Knowledge-Based Systems 201–202 (2020) 105987

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Sparse neighbor constrained co-clustering via category consistency learning

Zhoumin Lu [a,b], Genggeng Liu [a,b], Shiping Wang [a,b,*]

[a] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China
[b] Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou 350116, China

ARTICLE INFO

Article history:
Received 17 January 2020
Received in revised form 28 April 2020
Accepted 29 April 2020
Available online 12 May 2020

Keywords:
Machine learning
Co-clustering
Nonnegative matrix factorization
Category consistency
Neighbor constraint
Dual regularization

ABSTRACT

Clustering has long been an enduring and promising task in machine learning. However, developed one-side clustering is still insufficient to explore the context of data, such as texts and genes. Hence, developing two-way clustering has drawn more attention in recent years, which tends to cluster samples and features simultaneously. This paper proposes a sparse neighbor constrained co-clustering via category consistency learning, for alleviating the misclassification of close points. Following an additional observation, samples often fall into the same category as their neighbors, as do features. Accordingly, the co-clustering problem is formulated as nonnegative matrix tri-factorization appended dual regularizers, considering coherence between data affinity and label assignment. Then, a multiplicative alternating scheme is raised for objective optimization, whose convergence and correctness are theoretically guaranteed. Furthermore, the proposed approach is validated on six datasets using three evaluation metrics, whose parameter sensitivity is analyzed as well. Finally, comprehensive experiments show that our algorithm is competitive against existing ones.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Clustering, as a fundamental and important task in machine learning, has been greatly developed. Its applications include data mining, genetic engineering and computer vision. Concretely, clustering groups data without knowing their labels, and can be roughly divided into three categories: graph-based, density-based and distance-based clustering. Among the graph-based methods, ratio cut [1], min–max cut [2] and normalized cut [3] are well known. Density-based approaches include density-based spatial clustering of applications with noise (DBSCAN) [4], density peak clustering (DPC) [5] and maximin separation probability clustering (MSPC) [6]. For distance-based algorithms, besides the widely used Euclidean distance, Minkowski distance [7] and Mahalanobis distance [8] are also adopted. As a classic distance-based clustering, K-means [9] has abundant variants such as K-means++ [10], kernel K-means [11,12] and fuzzy K-means [13,14]. In addition, K-means and spectral clustering are closely related to nonnegative matrix factorization (NMF) [15,16].

However, recent studies have indicated that two-way clustering (co-clustering) is superior to the aforementioned one-side clustering. These co-clustering algorithms not only group samples but also synchronously partition features. Because of its duality between samples and features, co-clustering is naturally suited to structured data, resulting in many applications. For instance, it can cluster not only documents and words in text mining [17,18], but also users and movies in recommendation systems [19] concurrently. Moreover, co-clustering has been applied extensively in data mining [20–22], genetic engineering [23] and computer vision [24,25]. Currently, although a lot of algorithms [26–28] have appeared, there are still many problems to be settled. For example, similar samples are clustered into different groups.

In this paper, an effective co-clustering algorithm is proposed. It considers consistency between data affinity and label assignment, based on nonnegative matrix factorization appended two regularizers. For objective optimization, some iterative updates are employed, whose convergence is proved in theory. Furthermore, experiments demonstrate that our algorithm achieves superior performance. In short, the main contributions are summarized as follows.

- Transform co-clustering into nonnegative matrix tri-factorization with dual regularizers for learning category consistency.
- Develop and prove a multiplicative iterative updating algorithm for solving the problem of objective optimization.
- Confirm the validity of method by theories and experiments, achieving superior performance over compared ones.

The remainder is organized as below. Section 2 describes several related works briefly. Section 3 introduces the proposed

[*] Corresponding author.
E-mail address: shipingwangphd@163.com (S. Wang).

https://doi.org/10.1016/j.knosys.2020.105987
0950-7051/© 2020 Elsevier B.V. All rights reserved.

co-clustering algorithm in detail. In Section 4, comprehensive experiments are conducted for algorithm validation. Finally, some conclusions and future works are drawn in Section 5.

2. Related works

Since NMF [29,30] was put forward, it has captured the attention of developers. NMF decomposes the original matrix into the product of two nonnegative matrices for seeking a good low-rank approximation. This has a wide range of real-world applications, such as spectral data analysis, document clustering, face representation learning and gene expression. Additionally, its variants keep cropping up to accommodate varying problems and fundamental form is

$$\begin{aligned} J_{NMF} &= \left\| X - SG^T \right\|_F^2 \\ \text{s.t.} \quad & S \geq 0, G \geq 0 \end{aligned} \tag{1}$$

As one variant, semi-nonnegative matrix factorization (SNMF) [31] relaxes the nonnegative constraint of NMF and is more appropriate for general data. It makes an attempt to minimize the following.

$$\begin{aligned} J_{SNMF} &= \left\| X - SG^T \right\|_F^2 \\ \text{s.t.} \quad & G \geq 0 \end{aligned} \tag{2}$$

As another extension, graph regularized nonnegative matrix factorization (GNMF) [32] introduces a graph regularizer for NMF. GNMF attempts to exploit the underlying representation while also taking into account data manifold. It can be expressed as below.

$$\begin{aligned} J_{GNMF} &= \left\| X - SG^T \right\|_F^2 + \lambda \operatorname{Tr}(G^T L_G G) \\ \text{s.t.} \quad & S \geq 0, G \geq 0 \end{aligned} \tag{3}$$

Generally, the co-clustering problem is converted into a nonnegative matrix tri-factorization (NMTF) on account of inherent comparability. Unlike the NMF learning a hidden representation, NMTF obtains the clustering results directly. It factorizes the original matrix into three nonnegative matrices for simultaneous grouping by row and column.

Dual graph regularized nonnegative matrix tri-factorization (DNMTF) [33], as a classic co-clustering algorithm, deems that features are also subject to a manifold like data. Therefore, the following objective is formulated.

$$\begin{aligned} J_{DNMTF} &= \left\| X - FSG^T \right\|_F^2 + \lambda \operatorname{Tr}(F^T L_F F) + \mu \operatorname{Tr}(G^T L_G G) \\ \text{s.t.} \quad & F \geq 0, S \geq 0, G \geq 0 \end{aligned} \tag{4}$$

Akin to orthogonal nonnegative matrix tri-factorization (ONMTF) [34], dual local learning based co-clustering (DLLC) [35] adds the orthogonal constraint but also learns a local structure. It is formed as follows.

$$\begin{aligned} J_{DLLC} &= \left\| X - FSG^T \right\|_F^2 + \lambda \operatorname{Tr}(F^T L_F F) + \mu \operatorname{Tr}(G^T L_G G) \\ \text{s.t.} \quad & F \geq 0, S \geq 0, G \geq 0, F^T F = I, G^T G = I \end{aligned} \tag{5}$$

However, penalized nonnegative matrix factorization (PNMF) [36] replaces orthogonal constraints with two penalty terms. In addition, the third penalty term is introduced for inter-cluster discriminability. It is shown as below.

$$\begin{aligned} J_{PNMF} &= \left\| X - FSG^T \right\|_F^2 + \alpha \operatorname{Tr}\left(F \Phi F^T\right) + \beta \operatorname{Tr}\left(G \Psi G^T\right) + \gamma \operatorname{Tr}\left(S^T S\right) \\ \text{s.t.} \quad & F \geq 0, S \geq 0, G \geq 0 \end{aligned} \tag{6}$$

3. Neighbor constrained co-clustering

In this section, the neighbor constrained co-clustering problem is presented and then addressed by an iterative algorithm, whose convergence is proved.

3.1. Problem formulation

Given a data matrix $X \in \mathbb{R}^{d \times n}$, in which d represents the number of features and n corresponds to the number of samples, denote the feature space and sample space as $\mathcal{F} = \{f_1, \ldots, f_d\}$ and $\mathcal{X} = \{x_1, \ldots, x_n\}$. It is observed that $X = (x_1, \ldots, x_n) = (f_1, \ldots, f_d)^T$.

Traditional clustering tends to group samples into c clusters. If we relax the discrete indicator matrix into a continuous nonnegative one, this problem can be approximately transformed as the following optimization objective,

$$\begin{aligned} \min_{S,G} & \frac{1}{2} \left\| X - SG^T \right\|_F^2 \\ \text{s.t.} \quad & S \geq 0, G \geq 0 \end{aligned} \tag{7}$$

where coefficient matrix $S \in \mathbb{R}^{d \times c}$ and indicator matrix $G \in \mathbb{R}^{n \times c}$.

Analogously, the co-clustering attempts to partition the features into c_1 clusters as well as samples into c_2 clusters. If we alleviate two discrete indicator matrices into continuous nonnegative ones, this problem can be approximately transformed as the following optimization objective,

$$\begin{aligned} \min_{F,S,G} & \frac{1}{2} \left\| X - FSG^T \right\|_F^2 \\ \text{s.t.} \quad & F \geq 0, S \geq 0, G \geq 0 \end{aligned} \tag{8}$$

where $S \in \mathbb{R}^{c_1 \times c_2}$ is a coefficient matrix shared by clustering indicator matrices $F \in \mathbb{R}^{d \times c_1}$ and $G \in \mathbb{R}^{n \times c_2}$.

Naturally, the more similar the samples, the more consistent their category labels. Dually, the features are also in the same fashion. Therefore, we append two regularizers to hold coherence between data affinity and label assignment as follows,

$$\begin{aligned} \min_{F,S,G,Z_1,Z_2} & \frac{1}{2} \left\| X - FSG^T \right\|_F^2 + \frac{\alpha}{2} \left\| W_1 - FZ_1^T \right\|_F^2 + \frac{\beta}{2} \left\| W_2 - GZ_2^T \right\|_F^2 \\ \text{s.t.} \quad & F \geq 0, S \geq 0, G \geq 0 \end{aligned} \tag{9}$$

where W_1 and W_2 represent the similarities of features and samples, respectively, and Z_1, Z_2 are the coefficient matrices.

3.2. Optimization algorithm

As can be seen, we are unable to obtain a closed-form solution from Eq. (9) on account of its non-convexity. Fortunately, the objective function is convex in a single variable while fixing others. Hence, we propose a multiplicative iterative scheme for optimization, as follows.

3.2.1. Constructing $\mathcal{L}$

First, the original problem minimizes the following objective,

$$\begin{aligned} J &= \frac{1}{2} \left\| X - FSG^T \right\|_F^2 + \frac{\alpha}{2} \left\| W_1 - FZ_1^T \right\|_F^2 + \frac{\beta}{2} \left\| W_2 - GZ_2^T \right\|_F^2 \\ \text{s.t.} \quad & F \geq 0, S \geq 0, G \geq 0 \end{aligned} \tag{10}$$

where $\alpha, \beta \geq 0$ are employed to balance the reconstruction error. If $\alpha = \beta = 0$, then this objective will degenerate into the traditional one. Furthermore, the second term, with respect to features, strives for consistency between data affinity and label assignment. Then, the third one tends to keep accordant between data affinity and label assignment, regarding samples.

For solving constraints, the Lagrange multipliers Φ, Ψ and Ω are introduced, and then the Lagrange function is constructed as

$$\begin{aligned}\mathcal{L} =& \frac{1}{2}\left\|X - FSG^T\right\|_F^2 + \frac{\alpha}{2}\left\|W_1 - FZ_1^T\right\|_F^2 + \frac{\beta}{2}\left\|W_2 - GZ_2^T\right\|_F^2 \\ & - \operatorname{Tr}(\Phi F^T) - \operatorname{Tr}(\Psi S^T) - \operatorname{Tr}(\Omega G^T)\end{aligned} \tag{11}$$

It is observed that

$$\begin{aligned}\left\|X - FSG^T\right\|_F^2 =& \operatorname{Tr}((X - FSG^T)^T(X - FSG^T)) \\ =& \operatorname{Tr}(X^TX) - 2\operatorname{Tr}(X^TFSG^T) + \operatorname{Tr}(GS^TF^TFSG^T)\end{aligned} \tag{12}$$

$$\begin{aligned}\left\|W_1 - FZ_1^T\right\|_F^2 =& \operatorname{Tr}((W_1 - FZ_1^T)^T(W_1 - FZ_1^T)) \\ =& \operatorname{Tr}(W_1^TW_1) - 2\operatorname{Tr}(W_1^TFZ_1^T) + \operatorname{Tr}(Z_1F^TFZ_1^T)\end{aligned} \tag{13}$$

$$\begin{aligned}\left\|W_2 - GZ_2^T\right\|_F^2 =& \operatorname{Tr}((W_2 - GZ_2^T)^T(W_2 - GZ_2^T)) \\ =& \operatorname{Tr}(W_2^TW_2) - 2\operatorname{Tr}(W_2^TGZ_2^T) + \operatorname{Tr}(Z_2G^TGZ_2^T)\end{aligned} \tag{14}$$

Accordingly, the Lagrange function can be rewritten as

$$\begin{aligned}\mathcal{L} =& \frac{1}{2}\operatorname{Tr}(X^TX) - \operatorname{Tr}(X^TFSG^T) + \frac{1}{2}\operatorname{Tr}(GS^TF^TFSG^T) \\ & + \frac{\alpha}{2}\operatorname{Tr}(W_1^TW_1) - \alpha\operatorname{Tr}(W_1^TFZ_1^T) + \frac{\alpha}{2}\operatorname{Tr}(Z_1F^TFZ_1^T) \\ & + \frac{\beta}{2}\operatorname{Tr}(W_2^TW_2) - \beta\operatorname{Tr}(W_2^TGZ_2^T) + \frac{\beta}{2}\operatorname{Tr}(Z_2G^TGZ_2^T) \\ & - \operatorname{Tr}(\Phi F^T) - \operatorname{Tr}(\Psi S^T) - \operatorname{Tr}(\Omega G^T)\end{aligned} \tag{15}$$

3.2.2. Obtaining W_1 and W_2

For sparsity, it makes more sense to consider k-nearest neighbors [37] instead of all nodes. Hence, the feature affinity matrix W_1 is constructed as follows.

$$W_{ij}^1 = \begin{cases} 1, & \text{if } f_j \in \mathcal{N}(f_i) \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where W_{ij}^1 measures how close f_j is to f_i and $\mathcal{N}(f_i)$ denotes k_1 nearest neighbors of f_i. Similarly, the sample affinity matrix W_2 is structured as follows.

$$W_{ij}^2 = \begin{cases} 1, & \text{if } x_j \in \mathcal{N}(x_i) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where W_{ij}^2 measures how close x_j is to x_i and $\mathcal{N}(x_i)$ denotes k_2 nearest neighbors of x_i. In addition, some kernel functions [38] can be adopted to distinguish the differences between different neighbors, yet will introduce extra parameters.

3.2.3. Computing Z_1 and Z_2

By taking derivatives of $\mathcal{L}$ on Z_1 and Z_2, we have

$$\frac{\partial\mathcal{L}}{\partial Z_1} = -\alpha W_1^TF + \alpha Z_1F^TF \tag{18}$$

$$\frac{\partial\mathcal{L}}{\partial Z_2} = -\beta W_2^TG + \beta Z_2G^TG \tag{19}$$

Ulteriorly, setting $\frac{\partial\mathcal{L}}{\partial Z_1} = 0$ and $\frac{\partial\mathcal{L}}{\partial Z_2} = 0$ result in

$$Z_1 = W_1^TF(F^TF)^{-1} \tag{20}$$

$$Z_2 = W_2^TG(G^TG)^{-1} \tag{21}$$

3.2.4. Updating F, S and G

By taking derivatives of $\mathcal{L}$ on F, S and G, we have

$$\frac{\partial\mathcal{L}}{\partial F} = -XGS^T + FSG^TGS^T - \alpha W_1Z_1 + \alpha FZ_1^TZ_1 - \Phi \tag{22}$$

$$\frac{\partial\mathcal{L}}{\partial S} = -F^TXG + F^TFSG^TG - \Psi \tag{23}$$

$$\frac{\partial\mathcal{L}}{\partial G} = -X^TFS + GS^TF^TFS - \beta W_2Z_2 + \beta GZ_2^TZ_2 - \Omega \tag{24}$$

Combined with the Karush–Kuhn–Tucker conditions $\Phi_{ij}F_{ij} = 0$, $\Psi_{ij}S_{ij} = 0$ and $\Omega_{ij}G_{ij} = 0$, setting $\frac{\partial\mathcal{L}}{\partial F} = 0$, $\frac{\partial\mathcal{L}}{\partial S} = 0$ and $\frac{\partial\mathcal{L}}{\partial G} = 0$ lead to

$$(-XGS^T + FSG^TGS^T - \alpha W_1Z_1 + \alpha FZ_1^TZ_1)_{ij}F_{ij} = 0 \tag{25}$$

$$(-F^TXG + F^TFSG^TG)_{ij}S_{ij} = 0 \tag{26}$$

$$(-X^TFS + GS^TF^TFS - \beta W_2Z_2 + \beta GZ_2^TZ_2)_{ij}G_{ij} = 0 \tag{27}$$

Introducing $M = W_1Z_1 = M^+ - M^-$, $N = Z_1^TZ_1 = N^+ - N^-$, $P = W_2Z_2 = P^+ - P^-$ and $Q = Z_2^TZ_2 = Q^+ - Q^-$, Eqs. (25) and (27) can be rewritten as

$$(-XGS^T + FSG^TGS^T - \alpha M^+ + \alpha M^- + \alpha FN^+ - \alpha FN^-)_{ij}F_{ij} = 0 \tag{28}$$

$$(-X^TFS + GS^TF^TFS - \beta P^+ + \beta P^- + \beta GQ^+ - \beta GQ^-)_{ij}G_{ij} = 0 \tag{29}$$

where $A^+ = \frac{|A|+A}{2}$ and $A^- = \frac{|A|-A}{2}$ for any matrix A.

In line with an optimization framework for nonnegative quadratic problems, Eqs. (28), (26) and (29) bring about the following updating rules.

$$F_{ij} \leftarrow F_{ij}\left[\frac{(XGS^T + \alpha M^+ + \alpha FN^-)_{ij}}{(FSG^TGS^T + \alpha M^- + \alpha FN^+)_{ij}}\right]^{\frac{1}{2}} \tag{30}$$

$$S_{ij} \leftarrow S_{ij}\left[\frac{(F^TXG)_{ij}}{(F^TFSG^TG)_{ij}}\right]^{\frac{1}{2}} \tag{31}$$

$$G_{ij} \leftarrow G_{ij}\left[\frac{(X^TFS + \beta P^+ + \beta GQ^-)_{ij}}{(GS^TF^TFS + \beta P^- + \beta GQ^+)_{ij}}\right]^{\frac{1}{2}} \tag{32}$$

3.2.5. Label assignment

After completing the update, the final cluster indicator matrices $\widetilde{F}$ and $\widetilde{G}$ become available. Moreover, the label of the ith feature is assigned by

$$l(f_i) = \arg\max_j \widetilde{F}_{ij} \tag{33}$$

and the label of the ith sample is assigned by

$$l(x_i) = \arg\max_j \widetilde{G}_{ij} \tag{34}$$

Based on the above analyses, the entire optimization process can be summarized as Algorithm 1, named SNCC for short.

Algorithm 1 Sparse neighbor constrained co-clustering

```
Input: Data matrix $X \in \mathbb{R}^{d\times n}$, feature cluster numbers $c_1$, sample
    cluster numbers $c_2$, and parameters $\alpha$ and $\beta$.
Output: Feature labels $\{l(f_i)\}_{i=1}^{c_1}$ and sample labels $\{l(x_i)\}_{i=1}^{c_2}$.
Initialize $F$, $S$ and $G$;
Obtain $W_1$ and $W_2$;
while non-convergence do
    Compute $Z_1 = W_1^TF(F^TF)^{-1}$;
    Compute $Z_2 = W_2^TG(G^TG)^{-1}$;
    Update $F_{ij} \leftarrow F_{ij}\left[\frac{(XGS^T+\alpha M^++\alpha FN^-)_{ij}}{(FSG^TGS^T+\alpha M^-+\alpha FN^+)_{ij}}\right]^{\frac{1}{2}}$;
    Update $S_{ij} \leftarrow S_{ij}\left[\frac{(F^TXG)_{ij}}{(F^TFSG^TG)_{ij}}\right]^{\frac{1}{2}}$;
    Update $G_{ij} \leftarrow G_{ij}\left[\frac{(X^TFS+\beta P^++\beta GQ^-)_{ij}}{(GS^TF^TFS+\beta P^-+\beta GQ^+)_{ij}}\right]^{\frac{1}{2}}$;
end while
Assign feature labels $l(f_i)$ and sample labels $l(x_i)$.
```

3.3. Convergence analysis

Definition 1. If the following conditions are satisfied, $Z(h, h')$ is an auxiliary function of $F(h)$.

$$Z(h, h') \geq F(h), Z(h, h) = F(h) \tag{35}$$

Lemma 1. *If $Z(h, h')$ is an auxiliary function of $F(h)$, $F(h)$ is non-increasing under the updating scheme*

$$h^{(t+1)} = \arg\min_{h} Z(h, h^{(t)}) \tag{36}$$

Proof. $F(h^{(t+1)}) \leq Z(h^{(t+1)}, h^{(t)}) \leq Z(h^{(t)}, h^{(t)}) = F(h^{(t)})$ □

Lemma 2. *$\forall A \in \mathbb{R}_{+}^{n\times n}, B \in \mathbb{R}_{+}^{k\times k}, S \in \mathbb{R}_{+}^{n\times k}, S' \in \mathbb{R}_{+}^{n\times k}$, and A, B are symmetric, the following inequality holds*

$$\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{(AS'B)_{ij}S_{ij}^{2}}{S'_{ij}} \geq \mathrm{Tr}(S^{T}ASB) \tag{37}$$

Theorem 1. *Denote*

$$\begin{aligned} J(F) = &- \mathrm{Tr}(X^{T}FSG^{T}) + \frac{1}{2}\mathrm{Tr}(GS^{T}F^{T}FSG^{T}) \\ &- \alpha\,\mathrm{Tr}(MF^{T}) + \frac{\alpha}{2}\mathrm{Tr}(FNF^{T}) \end{aligned} \tag{38}$$

then its auxiliary function shows that

$$\begin{aligned} Z(F, F') = &- \sum_{i=1}^{d}\sum_{j=1}^{c_1}(XGS^{T})_{ij}F'_{ij}(1 + \log\frac{F_{ij}}{F'_{ij}}) \\ &+ \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{c_1}\frac{(F'SG^{T}GS^{T})_{ij}F_{ij}^{2}}{F'_{ij}} + \alpha\sum_{i=1}^{d}\sum_{j=1}^{c_1}M_{ij}^{-}\frac{F_{ij}^{2} + F_{ij}'^{2}}{2F'_{ij}} \\ &- \alpha\sum_{i=1}^{d}\sum_{j=1}^{c_1}M_{ij}^{+}F'_{ij}(1 + \log\frac{F_{ij}}{F'_{ij}}) + \frac{\alpha}{2}\sum_{i=1}^{d}\sum_{j=1}^{c_1}\frac{(F'N^{+})_{ij}F_{ij}^{2}}{F'_{ij}} \\ &- \frac{\alpha}{2}\sum_{i=1}^{d}\sum_{j=1}^{c_1}\sum_{k=1}^{c_1}N_{jk}^{-}F'_{ij}F'_{ik}(1 + \log\frac{F_{ij}F_{ik}}{F'_{ij}F'_{ik}}) \end{aligned} \tag{39}$$

which is convex in F. Moreover, its global minima is

$$F_{ij} = \arg\min_{F_{ij}} Z(F, F') = F'_{ij}\left[\frac{(XGS^{T} + \alpha M^{+} + \alpha FN^{-})_{ij}}{(FSG^{T}GS^{T} + \alpha M^{-} + \alpha FN^{+})_{ij}}\right]^{\frac{1}{2}} \tag{40}$$

Proof. See Appendix A. □

Theorem 2. *Denote*

$$J(S) = - \mathrm{Tr}(X^{T}FSG^{T}) + \frac{1}{2}\mathrm{Tr}(GS^{T}F^{T}FSG^{T}) \tag{41}$$

then its auxiliary function shows that

$$\begin{aligned} Z(S, S') = &- \sum_{i=1}^{c_1}\sum_{j=1}^{c_2}(F^{T}XG)_{ij}S'_{ij}(1 + \log\frac{S_{ij}}{S'_{ij}}) \\ &+ \frac{1}{2}\sum_{i=1}^{c_1}\sum_{j=1}^{c_2}\frac{(F^{T}FS'G^{T}G)_{ij}S_{ij}^{2}}{S'_{ij}} \end{aligned} \tag{42}$$

which is convex in S. Moreover, its global minima is

$$S_{ij} = \arg\min_{S_{ij}} Z(S, S') = S'_{ij}\left[\frac{(F^{T}XG)_{ij}}{(F^{T}FSG^{T}G)_{ij}}\right]^{\frac{1}{2}} \tag{43}$$

Proof. See Appendix B. □

Theorem 3. *Denote*

$$\begin{aligned} J(G) = &- \mathrm{Tr}(X^{T}FSG^{T}) + \frac{1}{2}\mathrm{Tr}(GS^{T}F^{T}FSG^{T}) \\ &- \beta\,\mathrm{Tr}(PG^{T}) + \frac{\beta}{2}\mathrm{Tr}(GQG^{T}) \end{aligned} \tag{44}$$

then its auxiliary function shows that

$$\begin{aligned} Z(G, G') = &- \sum_{i=1}^{n}\sum_{j=1}^{c_2}(X^{T}FS)_{ij}G'_{ij}(1 + \log\frac{G_{ij}}{G'_{ij}}) \\ &+ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{c_2}\frac{(G'S^{T}F^{T}FS)_{ij}G_{ij}^{2}}{G'_{ij}} + \beta\sum_{i=1}^{n}\sum_{j=1}^{c_2}P_{ij}^{-}\frac{G_{ij}^{2} + G_{ij}'^{2}}{2G'_{ij}} \\ &- \beta\sum_{i=1}^{n}\sum_{j=1}^{c_2}P_{ij}^{+}G'_{ij}(1 + \log\frac{G_{ij}}{G'_{ij}}) + \frac{\beta}{2}\sum_{i=1}^{n}\sum_{j=1}^{c_2}\frac{(G'Q^{+})_{ij}G_{ij}^{2}}{G'_{ij}} \\ &- \frac{\beta}{2}\sum_{i=1}^{n}\sum_{j=1}^{c_2}\sum_{k=1}^{c_2}Q_{jk}^{-}G'_{ij}G'_{ik}(1 + \log\frac{G_{ij}G_{ik}}{G'_{ij}G'_{ik}}) \end{aligned} \tag{45}$$

which is convex in G. Moreover, its global minima is

$$G_{ij} = \arg\min_{G_{ij}} Z(G, G') = G'_{ij}\left[\frac{(X^{T}FS + \beta P^{+} + \beta GQ^{-})_{ij}}{(GS^{T}F^{T}FS + \beta P^{-} + \beta GQ^{+})_{ij}}\right]^{\frac{1}{2}} \tag{46}$$

Proof. See Appendix C. □

3.4. Computational cost

For a dataset, its feature affinity and sample affinity matrices need $O(d^2n)$ and $O(n^2d)$ to construct, where d and n are the numbers of features and samples, respectively. Furthermore, the multiplicative altering process needs $O(dnct)$ to run, where c and t are the numbers of clusters and iterations, respectively. Overall, the computational cost of the algorithm is $O(d^2n + n^2d + dnct)$.

4. Experiments

In this section, the proposed approach is validated on six datasets measured by three evaluation metrics, whose parameter sensitivity is analyzed as well, showing its efficiency and effectiveness.

4.1. Evaluation metrics

The adopted evaluation metrics consist of clustering accuracy, normalized mutual information and adjusted Rand index. Clustering accuracy directly reflects the misclassification of data points, whereas normalized mutual information and adjusted Rand index judge the correctness of data pairs belonging to same category respectively from the perspectives of combinatorial mathematics and probability theory. In essence, they measure the difference between real distribution and predicted distribution. It is worth noting that the higher these metrics are, the better the performance is.

Given a dataset $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, let $S = \left\{\left(\mathbf{x}_i, \mathbf{x}_j\right) | c_i = c_j, \widetilde{c}_i = \widetilde{c}_j, i < j\right\}$ and $D = \left\{\left(\mathbf{x}_i, \mathbf{x}_j\right) | c_i \neq c_j, \widetilde{c}_i \neq \widetilde{c}_j, i < j\right\}$, where c_i and $\widetilde{c}_i$ represent label and prediction of the ith sample. Besides, denote $C = \{C_i\}_{i=1}^{k}$ and $\widetilde{C} = \{\widetilde{C}_j\}_{j=1}^{m}$ as the real and predicted clusters,

Table 1
Description for datasets.

Data set	Data type	#instances	#features	#classes
LUNG	Gene expression	203	3312	5
CLL_SUB_111	Gene expression	111	11340	3
BASEHOCK	Text document	1993	4862	2
RELATHE	Text document	1427	4322	2
MNIST_SUB	Digit image	6996	784	10
MSRA25	Face image	1799	256	12

respectively. Then, the mutual information, information entropy and Rand index are defined as

$$I(C,\widetilde{C}) = \sum_{i=1}^{k}\sum_{j=1}^{m} |C_i \cap \widetilde{C}_j| \log \frac{n|C_i \cap \widetilde{C}_j|}{|C_i| \cdot |\widetilde{C}_j|} \tag{47}$$

$$H(C) = \sum_{i=1}^{k} |C_i| \log \frac{|C_i|}{n} \tag{48}$$

$$\mathrm{RI} = \frac{2(|S| + |D|)}{n(n-1)} \tag{49}$$

Furthermore, the clustering accuracy, normalized mutual information and adjusted Rand index can be calculated by

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} \delta\left(c_i, \mathrm{map}\left(\widetilde{c}_i\right)\right)}{n} \tag{50}$$

$$\mathrm{NMI} = \frac{I(C,\widetilde{C})}{\sqrt{H(C)H(\widetilde{C})}} \tag{51}$$

$$\mathrm{ARI} = \frac{\mathrm{RI} - E(\mathrm{RI})}{\max(\mathrm{RI}) - E(\mathrm{RI})} \tag{52}$$

where $\mathrm{map}(\cdot)$ and $\delta(\cdot)$ represent the permutation mapping function and Kronecker delta function, respectively. In addition, the best match can be generated by Kuhn–Munkres algorithm.

4.2. Data sets

To verify the algorithm roundly, six publicly available datasets are selected from various data types, including gene expression, text documents, digit images and face images. For instance, the MSRA25 dataset contains 1799 samples with 256 features, collected from 12 classes. The details of adopted datasets are summarized in Table 1.

4.3. Experimental setting

From a comprehensive viewpoint, some related clustering and co-clustering methods are both compared, which stay either classic or state-of-the-art, such as K-means [9], NMF [30], SNMF [31], GNMF [32], DNMTF [33], DLLC [35], PNMF [36], FNMTF [39], BKM [40] and SOBG [41]. The details are as follows.

K-means is widely used as a baseline clustering algorithm. FNMTF is a fast nonnegative matrix tri-factorization algorithm for co-clustering. BKM utilizes a bilateral K-means algorithm for fast co-clustering. SOBG learns a structured optimal bipartite graph for co-clustering. Other compared algorithms have been introduced in related works.

For K-means, its settings are default. For GNMF, its regularization parameter α is set to 100. For DNMTF, its regularization parameters λ and μ are both set to 200. For SOBG, its parameter λ is set to 10. For DLLC, its regularization parameters α and β are both set to 1. For PNMF, its parameters α, β and γ are set to 1. For SNCC, its parameters α and β are set to 0.1. For NMF-based methods, they are all initialized by K-means and iterate up to 20 times.

For all algorithms, their sample cluster numbers are consistent with the number of real categories. For co-clustering, the number of feature clusters is set as same as sample clusters on account of unknown feature cluster numbers. For graph construction, k-nearest neighbor algorithm is adopted and the weights are binary. In addition, the nearest neighbor numbers of features and samples are set to 10. Considering the initialization sensitivity of some methods, all experiments are repeated 30 times. Meanwhile, their mean and standard deviation are recorded for comparison.

4.4. Result analysis

For better understanding, the clustering results of LUNG dataset are visualized in Fig. 1, depending on t-SNE [42]. As can be seen, the ground truth (GT) exhibits unbalanced samples and hard-to-separate clusters. For this dataset, even if some methods only group all samples into one category, they still have good performance in a little short of others, such as FNMTF and BKM. However, this case is not expected. Obviously, it is also non-expected to partition all samples into two or three classes, such as GNMF and DNMTF. Other approaches are better in the aforementioned aspect, yet still have more or less problems. For example, DLLC categorizes samples into one cluster overmuch, even though other categories exist. As a whole, SNCC does better because of its result closer to the real distribution.

As shown in Table 2, SNCC obtains higher clustering accuracy than other algorithms. By contrast, some methods are not good at MNIST and MSRA25 such as BKM and SOBG. Table 3 shows that, SNCC outperforms others on tested datasets except RELATHE, measured by normalized mutual information, but is second only to DNMTF. Seen from Table 4, SNCC still achieves acceptable performance in adjusted Rand index. As for runtime, it is reported in Fig. 2. Clearly, SOBG is time-consuming while the computational cost of SNCC is not much different from K-means. Confronted with image datasets, the time consumption of most methods fluctuate evidently, while SNCC is relatively stable. To all appearances, SNCC holds competitive against others.

In general, two-way clustering works better than one-side clustering but not always. On the one hand, sample clustering does benefit from feature grouping. On the other hand, feature grouping may cause the following problems.

- The number of feature clusters is unknown.
- The structure of features is difficult to explore.
- Most methods are sensitive to initialization.

For the third problem, one possible reason is that matrix factorization based algorithms can converge only to the local optimum in each iteration. In the future, how to solve these problems will become the focus.

4.5. Parameter sensitivity

In this subsection, the performance of our algorithm is analyzed with the change of parameters. To be specific, the number of nearest neighbors and the weight of regularizers are illustrated in Figs. 3 and 4, measured by clustering accuracy. For all parameters, they should not be large. Besides, if k_1 is slightly lower than k_2, the performance will be better. So do weights α and β. Generally, the proposed algorithm is robust to some extent, because its ups and downs are tolerable, varying over a relatively wide range.

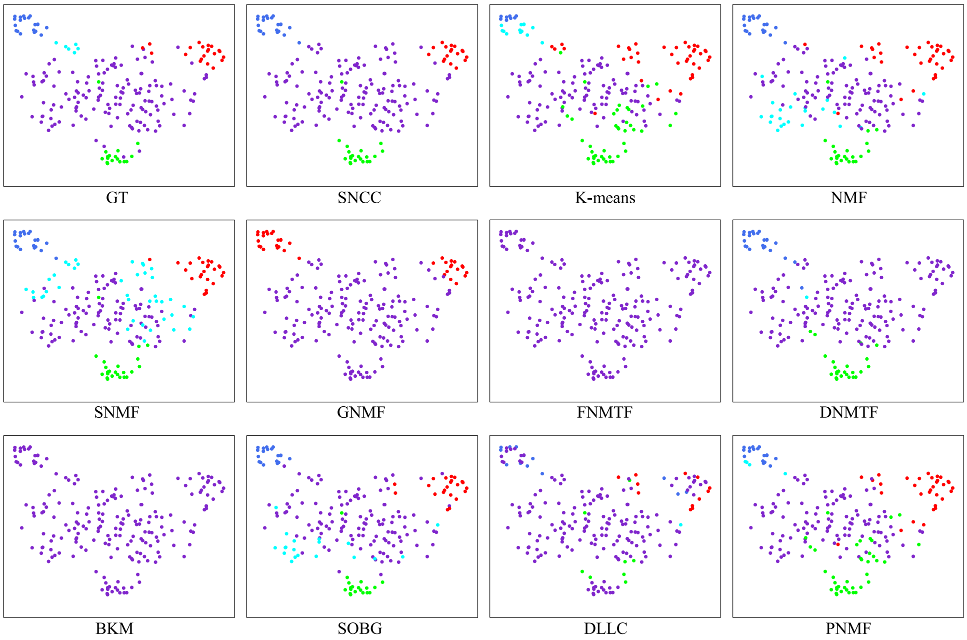


Fig. 1. Visualization for algorithm comparison on LUNG dataset.

Table 2
Accuracy (ACC% ± std%) for algorithm comparison on multiple datasets.

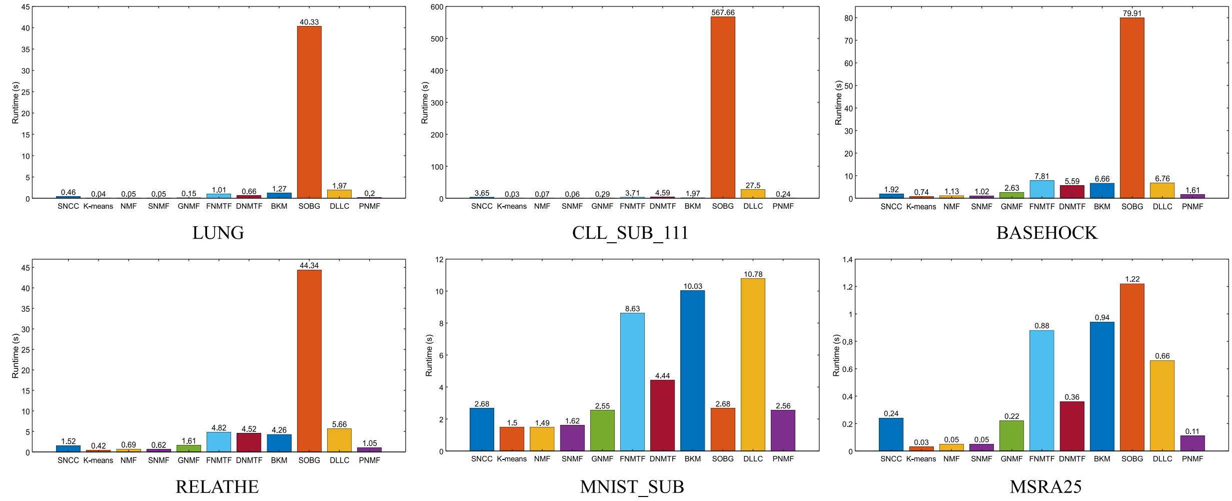
Dataset	LUNG	CLL_SUB_111	BASEHOCK	RELATHE	MNIST_SUB	MSRA25
SNCC	**84.5±6.6**	**54.7±0.9**	**69.0±2.9**	**57.3±4.0**	**53.9±5.1**	**52.6±5.3**
K-means	69.3±9.8	44.1±8.4	62.8±2.7	56.2±2.6	52.4±4.3	47.3±3.2
NMF	74.6±9.3	45.8±8.3	64.4±0.6	56.4±1.6	50.5±2.3	47.2±3.7
SNMF	73.6±9.5	45.9±8.5	64.3±1.0	56.7±1.9	38.8±6.5	47.9±3.0
GNMF	75.8±6.1	47.8±3.8	51.3±3.0	54.9±0.4	23.5±3.9	39.8±6.0
FNMTF	68.1±0.9	46.0±0.2	53.7±0.1	52.1±1.2	41.8±2.5	18.3±2.8
DNMTF	79.5±8.9	50.9±4.8	65.2±9.5	56.9±4.3	37.1±4.7	46.0±3.7
BKM	68.5±0.0	46.0±0.0	52.0±0.1	56.7±0.5	11.3±0.0	10.3±0.0
SOBG	77.8±0.0	51.4±0.0	50.1±0.0	54.8±0.0	11.4±0.0	18.0±0.0
DLLC	56.3±9.9	48.3±7.9	60.0±1.5	55.2±1.8	47.5±4.1	41.5±3.5
PNMF	76.6±8.7	43.8±8.4	62.4±2.9	56.3±2.2	51.8±3.3	48.2±3.1

Table 3
Normalized mutual information (NMI% ± std%) for algorithm comparison on multiple datasets.

Dataset	LUNG	CLL_SUB_111	BASEHOCK	RELATHE	MNIST_SUB	MSRA25
SNCC	**59.8±8.4**	**21.9±5.6**	**11.4±2.1**	1.7±1.1	**52.8±4.2**	**60.4±6.2**
K-means	52.0±5.8	10.4±10.2	5.2±2.2	1.2±1.0	51.1±2.3	54.4±3.8
NMF	55.5±6.9	14.2±11.2	6.2±0.5	1.1±0.6	46.2±1.0	53.8±3.6
SNMF	52.8±6.3	12.9±10.8	6.2±0.9	1.1±0.7	30.2±7.0	55.3±2.8
GNMF	35.6±13.4	10.2±9.5	0.6±1.3	0.4±0.4	15.5±4.6	44.1±6.6
FNMTF	8.0±5.8	5.7±2.4	0.4±0.0	0.2±0.2	36.3±2.3	14.7±5.3
DNMTF	56.8±8.0	17.9±10.6	10.1±9.3	**1.9±1.9**	31.5±5.7	55.0±3.4
BKM	4.8±0.0	2.9±0.0	0.4±0.0	0.7±0.2	0.1±0.0	0.6±0.0
SOBG	28.9±0.0	16.3±0.0	0.3±0.0	0.2±0.0	0.2±0.0	11.3±0.0
DLLC	23.4±8.3	14.0±11.1	3.1±0.7	0.8±0.7	42.4±2.7	44.8±3.1
PNMF	55.2±5.4	10.6±10.3	5.0±1.7	1.1±0.9	49.5±1.5	56.0±4.1

Table 4
Adjusted Rand index (ARI% ± std%) for algorithm comparison on multiple datasets.

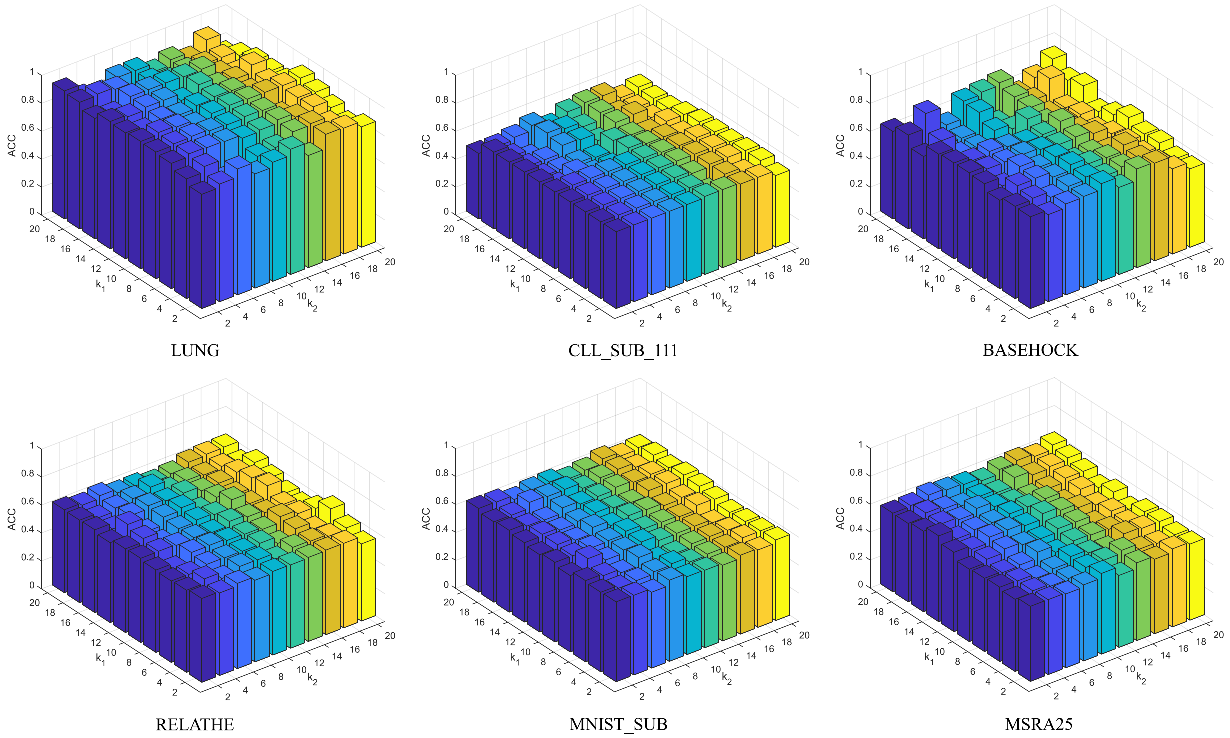
Dataset	LUNG	CLL_SUB_111	BASEHOCK	RELATHE	MNIST_SUB	MSRA25
SNCC	**63.0±18.4**	10.9±2.8	**14.6±3.5**	2.2±3.1	**39.5±5.1**	**33.4±8.0**
K-means	41.8±12.6	2.4±7.2	6.8±3.0	1.6±1.2	37.7±3.2	31.0±4.5
NMF	48.7±13.8	4.7±7.3	8.3±0.7	1.6±0.8	34.0±1.2	30.2±3.8
SNMF	46.1±13.1	4.0±7.5	8.2±1.1	0.2±0.3	20.0±5.7	32.6±3.2
GNMF	38.6±15.4	5.1±9.0	0.4±1.4	0.2±0.3	6.8±3.2	22.8±5.8
FNMTF	3.9±6.9	−0.9±0.7	0.5±0.0	0.2±0.4	24.8±2.1	6.6±2.0
DNMTF	59.1±16.4	7.9±6.7	11.1±9.4	**2.5±2.5**	17.7±4.2	32.2±3.5
BKM	0.0±0.0	0.0±0.0	0.1±0.0	1.3±0.3	0.0±0.0	0.0±0.0
SOBG	33.4±0.0	**16.0±0.0**	0.0±0.0	0.1±0.0	0.0±0.0	1.3±0.0
DLLC	18.4±14.1	5.7±7.8	4.0±1.0	1.0±0.8	30.4±3.2	23.7±3.4
PNMF	50.5±11.6	2.5±7.2	6.4±2.6	1.6±1.1	36.6±2.0	33.2±4.8



LUNG CLL_SUB_111 BASEHOCK

RELATHE MNIST_SUB MSRA25

Fig. 2. Runtime for algorithm comparison on multiple datasets.



LUNG CLL_SUB_111 BASEHOCK

RELATHE MNIST_SUB MSRA25

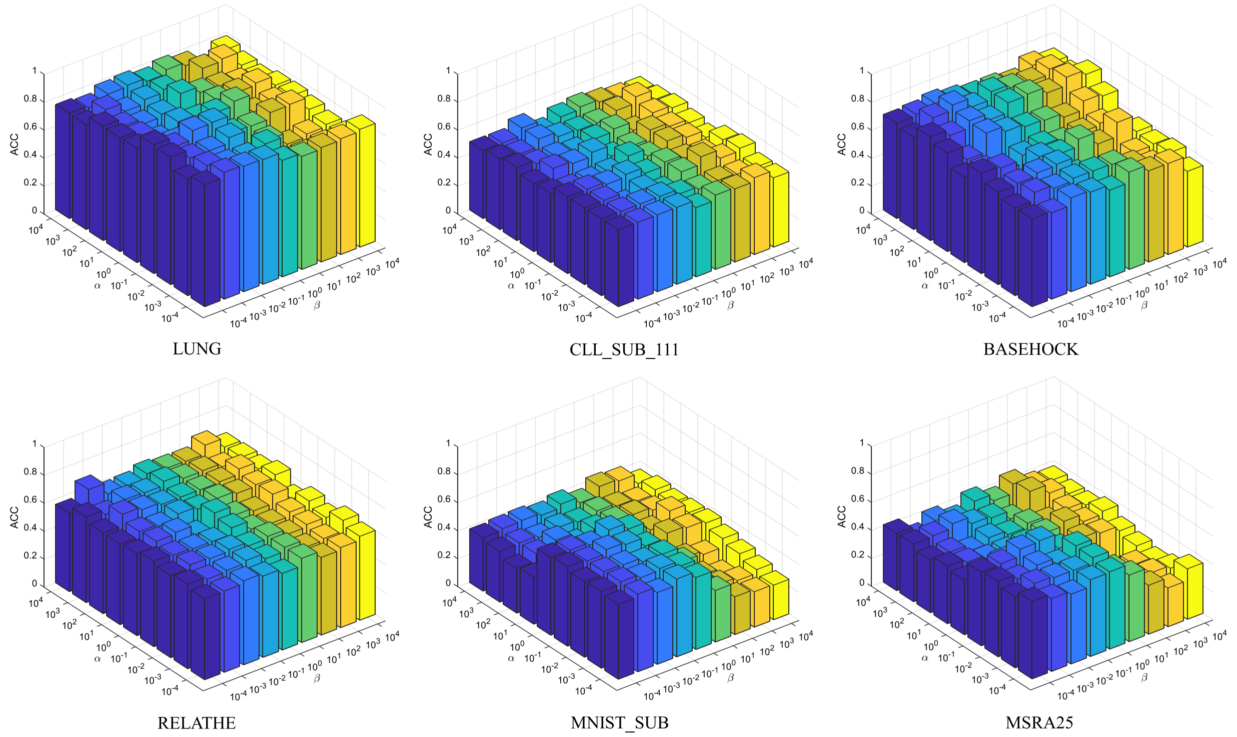
Fig. 3. Accuracy variations with parameters k_1 and k_2.

4.6. Discussion on affinities

This subsection investigates how to construct a similarity graph. As can be seen in Table 5, the performance of a sparse graph is similar to that of a dense graph, sometimes even better. Furthermore, Gaussian kernel and cosine similarity are good at images and texts respectively, whereas binary weighting consistently performs well for all kinds of data. In general, binary weighting is a good choice, with less computation and respectable performance.

5. Conclusions and future works

In this paper, a competitive co-clustering method is proposed. It holds coherence between data affinity and label assignment, based on nonnegative matrix tri-factorization appended two regularizers. For objective optimization, a multiplicative alternating rule is adopted, whose convergence is theoretically guaranteed. Furthermore, experiments demonstrate that our algorithm is effective and efficient, compared with others. In future work, better ways will be investigated to initialize parameters and construct affinities, considering the importance of initialization and affinity for robustness and effectiveness.



LUNG CLL_SUB_111 BASEHOCK

RELATHE MNIST_SUB MSRA25

Fig. 4. Accuracy variations with parameters α and β.

Table 5
Performance under multiple affinities and datasets.

Dataset	Metric	Binary	Cosine		Gaussian	
			Dense	Sparse	Dense	Sparse
LUNG	ACC	**92.1**	88.2	88.7	91.1	83.7
	NMI	72.3	64.1	68.3	**73.4**	58.7
	ARI	**81.7**	69.8	72.1	78.4	60.9
CLL_SUB_111	ACC	**55.0**	45.9	**55.0**	**55.0**	54.1
	NMI	**26.3**	12.9	23.9	23.9	21.5
	ARI	**12.4**	5.8	12.0	12.0	10.5
BASEHOCK	ACC	67.3	62.5	**69.5**	67.4	66.3
	NMI	11.3	4.8	**11.4**	9.1	8.3
	ARI	12.1	6.2	**15.2**	12.1	10.6
RELATHE	ACC	**60.8**	60.1	59.4	60.6	58.2
	NMI	**3.6**	2.9	2.6	3.1	2.0
	ARI	4.2	3.9	3.5	**4.4**	2.6
MNIST_SUB	ACC	56.5	51.4	23.4	56.1	**58.6**
	NMI	53.6	48.3	12.7	53.5	**54.8**
	ARI	40.5	35.2	6.2	40.4	**43.4**
MSRA25	ACC	**60.6**	53.4	52.0	52.6	56.7
	NMI	**64.9**	59.9	62.5	60.0	64.2
	ARI	41.9	38.5	39.6	38.5	**43.9**

CRediT authorship contribution statement

Zhoumin Lu: Conceptualization, Methodology, Software, Investigation, Visualization, Writing - original draft. **Genggeng Liu:** Data curation, Validation, Writing - review & editing. **Shiping Wang:** Resources, Formal analysis, Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (Nos. U1705262 and 61672159), the Technology Innovation Platform Project of Fujian Province under Grant (Nos. 2014H2005 and 2009J1007), the Fujian Collaborative Innovation Center for Big Data Application in Governments, the Fujian Engineering Research Center of Big Data Analysis and Processing.

Appendix A. Proof of Theorem 1

Proof. Firstly, $J(F)$ can be rewritten as

$$\begin{aligned} J(F) = &-\operatorname{Tr}(X^T FSG^T) + \frac{1}{2}\operatorname{Tr}(GS^T F^T FSG^T) - \alpha \operatorname{Tr}(M^+ F^T) \\ &+ \alpha \operatorname{Tr}(M^- F^T) + \frac{\alpha}{2}\operatorname{Tr}(FN^+ F^T) - \frac{\alpha}{2}\operatorname{Tr}(FN^- F^T) \end{aligned} \tag{A.1}$$

Complied with Lemma 2, we observe

$$\operatorname{Tr}(GS^T F^T FSG^T) = \operatorname{Tr}(F^T FSG^T GS^T) \leq \sum_{i=1}^{d}\sum_{j=1}^{c_1} \frac{(F'SG^T GS^T)_{ij} F_{ij}^2}{F'_{ij}} \tag{A.2}$$

$$\operatorname{Tr}(FN^+ F^T) = \operatorname{Tr}(F^T FN^+) \leq \sum_{i=1}^{d}\sum_{j=1}^{c_1} \frac{(F'N^+)_{ij} F_{ij}^2}{F'_{ij}} \tag{A.3}$$

Subject to the inequality: $\forall a, b > 0, a \leq \frac{a^2+b^2}{2b}$, we notice

$$\operatorname{Tr}(M^- F^T) = \sum_{i=1}^{d}\sum_{j=1}^{c_1} M^-_{ij} F_{ij} \leq \sum_{i=1}^{d}\sum_{j=1}^{c_1} M^-_{ij} \frac{F_{ij}^2 + F'^2_{ij}}{2F'_{ij}} \tag{A.4}$$

Abiding by the inequality: $\forall z > 0, z \geq 1 + \log z$, we obtain

$$\begin{aligned}\mathrm{Tr}(X^TFSG^T) = \mathrm{Tr}(SG^TX^TF) &= \sum_{i=1}^{d}\sum_{j=1}^{c_1}(XGS^T)_{ij}F_{ij}\\ &= \sum_{i=1}^{d}\sum_{j=1}^{c_1}(XGS^T)_{ij}F'_{ij}\frac{F_{ij}}{F'_{ij}}\\ &\geq \sum_{i=1}^{d}\sum_{j=1}^{c_1}(XGS^T)_{ij}F'_{ij}(1+\log\frac{F_{ij}}{F'_{ij}})\end{aligned} \tag{A.5}$$

$$\begin{aligned}\mathrm{Tr}(M^+F^T) &= \sum_{i=1}^{d}\sum_{j=1}^{c_1}M^+_{ij}F_{ij} = \sum_{i=1}^{d}\sum_{j=1}^{c_1}M^+_{ij}F'_{ij}\frac{F_{ij}}{F'_{ij}}\\ &\geq \sum_{i=1}^{d}\sum_{j=1}^{c_1}M^+_{ij}F'_{ij}(1+\log\frac{F_{ij}}{F'_{ij}})\end{aligned} \tag{A.6}$$

$$\begin{aligned}\mathrm{Tr}(FN^-F^T) = \mathrm{Tr}(N^-F^TF) &= \sum_{i=1}^{d}\sum_{j=1}^{c_1}\sum_{k=1}^{c_1}N^-_{jk}F_{ij}F_{ik}\\ &= \sum_{i=1}^{d}\sum_{j=1}^{c_1}\sum_{k=1}^{c_1}N^-_{jk}F'_{ij}F'_{ik}\frac{F_{ij}F_{ik}}{F'_{ij}F'_{ik}}\\ &\geq \sum_{i=1}^{d}\sum_{j=1}^{c_1}\sum_{k=1}^{c_1}N^-_{jk}F'_{ij}F'_{ik}(1+\log\frac{F_{ij}F_{ik}}{F'_{ij}F'_{ik}})\end{aligned} \tag{A.7}$$

Obviously, $Z(F, F')$ consists of all the bounds and satisfies Definition 1. Consequently, $Z(F, F')$ is an auxiliary function for $J(F)$.

To minimize $Z(F, F')$, we take

$$\begin{aligned}\frac{\partial Z(F,F')}{\partial F_{ij}} = &-\frac{(XGS^T)_{ij}F'_{ij}}{F_{ij}} + \frac{(F'SG^TGS^T)_{ij}F_{ij}}{F'_{ij}} - \alpha\frac{M^+_{ij}F'_{ij}}{F_{ij}}\\ &+ \alpha\frac{M^-_{ij}F_{ij}}{F'_{ij}} + \alpha\frac{(F'N^+)_{ij}F_{ij}}{F'_{ij}} - \alpha\frac{(F'N^-)_{ij}F'_{ij}}{F_{ij}}\end{aligned} \tag{A.8}$$

and obtain its Hessian matrix

$$\begin{aligned}\frac{\partial^2 Z(F,F')}{\partial F_{ij}\partial F_{kl}} = &\delta_{ik}\delta_{jl}(\frac{(XGS^T)_{ij}F'_{ij}}{F^2_{ij}} + \frac{(F'SG^TGS^T)_{ij}}{F'_{ij}} + \alpha\frac{M^+_{ij}F'_{ij}}{F^2_{ij}}\\ &+ \alpha\frac{M^-_{ij}}{F'_{ij}} + \alpha\frac{(F'N^+)_{ij}}{F'_{ij}} + \alpha\frac{(F'N^-)_{ij}F'_{ij}}{F^2_{ij}})\end{aligned} \tag{A.9}$$

where $\delta_{ik} = 1$ if and only if $i = k$, otherwise, $\delta_{ik} = 0$. Furthermore, the Hessian matrix is a diagonal matrix with positive elements, implying that $Z(F, F')$ is convex and $\arg\min_{F_{ij}} J(F) = \arg\min_{F_{ij}} Z(F, F')$ by setting $\frac{\partial Z(F,F')}{\partial F_{ij}} = 0$. □

Appendix B. Proof of Theorem 2

Proof. Firstly, $J(S)$ can be rewritten as

$$J(S) = -\mathrm{Tr}(X^TFSG^T) + \frac{1}{2}\mathrm{Tr}(GS^TF^TFSG^T) \tag{B.1}$$

Complied with Lemma 2, we observe

$$\mathrm{Tr}(GS^TF^TFSG^T) = \mathrm{Tr}(S^TF^TFSG^TG) \leq \sum_{i=1}^{c_1}\sum_{j=1}^{c_2}\frac{(F^TFS'G^TG)_{ij}S^2_{ij}}{S'_{ij}} \tag{B.2}$$

Abiding by the inequality: $\forall z > 0, z \geq 1 + \log z$, we obtain

$$\begin{aligned}\mathrm{Tr}(X^TFSG^T) = \mathrm{Tr}(G^TX^TFS) &= \sum_{i=1}^{c_1}\sum_{j=1}^{c_2}(F^TXG)_{ij}S_{ij}\\ &= \sum_{i=1}^{c_1}\sum_{j=1}^{c_2}(F^TXG)_{ij}S'_{ij}\frac{S_{ij}}{S'_{ij}}\\ &\geq \sum_{i=1}^{c_1}\sum_{j=1}^{c_2}(F^TXG)_{ij}S'_{ij}(1+\log\frac{S_{ij}}{S'_{ij}})\end{aligned} \tag{B.3}$$

Obviously, $Z(S, S')$ consists of all the bounds and satisfies Definition 1. Consequently, $Z(S, S')$ is an auxiliary function for $J(S)$.

To minimize $Z(S, S')$, we take

$$\frac{\partial Z(S,S')}{\partial S_{ij}} = -\frac{(F^TXG)_{ij}S'_{ij}}{S_{ij}} + \frac{(F^TFS'G^TG)_{ij}S_{ij}}{S'_{ij}} \tag{B.4}$$

and obtain its Hessian matrix

$$\frac{\partial^2 Z(S,S')}{\partial S_{ij}\partial S_{kl}} = \delta_{ik}\delta_{jl}(\frac{(F^TXG)_{ij}S'_{ij}}{S^2_{ij}} + \frac{(F^TFS'G^TG)_{ij}}{S'_{ij}} \tag{B.5}$$

where $\delta_{ik} = 1$ if and only if $i = k$, otherwise, $\delta_{ik} = 0$. Furthermore, the Hessian matrix is a diagonal matrix with positive elements, implying that $Z(S, S')$ is convex and $\arg\min_{S_{ij}} J(S) = \arg\min_{S_{ij}} Z(S, S')$ by setting $\frac{\partial Z(S,S')}{\partial S_{ij}} = 0$. □

Appendix C. Proof of Theorem 3

Proof. Firstly, $J(G)$ can be rewritten as

$$\begin{aligned}J(G) = &-\mathrm{Tr}(X^TFSG^T) + \frac{1}{2}\mathrm{Tr}(GS^TF^TFSG^T) - \beta\,\mathrm{Tr}(P^+G^T)\\ &+ \beta\,\mathrm{Tr}(P^-G^T) + \frac{\beta}{2}\mathrm{Tr}(GQ^+G^T) - \frac{\beta}{2}\mathrm{Tr}(GQ^-G^T)\end{aligned} \tag{C.1}$$

In the same way as the proof of Theorem 1, we obtain the following inequalities,

$$\mathrm{Tr}(GS^TF^TFSG^T) \leq \sum_{i=1}^{n}\sum_{j=1}^{c_2}\frac{(G'S^TF^TFS)_{ij}G^2_{ij}}{G'_{ij}} \tag{C.2}$$

$$\mathrm{Tr}(GQ^+G^T) \leq \sum_{i=1}^{n}\sum_{j=1}^{c_2}\frac{(G'Q^+)_{ij}G^2_{ij}}{G'_{ij}} \tag{C.3}$$

$$\mathrm{Tr}(P^-G^T) \leq \sum_{i=1}^{n}\sum_{j=1}^{c_2}P^-_{ij}\frac{G^2_{ij}+G'^2_{ij}}{2G'_{ij}} \tag{C.4}$$

$$\mathrm{Tr}(X^TFSG^T) \geq \sum_{i=1}^{n}\sum_{j=1}^{c_2}(X^TFS)_{ij}G'_{ij}(1+\log\frac{G_{ij}}{G'_{ij}}) \tag{C.5}$$

$$\mathrm{Tr}(P^+G^T) \geq \sum_{i=1}^{n}\sum_{j=1}^{c_2}P^+_{ij}G'_{ij}(1+\log\frac{G_{ij}}{G'_{ij}}) \tag{C.6}$$

$$\mathrm{Tr}(GQ^-G^T) \geq \sum_{i=1}^{n}\sum_{j=1}^{c_2}\sum_{k=1}^{c_2}Q^-_{jk}G'_{ij}G'_{ik}(1+\log\frac{G_{ij}G_{ik}}{G'_{ij}G'_{ik}}) \tag{C.7}$$

Obviously, $Z(G, G')$ consists of all the bounds and satisfies Definition 1. Consequently, $Z(G, G')$ is an auxiliary function for $J(G)$. Furthermore, its first- and second-order derivatives are shown as follows:

$$\begin{aligned}\frac{\partial Z(G,G')}{\partial G_{ij}} = &-\frac{(X^TFS)_{ij}G'_{ij}}{G_{ij}} + \frac{(G'S^TF^TFS)_{ij}G_{ij}}{G'_{ij}} - \beta\frac{P^+_{ij}G'_{ij}}{G_{ij}}\\ &+ \beta\frac{P^-_{ij}G_{ij}}{G'_{ij}} + \beta\frac{(G'Q^+)_{ij}G_{ij}}{G'_{ij}} - \beta\frac{(G'Q^-)_{ij}G'_{ij}}{G_{ij}}\end{aligned} \tag{C.8}$$

$$\begin{aligned}\frac{\partial^2 Z(G,G')}{\partial G_{ij}\partial G_{kl}} =&\delta_{ik}\delta_{jl}(\frac{(X^TFS)_{ij}G'_{ij}}{G^2_{ij}}+\frac{(G'S^TF^TFS)_{ij}}{G'_{ij}}+\beta\frac{P^+_{ij}G'_{ij}}{G^2_{ij}}\\ &+\beta\frac{P^-_{ij}}{G'_{ij}}+\beta\frac{(G'Q^+)_{ij}}{G'_{ij}}+\beta\frac{(G'Q^-)_{ij}G'_{ij}}{G^2_{ij}})\end{aligned} \tag{C.9}$$

It means that $Z(G,G')$ is convex and $\arg\min_{G_{ij}} J(G) = \arg\min_{G_{ij}} Z(G,G')$ by setting $\frac{\partial Z(G,G')}{\partial G_{ij}} = 0$. □

References

[1] L. Hagen, A. Kahng, New spectral methods for ratio cut partitioning and clustering, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 11 (9) (1992) 1074–1085.

[2] C. Ding, X. He, H. Zha, M. Gu, H. Simon, A min-max cut algorithm for graph partitioning and data clustering, in: Proceedings of the IEEE International Conference on Data Mining, 2001, pp. 107–114.

[3] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[4] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.

[5] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.

[6] G. Huang, J. Zhang, S. Song, Z. Chen, Maximin separation probability clustering, in: Proceeding of the AAAI Conference on Artificial Intelligence, 2015, pp. 2680–2686.

[7] R.C.D. Amorim, B. Mirkin, Minkowski Metric, feature weighting and anomalous cluster initializing in K-means clustering, Pattern Recognit. 45 (3) (2012) 1061–1075.

[8] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, G. Ridgeway, Learning a mahalanobis metric from equivalence constraints, J. Mach. Learn. Res. 6 (2005) 937–965.

[9] J.A. Hartigan, M.A. Wong, A K-means clustering algorithm, J. R. Stat. Soc. 28 (1) (1979) 100–108.

[10] D. Arthur, S. Vassilvitskii, K-means++: the advantages of careful seeding, in: Proceedings of the Annual ACM SIAM Symposium on Discrete Algorithms, 2007, pp. 1027–1035.

[11] I. Dhillon, Y. Guan, B. Kulis, Kernel k-means, spectral clustering and normalized cuts, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 551–556.

[12] S. Yu, L. Tranchevent, X. Liu, W. Glänzel, J.A.K. Suykens, B.D. Moor, Y. Moreau, Optimized data fusion for kernel k-means clustering, IEEE Trans. Pattern Anal. Mach. Intell. 34 (5) (2012) 1031–1039.

[13] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, J. Cybern. 3 (3) (1973) 32–57.

[14] R. Zhang, X. Li, H. Zhang, F. Nie, Deep fuzzy K-means with adaptive loss and entropy regularization, IEEE Trans. Fuzzy Syst. 20 (10) (2019) 1–11.

[15] C. Ding, X. He, H.D. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, in: Proceedings of the SIAM International Conference on Data Mining, pp. 606–610.

[16] T. Li, C. Ding, The relationships among various nonnegative matrix factorization methods for clustering, in: Proceedings of the IEEE International Conference on Data Mining, 2006, pp. 362–371.

[17] M. Rege, M. Dong, F. Fotouhi, Co-clustering documents and words using bipartite isoperimetric graph partitioning, in: Proceedings of the IEEE International Conference on Data Mining, 2006, 532–541.

[18] R. Boutalbi, L. Labiod, M. Nadif, Co-clustering from tensor data, in: Proceedings of the Pacific Asia Conference on Knowledge Discovery and Data Mining, 2019, pp. 370–383.

[19] I. Konstas, V. Stathopoulos, J.M. Jose, On social networks and collaborative recommendation, in: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2009, pp. 195–202.

[20] J. Jacques, C. Biernacki, Model-based co-clustering for ordinal data, Comput. Statist. Data Anal. 123 (2) (2018) 101–115.

[21] Y.B. Slimen, S. Allio, J. Jacques, Model-based co-clustering for functional data, Neurocomputing 291 (2) (2018) 97–108.

[22] E. Lima, W. Shi, X. Liu, Q. Yu, Integrating multi-level tag recommendation with external knowledge bases for automatic question answering, ACM Trans. Internet Technol. 19 (3) (2019) 1–22.

[23] Y. Liu, Q. Gu, J.P. Hou, J. Han, J. Ma, A network-assisted co-clustering algorithm to discover cancer subtypes based on gene expression, BMC Bioinformatics 15 (37) (2014) 1–11.

[24] Y. Chen, M. Dong, W. Wan, Image co-clustering with multi-modality features and user feedbacks, in: Proceedings of the ACM International Conference on Multimedia, 2009, pp. 689–692.

[25] J. Sun, Z. Wang, F. Sun, H. Li, Sparse dual graph-regularized NMF for image co-clustering, Neurocomputing 316 (1) (2018) 156–165.

[26] Q. Gu, J. Zhou, Co-clustering on manifolds, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 359–368.

[27] S. Huang, H. Wang, D. Li, Y. Yang, T. Li, Spectral co-clustering ensemble, Knowl.-Based Syst. 84 (2015) 46–55.

[28] N.D. Buono, G. Pio, Non-negative matrix tri-factorization for co-clustering: An analysis of the block matrix, Inf. Sci. 301 (2015) 13–26.

[29] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.

[30] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Proceedings of the Annual Conference on Neural Information Processing Systems, 2000, pp. 556–562.

[31] C. Ding, T. Li, M. Jordan, Convex and semi-nonnegative matrix factorizations, IEEE Trans. Pattern Anal. Mach. Intell. 32 (1) (2010) 45–55.

[32] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.

[33] F. Shang, L. Jiao, F. Wang, Graph dual regularization non-negative matrix factorization for co-clustering, Pattern Recognit. 45 (6) (2012) 2237–2250.

[34] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 126–135.

[35] S. Wang, W. Guo, Robust co-clustering via dual local learning and high-order matrix factorization, Knowl.-Based Syst. 138 (2017) 176–187.

[36] S. Wang, A. Huang, Penalized nonnegative matrix tri-factorization for co-clustering, Expert Syst. Appl. 78 (2017) 64–73.

[37] M. Wu, B. Schölkopf, A local learning approach for clustering, in: Proceedings of the Annual Conference on Neural Information Processing Systems, 2006, pp. 1529–1536.

[38] X. He, P. Niyogi, Locality preserving projections, in: Proceedings of the Annual Conference on Neural Information Processing Systems, 2003, pp. 153–160.

[39] H. Wang, F. Nie, H. Huang, F. Makedon, Fast nonnegative matrix tri-factorization for large-scale data co-clustering, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2011, pp. 1553–1558.

[40] J. Han, K. Song, F. Nie, X. Li, Bilateral K-Means algorithm for fast co-clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 1969–1975.

[41] F. Nie, X. Wang, C. Deng, H. Huang, Learning a structured optimal bipartite graph for co-clustering, in: Proceedings of the Annual Conference on Neural Information Processing Systems, 2017, pp. 4129–4138.

[42] L. van der Maaten, G. Hintton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (1) (2008) 2579–2625.