statlog	1000	20	2
USPS	9298	256	10

B. EVALUATION METRIC

In order to evaluate the clustering results of diverse clustering algorithms, we adopt four evaluation metrics used in [34], [51], and [52]. As the standard measures, these metrics are widely used for clustering [17]. These evaluation metrics including clustering accuracy (ACC), normalized mutual information (NMI), adjusted rand index (ARI) and purity.

- **Clustering accuracy** (ACC) describes the relationship between clusters and truth label. It measures the degree about each cluster contains sample data from the matching class. Given a sample $x_i\in\{x_i\}_{i=1}^n$, p_i denotes the true class label and q_i denotes the prediction clustering label. The ACC is defined as follows:
$$ACC=\frac{\sum_{i=1}^n\delta\left(p_i,map\left(q_i\right)\right)}{n}\tag{41}$$
where $\delta(a,b)$ equals one if $a=b$ and equals zero otherwise. And $map(\circ)$ is the best permutation mapping function such as the Kuhn-Munkres algorithm [53] that matches the prediction clustering label to the true label. The larger value of the ACC is, the better clustering performs.
- **Normalized mutual information** (NMI) is used for measuring the quality of clusters. Given two random variables p and q, NMI is defined as follows:
$$NMI\left(p,q\right)=\frac{I\left(p;q\right)}{\sqrt{H\left(p\right)H\left(q\right)}}\tag{42}$$

where $I(\circ)$ is the mutual information of input data and $H(\circ)$ denotes the entropies. The clustering result $\tilde{C} = \{\tilde{C}_i\}_{i=1}^{\tilde{C}}$ based on the true labels $C = \{C_j\}_{j=1}^{C}$ for all sample data. NMI is rewritten as:

$$NMI\left(C, \tilde{C}\right) = \frac{\sum_{i=1}^{\tilde{C}} \sum_{j=1}^{C} \left|\tilde{C}_i \cap C_j\right| \log \frac{n\left|\tilde{C}_i \cap C_j\right|}{\left|\tilde{C}_i\right|\left|C_j\right|}}{\sqrt{\left(\sum_{i=1}^{\tilde{C}} \left|\tilde{C}_i\right| \log \frac{\left|\tilde{C}_i\right|}{n}\right)\left(\sum_{j=1}^{C} \left|C_j\right| \log \frac{\left|C_j\right|}{n}\right)}} \tag{43}$$

It is worth noting that $\tilde{C}$ and C are not necessarily equal and the larger NMI becomes, the better clustering performs.

- **Adjusted rand index** (ARI) is defined as the number of couples of objects that are both located in the same cluster and the same class or different otherwise and is divided by the total number of objects classes [54], [55]. The ARI is defined as follows:

$$ARI = \frac{a - \frac{bc}{n(n-1)/2}}{(1/2)(b+c) - \frac{bc}{n(n-1)/2}}, \tag{44}$$

where $a = \sum_{i,j} \frac{V_{ij}(V_{ij}-1)}{2}$, $b = \sum_{i} \frac{V_i(V_i-1)}{2}$ and $c = \sum_{j} \frac{V_j(V_j-1)}{2}$. The number of objects that are in both of class i and cluster j is expressed by V_{ij}. Meanwhile, V_i, V_j could be regarded as the quantity of objects in the class i and cluster j, respectively. The larger value of ARI, the more resemblant to the labels clustering results.

- **Purity** measures the degree of each cluster containing data points from one class [56]. The cluster purity value is measured by:

$$Purity = \sum_{i=1}^{k} \frac{n_i}{n} P(S_i), P(S_i) = \frac{1}{n_i} \max_j P\left(n_i^j\right) \tag{45}$$

where S_i is a particular cluster with size n_i and n_i^j denotes the data of i-th input class which is expect to distribute to j-th cluster. k is the number of clusters and n is the total number of sample data. It is noted that the larger values of purity represent the better performance.

There are some parameters to be decided in advance. Here we also take into consideration the sensitivity of initial values of most clustering algorithms, so that experiments are iterated 20 times and took their mean and standard deviation as the results. For co-clustering algorithms, the quantity of clusters to divide samples is adjusted to that of clusters to converge features. And for all involved clustering algorithms in experiments, the quantity of clusters is set from the label information of classes provided in datasets. As to diverse described clustering algorithms, their parameters are fixed as default values. Numbers nearest neighbors for sample graph and feature graph in DRCC and DNMTF which are learned in manifold is set as 11. Meanwhile the weight mode is fixed as binary values. The regularization parameters of DRCC are tuned as $\lambda = 1$ and $\mu = 1$, DNMTF are set as $\lambda = 200$ and $\mu = 200$ and PNMT are fixed as $\alpha = 1$, $\beta = 1$ and $\gamma = 1$, respectively. For the SOBG, the number of neighbors is set to be 5, the value of $\lambda = 1$ and σ is self-tuned.

C. VISUALIZATION

For the process of visualization, it aims to present the performance of these described algorithms. Given a set of point data constructed in two moons pattern, this set contains two natural clusters: the upper and the lower moon, as shown in Figure 2(a). We conduct the clustering through the described algorithms based on this set, respectively, and demonstrate the clustering results. We aim to provide brief insights and exhibition of these algorithms.

As we can observe, for this set the clustering result given by DRCC 2(b) is well separated than others. While the clustering results of PNMT 2(d) and FNMTF 2(e) are similar. It seems that the clustering results given by DNMTF 2(c) and BKM 2(f) are not so good as the dataset with low dimension and low number of labels. This process conduct the visualization of described algorithms, the comprehensive experiments would be conducted to evaluate the performance based on the evaluation metrics mentioned above.

D. RESULT AND ANALYSIS

This section aims to evaluate the performance of described algorithms based on the evaluation metrics above. The clustering accuracy, normalized mutual information, adjusted

TABLE 2. Clustering accuracy (mean% ± std%) of diverse co-clustering algorithms on different datasets. The higher the better.

Data sets	BASEHOCK	Caltech101	Coil20	ISOLET	MNIST	PCMAC	Statlog	USPS
NMF	54.3 ± 0.6	24.8 ± 1.2	52.4 ± 4.6	50.9 ± 0.9	48.2 ± 3.2	47.6 ± 0.3	67.5 ± 1.2	25.1 ± 3.5
k-means	62.3 ± 1.9	28.9 ± 0.8	52.3 ± 6.4	56.4 ± 0.7	45.7 ± 2.9	50.6 ± 0.1	67.1 ± 0.1	53.6 ± 4.2
DRCC	61.6 ± 2.2	29.1 ± 0.6	56.9 ± 5.6	57.9 ± 2.3	50.4 ± 1.3	50.5 ± 0.1	67.1 ± 0.1	54.3 ± 5.1
DNMTF	54.3 ± 0.2	21.1 ± 0.5	77.3 ± 1.9	58.1 ± 5.8	40.6 ± 2.6	50.3 ± 0.3	56.5 ± 3.4	53.6 ± 4.2
PNMT	64.7 ± 1.9	29.4 ± 1.3	55.1 ± 3.6	57.3 ± 0.2	46.8 ± 1.9	55.3 ± 4.2	67.1 ± 0.1	53.9 ± 3.6
SOBG	49.6 ± 0.1	9.7 ± 0.1	10.4 ± 0.1	10.8 ± 0.1	11.5 ± 0.1	50.6 ± 0.1	70.1 ± 0.1	18.2 ± 0.1
FNMTF	51.7 ± 0.1	28.1 ± 0.6	34.2 ± 3.1	59.7 ± 0.1	43.7 ± 3.3	50.5 ± 0.1	67.1 ± 0.1	40.2 ± 3.7
BKM	52.1 ± 1.5	9.8 ± 0.2	5.2 ± 0.7	53.6 ± 0.2	11.5 ± 0.5	50.6 ± 0.3	58.4 ± 0.1	24.9 ± 0.4

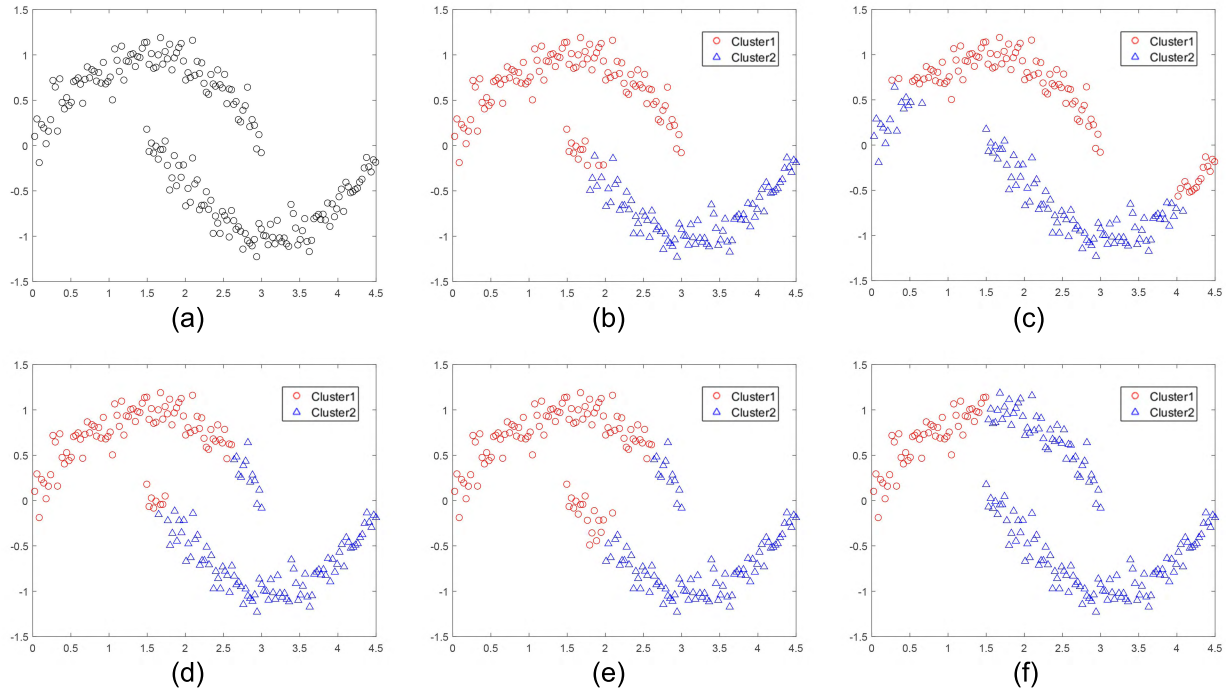


FIGURE 2. Clustering on the two moons pattern. (a) original dataset. (b) Clustering results given by DRCC. (c) Clustering results given by DNMTF. (d) Clustering results given by PNMT. (e) Clustering results given by FNMTF. (f) Clustering results given by BKM.

TABLE 3. Normalized mutual information (mean% ± std%) of diverse co-clustering algorithms on different datasets. The higher the better.

Data sets	BASEHOCK	Caltech101	Coil20	ISOLET	MNIST	PCMAC	Statlog	USPS
NMF	6.1 ± 0.4	46.3 ± 0.6	69.7 ± 2.1	0.1 ± 0.1	43.1 ± 1.5	3.6 ± 1.8	0.7 ± 0.5	11.2 ± 4.4
k-means	4.7 ± 1.3	52.9 ± 0.5	69.8 ± 3.2	1.2 ± 0.1	48.2 ± 1.8	3.6 ± 1.9	1.1 ± 0.1	60.4 ± 2.5
DRCC	4.3 ± 1.1	52.9 ± 0.3	72.1 ± 2.9	73.8 ± 1.1	45.2 ± 1.5	2.5 ± 1.6	1.1 ± 0.1	51.3 ± 1.7
DNMTF	3.2 ± 0.1	44.7 ± 0.4	88.5 ± 0.6	5.1 ± 4.8	45.7 ± 1.8	0.2 ± 0.1	0.1 ± 0.1	60.4 ± 2.5
PNMT	6.5 ± 1.9	53.2 ± 0.7	69.2 ± 0.2	1.6 ± 0.7	39.8 ± 1.9	5.5 ± 3.1	1.1 ± 0.1	46.3 ± 2.2
SOBG	1.7 ± 0.1	1.5 ± 0.1	8.9 ± 0.1	0.9 ± 0.1	0.2 ± 0.1	0.1 ± 0.1	0.2 ± 0.1	3.1 ± 0.1
FNMTF	0.1 ± 0.1	45.1 ± 0.8	46.7 ± 2.9	2.8 ± 0.1	38.3 ± 2.8	56.2 ± 4.4	1.1 ± 0.1	37.4 ± 1.4
BKM	0.2 ± 0.1	1.6 ± 0.5	0.3 ± 1.1	0.5 ± 0.1	0.2 ± 0.5	0.3 ± 0.2	0.6 ± 0.1	11.2 ± 0.7

TABLE 4. Adjusted rand index (mean% ± std%) of diverse co-clustering algorithms on different datasets. The higher the better.

Data sets	BASEHOCK	Caltech101	Coil20	ISOLET	MNIST	PCMAC	Statlog	USPS
NMF	8.2 ± 0.7	25.7 ± 2.4	48.3 ± 5.2	1.6 ± 0.1	34.9 ± 2.2	4.4 ± 3.2	5.2 ± 0.1	44.4 ± 3.9
k-means	6.2 ± 1.8	25.7 ± 2.4	48.3 ± 5.2	1.6 ± 0.1	34.9 ± 2.2	4.4 ± 3.1	5.2 ± 0.1	44.4 ± 3.9
DRCC	5.5 ± 1.5	26.1 ± 1.4	51.4 ± 5.6	51.6 ± 1.4	32.5 ± 1.4	29.1 ± 2.5	5.2 ± 0.1	39.9 ± 3.6
DNMTF	0.7 ± 0.1	16.9 ± 0.9	73.7 ± 1.2	3.7 ± 4.7	28.1 ± 1.7	0.2 ± 0.1	0.2 ± 1.1	44.4 ± 3.9
PNMT	8.7 ± 2.6	26.8 ± 3.2	48.2 ± 3.8	2.2 ± 0.9	28.1 ± 1.5	7.1 ± 4.2	5.2 ± 0.1	36.2 ± 3.2
SOBG	0.1 ± 0.1	0.1 ± 0.1	0.6 ± 0.1	0.1 ± 0.1	0.1 ± 0.1	0.1 ± 0.1	0.3 ± 0.1	0.4 ± 0.1
FNMTF	0.1 ± 0.1	31.4 ± 1.4	27.6 ± 3.5	3.7 ± 0.1	25.5 ± 3.1	39.6 ± 5.7	5.2 ± 0.1	25.9 ± 2.2
BKM	0.2 ± 0.1	0.2 ± 0.1	0.1 ± 0.2	0.5 ± 0.1	0.1 ± 0.1	0.2 ± 0.2	1.9 ± 0.1	0.3 ± 0.2

rand index and purity are listed in Tables 2, 3, 4 and 5 respectively. Meanwhile, The described algorithms also are compared with single-way clustering algorithms: k-means and NMF which are regarded as the baselines. From the results shown in these tables, some interesting observations emerged this research. On one hand, the most proposed co-clustering algorithms comes with better performance than the compared single-way clustering algorithms in most mentioned datasets.

TABLE 5. Purity (mean% ± std%) of diverse co-clustering algorithms on different datasets. The higher the better.

Data sets	BASEHOCK	Caltech101	Coil20	ISOLET	MNIST	PCMAC	Statlog	USPS
NMF	54.3 ± 0.6	38.5 ± 0.6	58.2 ± 4.1	50.9 ± 0.9	53.5 ± 2.2	57.6 ± 0.3	70.1 ± 0.1	28.1 ± 3.6
k-means	52.3 ± 1.9	46.2 ± 0.4	57.1 ± 5.3	56.4 ± 0.1	53.6 ± 2.8	50.6 ± 0.1	70.1 ± 0.1	62.9 ± 3.7
DRCC	61.6 ± 2.2	46.1 ± 0.3	60.2 ± 4.7	62.2 ± 1.8	54.8 ± 1.4	50.6 ± 0.1	70.1 ± 0.1	61.4 ± 2.8
DNMTF	54.3 ± 0.1	36.7 ± 0.4	82.6 ± 1.1	58.1 ± 5.8	51.2 ± 2.3	50.6 ± 0.1	70.0 ± 0.1	62.9 ± 3.7
PNMT	64.7 ± 1.9	46.4 ± 0.7	58.6 ± 3.4	57.4 ± 1.8	50.9 ± 1.9	55.4 ± 4.1	70.0 ± 0.1	57.9 ± 3.7
SOBG	51.1 ± 0.1	10.5 ± 0.1	12.9 ± 0.1	11.0 ± 0.1	11.6 ± 0.1	50.6 ± 0.1	70.1 ± 0.1	19.1 ± 0.1
FNMTF	51.7 ± 0.1	36.7 ± 0.7	34.3 ± 3.1	59.7 ± 0.1	47.6 ± 3.3	50.5 ± 0.1	70.0 ± 0.1	49.1 ± 2.3
BKM	52.1 ± 1.5	9.8 ± 0.2	5.3 ± 0.7	53.6 ± 0.2	11.5 ± 0.5	50.6 ± 0.1	70.0 ± 0.1	26.8 ± 0.5

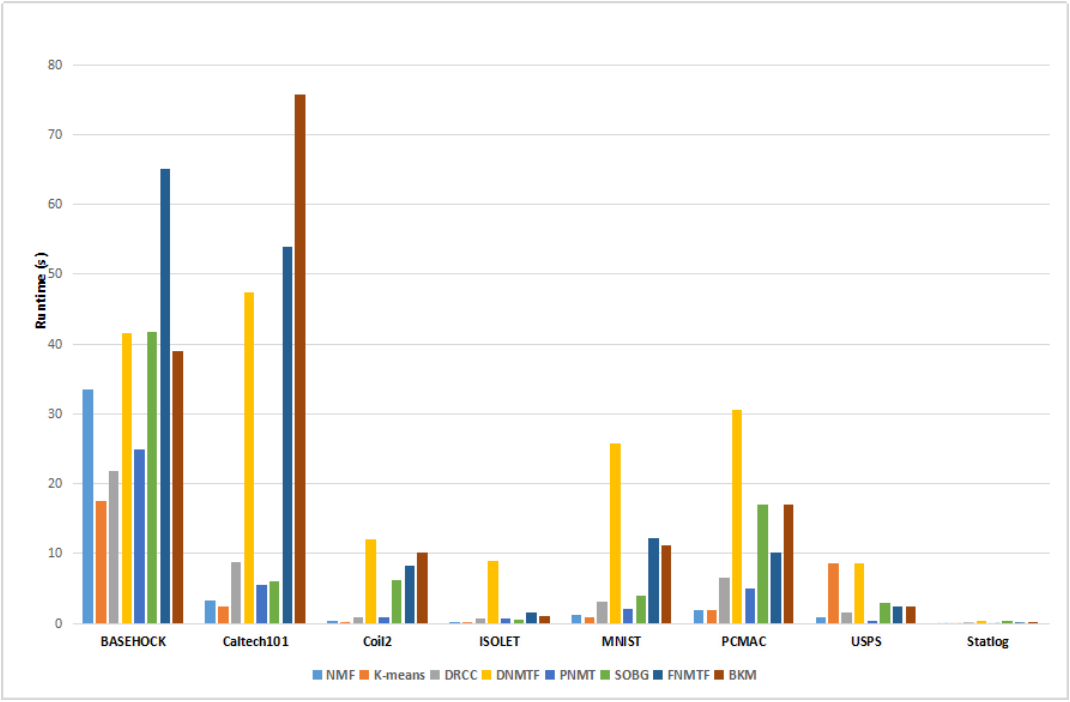


FIGURE 3. Runtime of diverse clustering algorithms in all tested datasets.

However, single-way clustering algorithms also have advantages compared with co-clustering algorithms, which may be resulted by the fact of the uncertainty class label information and the quantity of clusters is treated as a latent factor. We set the quantity of clusters for features equal to the quantity of clusters for samples which may influence the clustering performance to solve the uncertainty class label information problem. On the other hand, the co-clustering algorithms DNMTF and PNMT perform better than other described algorithms in most tested datasets. It seems that SOBG and FNMTF show superiority over other algorithms in small datasets such as Statlog..

The runtime list of diverse clustering algorithms is shown in Figure 3. We have the following observations from this figure. With the increasing of sample data and features, different clustering algorithms demonstrate varying efficiency. For example, described algorithms perform fast in the dataset Statlog which contains less samples and features, while slow in the dataset BASEHOCK. Meanwhile DNMTF and FNMTF perform slower than other described algorithms in most tested datasets. It seems to be true that the runtime of described co-clustering algorithms is positively related to the number of samples and features.

IV. CONCLUSION AND FURTHER WORK

In this paper, we revisited six existing co-clustering algorithms. Then we conducted and compared them through existing evaluation metrics. We concentrated the co-clustering ability of these algorithms, through conducting comprehensive experiments in a set of datasets to measure the results to compare the performance of these algorithms. Meanwhile, we analyzed the results to explore the relationship between described algorithms and tested datasets in runtime and clustering ability.

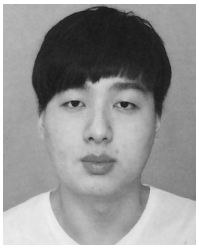
This paper aims to provide insights and advice for selecting the co-clustering algorithms in diverse field such as text

mining, recommendation systems and gene expression. In the future, co-clustering algorithms will be improved on speed and clustering ability. With the rapid development of this field co-clustering will become much better and we will explore more efficient algorithms via matrix factorization to address more general co-clustering problems in our future work.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.
[2] F. Shang, L. C. Jiao, J. Shi, F. Wang, and M. Gong, "Fast affinity propagation clustering: A multilevel approach," *Pattern Recognit.*, vol. 45, no. 1, pp. 474–486, 2012.
[3] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.
[4] E. Izquierdo-Verdiguier, R. Jenssen, L. Gómez-Chova, and G. Camps-Valls, "Spectral clustering with the probabilistic cluster kernel," *Neurocomputing*, vol. 149, no. 5, pp. 1299–1304, 2015.
[5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[6] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Proc. Int. Conf. Data Mining*, 2001, pp. 107–114.
[7] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.
[8] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Inf. Process. Manage.*, vol. 42, no. 2, pp. 373–386, 2006.
[9] M. Bendechache, A. K. Tari, and M. T. Kechadi, "Parallel and distributed clustering framework for big spatial data mining," in *Proc. Int. J. Parallel, Emergent Distrib. Syst.*, 2018, pp. 1–19.
[10] M. Gong, Y. Liang, J. Shi, W. Ma, and J. Ma, "Fuzzy C-means clustering with local information and kernel metric for image segmentation," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 573–584, Feb. 2013.
[11] X. Lu, H. Wu, and Y. Yuan, "Double constrained NMF for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2746–2758, May 2014.
[12] M. Ailem, F. Role, and M. Nadif, "Co-clustering document-term matrices by direct maximization of graph modularity," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 1807–1810.
[13] F. Nie, D. Xu, I.-W. Tsang, and C. Zhang, "Spectral embedded clustering," in *Proc. Int. Jont Conf. Artif. Intell.*, 2009, pp. 1181–1186.
[14] N. Van Pham, L. T. Pham, T. D. Nguyen, and L. T. Ngo, "A new cluster tendency assessment method for fuzzy co-clustering in hyperspectral image analysis," *Neurocomputing*, vol. 307, pp. 213–226, Sep. 2018.
[15] D. Hu, F. Nie, and X. Li. (2018). "Deep co-clustering for unsupervised audiovisual learning." [Online]. Available: https://arxiv.org/abs/1807.03094
[16] J. Jacques and C. Biernacki, "Model-based co-clustering for ordinal data," *Comput. Statist. Data Anal.*, vol. 123, pp. 101–115, Jul. 2018.
[17] Q. Gu and J. Zhou, "Co-clustering on manifolds," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 359–368.
[18] S. Wang, W. Pedrycz, Q. Zhu, and W. Zhu, "Subspace learning for unsupervised feature selection via matrix factorization," *Pattern Recognit.*, vol. 48, no. 1, pp. 10–19, 2015.
[19] S. Wang and W. Zhu, "Sparse graph embedding unsupervised feature selection," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 3, pp. 329–341, Mar. 2018.
[20] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1944–1957, Nov. 2007.
[21] H. Zha, X. He, C. Ding, H. D. Simon, and M. Gu, "Spectral relaxation for k-means clustering," in *Proc. Int. Conf. Neural Inf. Process. Syst., Natural Synth.*, 2001, pp. 1057–1064.
[22] S. Wang and W. Guo, "Robust co-clustering via dual local learning and high-order matrix factorization," *Knowl.-Based Syst.*, vol. 138, pp. 176–187, Dec. 2017.
[23] E. E. Papalexakis, N. D. Sidiropoulos, and R. Bro, "From K-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 493–506, Jan. 2013.
[24] M. Rege, M. Dong, and F. Fotouhi, "Co-clustering documents and words using bipartite isoperimetric graph partitioning," in *Proc. Int. Conf. Data Mining*, 2006, pp. 532–541.
[25] Y. Chen, M. Dong, and W. Wan, "Image co-clustering with multi-modality features and user feedbacks," in *Proc. Int. Conf. Multimedia*, 2009, pp. 689–692.
[26] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
[27] J. Luo, B. Liu, B. Cao, and S. Wang, "Identifying miRNA-mRNA regulatory modules based on overlapping neighborhood expansion from multiple types of genomic data," in *Proc. Int. Conf. Intell. Comput.*, 2016, pp. 234–246.
[28] G. Pio, M. Ceci, C. Loglisci, D. D'Elia, and D. Malerba, "Hierarchical and overlapping co-clustering of mRNA: miRNA interactions," in *Proc. Eur. Conf. Artif. Intell.*, 2012, pp. 654–659.
[29] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.
[30] D. D. Lee and H. S. Seung, "Unsupervised learning by convex and conic coding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9, 1997, pp. 515–521.
[31] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.
[32] J. Kim and H. Park, "Sparse nonnegative matrix factorization for clustering," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep., 2008.
[33] P. Paatero, "Least squares formulation of robust non-negative factor analysis," *Chemometrics Intell. Lab. Syst.*, vol. 37, no. 1, pp. 23–35, May 1997.
[34] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. Knowl. Discovery Data Mining*, 2006, pp. 126–135.
[35] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 269–274.
[36] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *Proc. Knowl. Discovery Data Mining*, 2003, pp. 89–98.
[37] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 153–160.
[38] R. K. C. Fan, *Spectral Graph Theory*. Providence, RI, USA: AMS, 1997.
[39] F. Shang, L. C. Jiao, and F. Wang, "Graph dual regularization non-negative matrix factorization for co-clustering," *Pattern Recognit.*, vol. 45, no. 6, pp. 2237–2250, 2012.
[40] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.
[41] S. Wang and A. Huang, "Penalized nonnegative matrix tri-factorization for co-clustering," in *Expert Syst. Appl.*, vol. 78, pp. 64–73, Jul. 2017.
[42] F. Pompili, N. Gillis, P.-A. Absil, and F. Glineur, "Two algorithms for orthogonal nonnegative matrix factorization with application to clustering," *Neurocomputing*, vol. 141, pp. 15–25, Oct. 2014.
[43] C. Ding, X. He, H. D. Simon, and R. Jin, "On the equivalence of nonnegative matrix factorization and k-means- spectral clustering," in *Factorization*, Tech. Rep. GT-CSE-08-01, 2005.
[44] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4129–4138.
[45] B. Mohar, Y. Alavi, G. Chartrand, and O. R. Oellermann, "The laplacian spectrum of graphs," in *Graph Theory, Combinatorics, and Applications*. New York, NY, USA: Wiley, 1991, pp. 871–898.
[46] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 977–986.
[47] K. Fan, "On a theorem of Weyl concerning eigenvalues of linear transformations I," in *Proc. Nat. Acad. Sci. USA*, vol. 35, no. 11, pp. 652–655, 1949.
[48] H. Wang, F. Nie, H. Huang, and F. Makedon, "Fast nonnegative matrix tri-factorization for large-scale data co-clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1553–1558.
[49] J. Han, K. Song, F. Nie, and X. Li, "Bilateral k-means algorithm for fast co-clustering," in *Proc. Nat. Conf. Artif. Intell.*, 2017, pp. 1969–1975.
[50] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 313–319.
[51] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *Proc. Int. Conf. Data Mining*, 2008, pp. 63–72.
[52] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2003, pp. 267–273.

[53] L. Lovász and M. D. Plummer, *Matching Theory*. Providence, RI, USA: AMS, 2009, vol. 367.
[54] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statist. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.
[55] G. W. Milligan and M. C. Cooper, "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivariate Behav. Res.*, vol. 21, no. 4, pp. 441–458, 1986.
[56] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Mach. Learn.*, vol. 55, no. 3, pp. 311–331, 2004.



RENJIE LIN received the master's degree from the Department of Computer Science and Informatics, University of Leicester, U.K., in 2017. He is currently pursuing the Ph.D. degree with the College of Mathematics and Computer Science, Fuzhou University. His research interests include machine learning, computer vision, and natural language processing.

SHIPING WANG received the Ph.D. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2014. From 2015 to 2016, he was a Research Fellow with Nanyang Technological University, Singapore. He is currently a Qishan Scholar with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His research interests include machine learning, computer vision, and granular computing.

WENZHONG GUO received the Ph.D. degree from the Department of Physics and Information Engineering, Fuzhou University, Fuzhou, China, in 2010, where he is currently a Professor and the Director of the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing. He was a Postdoctoral Research Scholar with the Department of Computer Science, National University of Defense and Technology, Changsha, China, from 2011 to 2014. He was a Visiting Professor with the Faculty of Engineering, Information, and System, University of Tsukuba, Japan, in 2013, and with the Department of Computer Science and Engineering, State University of New York at Buffalo, USA, in 2016. His research interests include the fields of data mining, machine learning, and artificial intelligence.