Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Bilateral k-Means Algorithm for Fast Co-Clustering

Junwei Han,[†*] Kun Song,[†] Feiping Nie,[‡*] Xuelong Li[§]
[†]School of Automation, Northwestern Polytechnical University, Xi'an, 710072, Shaanxi, P. R. China
[‡]School of Computer Science and Center for OPTIMAL, Northwestern Polytechnical University, Xi'an, 710072, P. R. China
[§]Center for OPTIMAL, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute
of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China.
{junweihan2010, songkun123000, feipingnie}@gmail.com, xuelong_li@ieee.org

Abstract

With the development of the information technology, the amount of data, e.g. text, image and video, has been increased rapidly. Efficiently clustering those large scale data sets is a challenge. To address this problem, this paper proposes a novel co-clustering method named bilateral k-means algorithm (BKM) for fast co-clustering. Different from traditional k-means algorithms, the proposed method has two indicator matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ and a diagonal matrix $\boldsymbol{S}$ to be solved, which represent the cluster memberships of samples and features, and the co-cluster centres, respectively. Therefore, it could implement different clustering tasks on the samples and features simultaneously. We also introduce an effective approach to solve the proposed method, which involves less multiplication. The computational complexity is analyzed. Extensive experiments on various types of data sets are conducted. Compared with the state-of-the-art clustering methods, the proposed BKM not only has faster computational speed, but also achieves promising clustering results.

Introduction

Co-clustering, also referred as biclustering, seeks to cluster the set of samples and the set of features in a data matrix simultaneously. By doing permutations of rows and columns, the co-clustering algorithms aim to reorganize the initial data matrix into homogeneous blocks. These blocks also called co-clusters can therefore be defined as subsets of the data matrix characterized by a set of observations and a set of features whose elements are similar. The surveys of types of co-cluster and the related co-clustering algorithms are referred in literatures (Madeira and Oliveira 2004; Tanay, Sharan, and Shamir 2005; Charrad and Ahmed 2011). Since co-clustering algorithms utilize the relations between samples clusters and feature clusters, they make the data sets more predictable and the co-clustering performance more excellent compared with traditional one-side clustering.

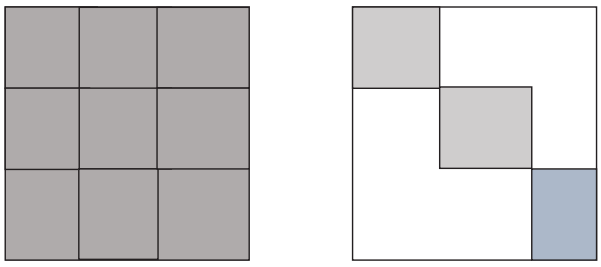
Recently, several research topics involving co-clustering become hot issues due to the development of internet technology. Examples include text data mining, image retrieving, image segmentation, and video analysis (Zhang et al. 2004; Lu, Yuan, and Yan 2013; Lu, Wu, and Yuan 2014; Ailem, Role, and Nadif 2015; Liu and Lin 2015; Han et al. 2015; Cheng et al. 2015). In those tasks, the data sets are often represented by sparse matrices, and most of the related co-clustering methods (Ding, Li, and Jordan 2010; Ding et al. 2006) are based on checkerboard co-cluster model (as illustrate in Figure 1). In fact, the checkerboard co-cluster model is not quit suitable for those applications. The checkerboard co-cluster supposes every element in the data matrix should belong to one co-cluster, however, much of the entries of sparse matrix equals to zero which contribute none to the co-clustering. In real applications, the sparse data matrices are often mixed with noises which may change the values of elements equaling zero. It would reduce the co-clustering performance greatly. Therefore, according to the duality of the sample clustering and features clustering, using the diagonal co-cluster structure (as illustrate in Figure 1.) instead of the checkerboard co-cluster model in those applications is a reasonable choice.

Among the diagonal co-cluster based algorithms, bipartite spectral graph partition (BSGP) approach (Inderjit 2001) is the most famous one due to its notable performance. Its variants have been applied in many fields (Zhang et al. 2004; Li, Wu, and Chang 2012; Trivedi et al. 2012). However, BSGP is computationally prohibitive for large data collections since it involves the singular value decomposition in the solution process, which severely limits the range of BSGP in the real world applications. In addition, BSGP is based on bipartitioning Ncuts of a spectral graph. When dealing with multipartitioning problem, it needs to relax the origin discrete problem to a continuous problem firstly, then use k-means algorithm to output discrete result. In this discrete-continuous-discrete transformation procedure, the final resulting optimization problem is much deviate from the original objective problem of BSGP, and the performance of BSGP would be damaged.

In this paper, we propose a novel co-clustering method named bilateral k-means algorithm (BKM), by discovering the shortages of BSGP. Our proposed method is interesting from following perspectives:

1. Different from BSGP transforming the optimization problem of minimizing normalized cuts into a continuous relaxation, BKM relaxes the minimum normalized cuts problem to a special non-negative matrix decomposition (NMF) which involves constraints of indicator matrices. Such indi-

*Corresponding authors.



(a) (b)

Figure 1: Two types of the co-cluster model, the dark area are co-clusters. (a) checkerboard co-cluster model, assuming every element in the data matrix should belong to one co-cluster; (b) diagonal co-cluster model, only considering parts of elements in the data matrix to form the co-clusters.

cators store the clustering results of columns and rows. However, many existing clustering methods, such as BSGP and NMF, need a post-processing steps to output the clustering result.

2. The optimization problem of BKM, is decomposed into three subproblems and solved in an alternative way. In each iteration, it involves a subproblem with much smaller sizes, which consists of much less matrix multiplications. Thus, our approach is computationally efficient and performs well in large-scale data collections.

3. Extensive experiments on synthetic data sets and real world data sets have been conducted. Compared with state-of-the-art methods, our method not only has a lower computational complexity, but also achieves a competitive clustering performance.

Notations and problem formulation. In this paper, we write matrices as boldface uppercase letters and vectors as boldface lowercase letters.

Given a data matrix $\boldsymbol{X} \in R^{d\times n}$, in which (i,j)-element is x_{ij}, its i-th row and j-th column are denoted as $\boldsymbol{x}_{i\cdot}$ and $\boldsymbol{x}_{\cdot j}$ respectively. The columns represent samples and the rows represent features. Co-clustering based on diagonal co-cluster aims at grouping the samples $\{\boldsymbol{x}_{\cdot j}\}_{j=1}^{n}$ into c clusters $\{\Theta_l\}_{l=1}^{c}$ and the features $\{\boldsymbol{x}_{j\cdot}\}_{j=1}^{d}$ into c clusters $\{\Omega_l\}_{l=1}^{c}$, simultaneously.

Two partition matrices $\boldsymbol{P} = [\boldsymbol{p}_{1\cdot}^T, \cdots, \boldsymbol{p}_{d\cdot}^T]^T \in \{0,1\}^{d\times c}$ and $\boldsymbol{Q} = [\boldsymbol{q}_{1\cdot}^T, \cdots, \boldsymbol{p}_{n\cdot}^T]^T \in \{0,1\}^{n\times c}$ are utilized to represent clustering results of features and samples respectively. If i-th feature $\boldsymbol{x}_{i\cdot}$ belongs to cluster Ω_j, $p_{ij} = 1$, and If i-th sample $\boldsymbol{x}_{\cdot i}$ belongs to cluster Θ_j, $q_{ij} = 1$. In $\boldsymbol{P}$ and $\boldsymbol{Q}$, each row, i.e. $\boldsymbol{p}_{i\cdot}(1 \le i \le d)$ or $\boldsymbol{q}_{i\cdot}(1 \le i \le n)$ has one and only one element equal to 1 which indicates the cluster membership of i-th feature or sample, and the rest elements are 0. Therefore, we call such type of matrices as indicator matrices, and denote the set of them as Φ, thus $\boldsymbol{P} \in \Phi_{d\times c}$ and $\boldsymbol{Q} \in \Phi_{n\times c}$.

Revisit BSGP

Bipartite spectral graph partitioning (BSGP) based co-clustering approach, constructs a bipartite graph $G = \{V, \boldsymbol{A}\}$ between samples and features of data matrix $\boldsymbol{X}$. Here, V is the set of vertices corresponding to samples and features, and $\boldsymbol{A}$ is the adjacency matrix constructed by data matrix $\boldsymbol{X}$ as Eq. (1)

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{X} \\ \boldsymbol{X}^T & \boldsymbol{0} \end{bmatrix} \tag{1}$$

Using the bipartite graph model, the feature cluster Ω_l and sample cluster Φ_l are determined as follows.

$$\Omega_l = \{x_{i\cdot} : \sum_{j\in\Theta_l} X_{ij} \ge \sum_{j\in\Theta_k} X_{ij}, \forall k = 1, \cdots, c\} \tag{2}$$

$$\Theta_l = \{x_{\cdot j} : \sum_{i\in\Omega_l} X_{ij} \ge \sum_{i\in\Omega_k} X_{ij}, \forall k = 1, \cdots, c\} \tag{3}$$

It is easy to see that there is a recursive relationship between Ω_l and Θ_l, and the relations described in Eq.(2) and Eq.(3) determine BSGP is based on diagonal co-cluster structure.

Then, BSGP aims to finding the minimal Ncuts of G by solving the optimization problem in Eq. (4)

$$\min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T\boldsymbol{L}\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y}}, s.t. \boldsymbol{y} \in \{-1,1\}^{(d+n)\times 1} \tag{4}$$

where $\boldsymbol{D}$ is the diagonal "degree" matrix with $\boldsymbol{D}_{ii} = \sum_k \boldsymbol{A}_{ik}$, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. $\boldsymbol{y} = [\boldsymbol{p}^T, \boldsymbol{q}^T]^T, \boldsymbol{p} \in \{1,-1\}^{d\times 1}$ stores the membership of features, and $\boldsymbol{q} \in \{-1,1\}^{n\times 1}$ stores the membership of samples. When $\boldsymbol{p}_i = 1$, the i-th feature belongs to the first feature cluster, otherwise, it belongs to the second feature cluster. Similarly, when $\boldsymbol{q}_i = 1$, the i-th sample belongs to the first sample cluster, otherwise, it belongs to the second sample cluster.

Since the problem in Eq. (4) is a NP-complete problem, it is relaxed into Eq. (5)

$$\min_{\boldsymbol{q}\neq 0} \frac{\boldsymbol{q}^T\boldsymbol{L}\boldsymbol{q}}{\boldsymbol{q}^T\boldsymbol{D}\boldsymbol{q}}, s.t. \boldsymbol{q}^T\boldsymbol{D}\boldsymbol{e} = 0 \tag{5}$$

where $\boldsymbol{e}$ is a $(d+n)\times 1$ matrix with all elements equal to 1. Considering the structure of data matrix $\boldsymbol{A}$, the problems in Eq. (5) can be solved by calculating singular vector corresponding to the second smallest singular value of matrix $\boldsymbol{D}_1^{-1/2}\boldsymbol{X}\boldsymbol{D}_2^{-1/2}$, where $\boldsymbol{D}_1 \in R^{d\times d}$ and $\boldsymbol{D}_2 \in R^{n\times n}$ satisfy

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_1 & \\ & \boldsymbol{D}_2 \end{bmatrix} \tag{6}$$

The procedure mentioned above is about bipartition clustering. When dealing with multipartitioning problem, suppose there are c clusters in the data set, it needs to calculate $l = log_2(c)$ left vectors $\boldsymbol{u}_2, \cdots, \boldsymbol{u}_{l+1}$ and l right singular vectors $\boldsymbol{v}_2, \cdots, \boldsymbol{v}_{l+1}$, which is deemed to contain c-modal information about the data set. Thus we can form a l-dimensional data set

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{D}_1^{-1/2}\boldsymbol{U} \\ \boldsymbol{D}_2^{-1/2}\boldsymbol{V} \end{bmatrix} \tag{7}$$

where $\boldsymbol{U} = [\boldsymbol{u}_2, \cdots, \boldsymbol{u}_{l+1}]$, $\boldsymbol{V} = [\boldsymbol{v}_2, \cdots, \boldsymbol{v}_{l+1}]$. Lastly, the k-means algorithm is required to cluster the new data set $\boldsymbol{Z}$, and output the clustering results, in which the first d results of $\boldsymbol{Z}$ correspond to the feature clustering results of $\boldsymbol{X}$, and the last n results correspond to sample clustering results.

Bilateral K-means algorithm

In this section, we replace the bipartitioning normalized cuts in BSGP by multipartitioning normalized cuts, and derive the formulation of the proposed method. Then, an efficient algorithm for solving the proposed optimization problem is introduced.

Formulation of bilateral k-means (BKM) algorithm

We now apply the multipartitioning Ncuts into BSGP. The optimization problem in Eq. (4) is transformed to

$$\min_{\boldsymbol{Y}} \sum_{k=1}^{c} \frac{\boldsymbol{y}_k^T \boldsymbol{L} \boldsymbol{y}_k}{\boldsymbol{y}_k^T \boldsymbol{D} \boldsymbol{y}_k}, s.t. \boldsymbol{Y} \in \Phi_{(d+n)\times c} \tag{8}$$

Since 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, optimization problem in Eq. (8) can be transformed to the matrix form as follows.

$$\min_{\boldsymbol{Y}} Tr(\boldsymbol{Y}^T\boldsymbol{L}\boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}), s.t. \boldsymbol{Y} \in \Phi_{(m+n)\times c} \tag{9}$$

where $Tr(\cdot)$ is the matrix trace.

Substituting $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ into the objective function of problem in Eq.(9), there is an equation as follows

$$\begin{aligned} &Tr(\boldsymbol{Y}^T\boldsymbol{L}\boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}) \\ &= Tr(\boldsymbol{Y}^T(\boldsymbol{D}-\boldsymbol{A})\boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}) \\ &= Tr(\boldsymbol{I} - \boldsymbol{Y}^T\boldsymbol{A}\boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}) \end{aligned} \tag{10}$$

where I is an identity matrix.

As bipartite graph G is constructed between samples and features, the indicator matrix $\boldsymbol{Y}$ can be rewritten as $\boldsymbol{Y}^T = [\boldsymbol{P}^T, \boldsymbol{Q}^T]$. $\boldsymbol{P}$ contains the membership of features and $\boldsymbol{Q}$ contains membership of samples. Substituting A defined in Eq.(3) into Eq. (10), There is $Tr(\boldsymbol{Y}^T\boldsymbol{L}\boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}) = Tr(\boldsymbol{I} - 2\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{Q}(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1})$. Thus, the optimization problem in Eq. (9) is transformed into

$$\begin{aligned} &\min_{\boldsymbol{P},\boldsymbol{Q}} Tr(-\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{Q}(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}) \\ &s.t. \boldsymbol{P} \in \Phi_{d\times c}, \boldsymbol{Q} \in \Phi_{n\times c} \end{aligned} \tag{11}$$

The objective in Eq. (11) is a NP-complete problem (Yu and Shi 2003), so it should be relaxed into a easy solved problem. Different from the traditional way relaxing the discrete constrained problem of finding minimal Ncuts into that with continuous constraints, we relax the optimization problem in Eq. (11) into a matrix decomposition problem. By adding two terms $Tr((\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}\boldsymbol{P}^T\boldsymbol{P}(Y^T\boldsymbol{D}\boldsymbol{Y})^{-1}\boldsymbol{Q}^T\boldsymbol{Q})$ and $Tr(X^TX)$ into the objective function in Eq. (11), the optimization problem in Eq. (11) is changed to

$$\begin{aligned} &\min_{\boldsymbol{P},\boldsymbol{Q}} \| \boldsymbol{X} - \boldsymbol{P}(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}\boldsymbol{Q}^T\|_F^2 \\ &s.t. \boldsymbol{P} \in \Phi_{d\times c}, \boldsymbol{Q} \in \Phi_{n\times c} \end{aligned} \tag{12}$$

The optimization problem in Eq. (12) involves the matrix inverse, which would increase the difficulty of finding the solution. We known $(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}$ is a diagonal matrix. To simplify the problem, a diagonal matrix $\boldsymbol{S}$ is used to replace $(\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y})^{-1}$, and $\boldsymbol{S}$ is considered as a parameter to be solved. Therefore, we have the optimization problem of bilateral k-means algorithm as follows.

$$\begin{aligned} &\min_{\boldsymbol{P},\boldsymbol{Q},\boldsymbol{S}} \|\boldsymbol{X} - \boldsymbol{P}\boldsymbol{S}\boldsymbol{Q}^T\|_F^2 \\ &s.t. \boldsymbol{P} \in \Phi_{d\times c}, \boldsymbol{Q} \in \Phi_{n\times c},\ \boldsymbol{S} \in diag \end{aligned} \tag{13}$$

where $diag$ represents the set of diagonal matrices. In Eq. (13), $\boldsymbol{P}$ and $\boldsymbol{Q}$ are the indicator matrices storing the clustering results of features and samples, respectively. $\boldsymbol{S}$ plays a role of connecting $\boldsymbol{P}$ and $\boldsymbol{Q}$. Then, we can transform $\boldsymbol{P}\boldsymbol{S}\boldsymbol{Q}^T$ to a diagonal block matrix by doing some permutations on both columns and rows.

An efficient optimization algorithm

Before giving the solution of BKM, two useful propositions are presented.

Definition 1. Suppose $\boldsymbol{B} \in R^{k\times k}$, its (i,i)-element is b_{ii}, a function $f(\boldsymbol{B})$ is defined as $f(\boldsymbol{B}) = [b_{11}, \cdots, b_{kk}]^T = \boldsymbol{b}$.

Proposition 1. If $\boldsymbol{C} \in R^{k\times k}$ is a diagonal matrix, for a arbitrary matrix $\boldsymbol{B} \in R^{k\times k}$, there exists $Tr(\boldsymbol{B}\boldsymbol{C}) = f(\boldsymbol{B})^T f(\boldsymbol{C})$.

Proof: Since $\boldsymbol{C}$ is a diagonal matrix, $Tr(\boldsymbol{B}\boldsymbol{C}) = \sum_i^k b_{ii}c_{ii} = [b_{11}, \cdots, b_{kk}][c_{11}, \cdots, c_{kk}]^T = f(\boldsymbol{B})^T f(\boldsymbol{C})$ □

Proposition 2. If matrices $\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D} \in R^{k\times k}$ are diagonal matrices, there are $Tr(\boldsymbol{B}\boldsymbol{C}\boldsymbol{D}) = Tr(\boldsymbol{B}\boldsymbol{D}\boldsymbol{C}) = f(\boldsymbol{B})^T\boldsymbol{D}f(\boldsymbol{C})$.

Since the proof is simple, we does not provide them here.

Similar to the standard k-means algorithm, the proposed method is solved in an alternative way.

Firstly, with $\boldsymbol{P}$ and $\boldsymbol{Q}$ fixed, we solve $\boldsymbol{S}$. Let us denote the objective function in Eq. (13) as J_1, and rewrite it into the sum of several matrix traces as follows.

$$\begin{aligned} J_1 &= \|\boldsymbol{X} - \boldsymbol{P}\boldsymbol{S}\boldsymbol{Q}^T\|_F^2 \\ &= Tr(\boldsymbol{X}^T\boldsymbol{X}) - 2Tr(\boldsymbol{Q}^T\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{S}) \\ &+ Tr(\boldsymbol{S}\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{S}\boldsymbol{Q}^T\boldsymbol{Q}) \end{aligned} \tag{14}$$

Since $\boldsymbol{P}$ and $\boldsymbol{Q}$ are indicator matrices, $\boldsymbol{P}^T\boldsymbol{P}$ and $\boldsymbol{Q}^T\boldsymbol{Q}$ are diagonal matrices. According to Proposition 1, $Tr(\boldsymbol{S}\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{S}\boldsymbol{Q}^T\boldsymbol{Q}) = Tr(\boldsymbol{S}\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{S}) = f(\boldsymbol{S})^T(\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{Q})f(\boldsymbol{S})$. According to Proposition 2, $Tr(\boldsymbol{Q}^T\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{S}) = f(\boldsymbol{S})^T f(\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{Q})$.

We use $\boldsymbol{H}$ to denote $\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{Q}$, $\boldsymbol{s}$ to denote $f(\boldsymbol{S})$, and $\boldsymbol{r}$ to denote $f(\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{Q})$. Therefore, J_1 can be rewritten into

$$J_1 = Tr(\boldsymbol{X}^T\boldsymbol{X}) - 2\mathbf{r}^T\mathbf{s} + \mathbf{s}^T\boldsymbol{H}\boldsymbol{s} \tag{15}$$

Then the problem for solving $\boldsymbol{S}$ is transformed to solve $\boldsymbol{s}$. Let us calculate the partial derivative of J_1 with respect to $\boldsymbol{s}$, and make it equal to 0, there is

$$\frac{\partial J_1}{\partial \boldsymbol{s}} = 2(\boldsymbol{H}\boldsymbol{s} - \boldsymbol{r}) = 0 \tag{16}$$

since $\boldsymbol{H}$ is a diagonal matrix, $\boldsymbol{H}^{-1}$ is easily to be solved. So,

$$\boldsymbol{s} = \boldsymbol{H}^{-1}\boldsymbol{r} \tag{17}$$

and $\boldsymbol{S}$ is solved.

Secondly, we calculate $\boldsymbol{Q}$. With $\boldsymbol{P}$ and $\boldsymbol{S}$ fixed, the optimization problem to solve $\boldsymbol{Q}$ can be decomposed into n simple subproblems for each $i(1 \leq i \leq n)$

$$\begin{aligned} &\min_{\boldsymbol{Q}} \|\boldsymbol{x}_{\cdot i} - \boldsymbol{R}\boldsymbol{q}_{i\cdot}^T\|_F^2 \\ &s.t. \boldsymbol{Q} \in \Phi_{n\times c} \end{aligned} \tag{18}$$

where $\boldsymbol{R} = \boldsymbol{PS}$.

Because there is only one element equal to 1 and the rest are zeros in vector $\boldsymbol{q}_{i\cdot}$, the solution of Eq.(18) is determined by

$$q_{ij} = \begin{cases} 1 & j = \text{argmin}_k \|\mathbf{x}_{\cdot i} - \boldsymbol{r}_{\cdot k}\|^2 \\ 0 & otherwise \end{cases} \tag{19}$$

where $\boldsymbol{r}_{\cdot k}$ is the k-th column of $\boldsymbol{R}$.

Finally, we calculate $\boldsymbol{P}$. With the $\boldsymbol{Q}$ and $\boldsymbol{S}$ fixed, the optimization problem for solving $\boldsymbol{P}$ is decomposed into m simple subproblems as in Eq. (20) for each $j(1 \leq j \leq m)$.

$$\begin{aligned} &\min_{\boldsymbol{P}} \|\boldsymbol{x}_{j\cdot} - \boldsymbol{p}_{j\cdot}^T\boldsymbol{L}\|_F^2 \\ &s.t. \boldsymbol{P} \in \Phi_{d\times c} \end{aligned} \tag{20}$$

where $\boldsymbol{L} = \boldsymbol{S}\boldsymbol{Q}^T$.

$$p_{ij} = \begin{cases} 1 & j = \text{argmin}_k \|\mathbf{x}_{j\cdot} - \boldsymbol{l}_{k\cdot}\|^2 \\ 0 & otherwise \end{cases} \tag{21}$$

where $\boldsymbol{l}_k$ is the k-th row of $\boldsymbol{L}$.
The procedures of solving the model of BKM described in Eq. (13) are summarized in Algorithm 1.

Algorithm 1 Bilateral k-means algorithm

Input: Data matrix $\boldsymbol{X} = [\boldsymbol{x}_{\cdot 1}, \cdots, \boldsymbol{x}_{\cdot n}] \in R^{d\times n}$.
Initialize $\boldsymbol{P}$ and $\boldsymbol{Q}$ with arbitrary class indicator matrices.
Repeat

1. Calculate $\boldsymbol{S}$ by Eq. (17);
2. Calculate $\boldsymbol{P}$ by Eq.(19);
3. Calculate $\boldsymbol{Q}$ by Eq.(21);

Until Convergence **Output:** Indicator matrix $\boldsymbol{P}$ for feature clustering and $\boldsymbol{Q}$ for sample clustering.

The computational complexity

In this section, we compare the computational complexity of BKM with that of traditional k-means and BGSP.

As seen from Algorithm 1, the computational complexity of BKM consists of three parts. They are the computational complexities of solving $\boldsymbol{S}$, $\boldsymbol{P}$, $\boldsymbol{Q}$. Since operating multiplication requires more time than operating addition, we only consider the multiplications of proposed methods.

For solving $\boldsymbol{S}$, it involves c times multiplications, and $ndct$ times multiplications in solving $\boldsymbol{Q}$. Here n and d represent the numbers of columns and rows of the data matrix respectively, c represents cluster number, and t represents iteration number. In summary, there are $2dnct + c$ times of multiplications while that of k-means algorithm is $dnct + c$. Thus, the computational complexity of BKM is $O(ndct)$, which is the same with traditional k-means. In many real applications, dct is much less than n, so the complexity of BKM and k-means is often referred as $O(n)$.

BGSP involves a singular value decomposition (SVD), so its complexity is more than $O(n^2d)$. When dealing with large scale data, the number of samples n is much larger than ct. So the complexity of BSGP is much larger than the proposed model BKM. In summary, BKM is more suitable than BGSP for dealing with large scale data.

Experiment results

In this section, several experiments are developed to explore the performance of our proposed method. There are two aspects of experiments. The first is to compare the computational consumption of the proposed BKM with other clustering or co-clustering methods. The second is the comparison of clustering results.

We compare BKM with several state-of-the-art co-clustering methods, such as BGSP (Inderjit 2001),and Orthogonal NMTF(ONMTF) (Ding et al. 2006), Fast NMTF (FNMTF) (Cheng 2015), and multi-linear decomposition with sparse latent factors algorithm (MDSLF) (Papalexakis 2013). Among those methods, FNMTF is a fast co-clustering framework, and MDSLF is designed for sparse data matrix co-clustering. We also compare with the one-way clustering methods, i.e. k-means (MacQueen 1967), NMF (Lee 2001).

Data description and experimental setting

By following previous works, we adopt two types of data sets in our experiments, i.e. real world data sets and synthetic data sets.

The real world data sets consist of WebKB4 (Ding et al. 2006), WebACE (Cai, Wu, and Han 2008), CSTR (Gu and Zhou 2009) and RCV1 (Lewis, Rose, and Li 2004), which are summarized in Table 1. The first three data sets are widely used as benchmarks in clustering literatures. The last data set has very large samples size and feature size. We use it to evaluate the performance of the proposed method for dealing with large data set. For feasibility of calculation on our computer, the keywords (features) appearing less than 100 times in the corpus are removed, which results in 2979 (out of 47236) keywords in our experiments.

Since synthetic data matrix has the exact co-cluster structure, we use it to demonstrate the performance of the proposed method. In our paper, three 200×600 block diagonal synthetic data matrices are generalized, and they are under different noise levels of $\{0.05, 0.10, 0.15\}$ respectively. Each data matrix consists of $c = 5$ clusters. $noise = N/(d \times n)$, where N is the number of non-zero elements in non-diagonal block area. Examples of synthetic data set are shown in Figure 2.

In the experiments, the number of column clusters is set equal to that of row clusters for all the co-clustering methods. Two parameters of MDSLF are set to equal empirically, and they are determined by method in (Papalexakis 2013), As the order presented above, for the four real data sets, the

parameters are $\{80, 92, 43, 94\}$, respectively. For the synthetic data set, they are $\{37, 42, 51\}$. For k-menas and NFM, except the number of clusters, there is no parameter needed to be tuned.

Table 1: Description of real world data sets

Data sets	#Sample	#Feature	#Classes
WebKB4	1140	1644	4
WebACE	4199	1000	20
CSTR	2340	1000	4
RCV1	193844	2979	100

Computational complexity

In this subsection, we explore the computational complexity of the comparison clustering methods. All the experiments run on the computer with Intel (R) Xeon(R) CPU E3-1225 V2, 3.2GHZ CPU and 16.0G memory.

Since most of comparison methods are solved alternately, the iteration number is closely related to the running speed. Thus, before testing the computational times, we examine iteration numbers of the compared methods when they reach the convergence. Every method takes 50 runs. The average results are reported in Table 2. As shown in Table 2, the methods involving discrete solution space (i.e. K-means, FNMF, BMK) have less iteration number than the methods with continuous solution space. This is because the searching scale of discrete solution space is smaller than that of continuous solution space. Among those methods with discrete solution space, the one-way clustering method k-means need more iterations than co-clustering mehtods FNMF and BKM. It is because co-clustering methods utilize the inter-relationship between samples cluster and feature cluster. The reason for BKM converging faster than FNMF is that diagonal-block co-cluster structure is better than checker-board structure in our tasks.

At last, we report the average convergence times of the comparison methods on four real data sets. Every method takes 50 runs, and the average results are presented in Figure 3. As shown in Figure 3, we can see that the BKM costs the least time to converge. Especially, BKM runs faster than k-means method which is well known as linear computational complexity as $O(n)$. That is because BKM requires less iteration numbers than k-means. Thus the results are consistent with the analysis in previous section.

Table 2: Average iteration numbers to converge of different methods.

Data	k-means	NMF	ONMTF	MDSLF	FNMTF	BKM
WebKB4	40.1	43.1	48.2	154	20.1	**16.1**
WebACE	41.3	42.1	46.3	115	26.8	**15.4**
CSRT	38.2	39.2	40.2	132	24.3	**13.2**
RCV1	91.4	98.9	104.6	500	72.1	**43.2**

Clustering results

We evaluate the proposed method on two types of data sets, i.e. real world data sets and synthesis data sets. Three measures widely adopted to evaluate the results of clustering in literatures are used, i.e. Accuracy (Cai, Wu, and Han 2008) , NMI (Cai, Wu, and Han 2008) and Purity (Ding et al. 2006). In the experiments, each method takes 50 runs, and the average results are computed. In Table 3 and Table 4, the results of real world data sets and synthesis data sets are reported respectively. As seen from the Table 3, we can find that the proposed method BKM achieves the better results comparing with other methods. Meanwhile we also could observe that the co-clustering methods, including BKM, MDLSF, BSGP, ONMTF and FMNTF, outperform the traditional cluster methods. It is consistent with the widely accepted hypothesis that utilizing the duality of feature clustering and sample clustering can help clustering of data samples. As seen from Table 4, we can find that BKM achieves better results of column clustering and row clustering than other co-clustering approaches. Meanwhile, the clustering results of BKM on synthesis data sets are shown in Figure 2. It shows the excellent performance of BKM for clustering the columns and rows of data matrix simultaneously.

Conclusions

In this paper, we have proposed a novel co-clustering method named bilateral k-means algorithm. We adopted the main idea of BSGP, and used multipartitioning normalized cuts to construct a new optimization problem for finding the minimal cuts. Different from BSGP, we relaxed the new optimization problem into a special non-negative matrix decomposition and yielded the model of BKM. The BKM is with constraints of indicator matrix, which makes its solution involve much less multiplication than BSGP and other state-of-the-art co-clustering or clustering methods. It has indicated that our method is good at dealing with real world large-scale data set. We also conducted extensive experiments to evaluate the computational complexity and clustering performance of the proposed method. Promising results have shown that BKM not only runs faster than other clustering methods, but also achieves a better performance.

Acknowledgments

This work was supported in part by the National Science Foundation of China under Grant 61473231 and Grant 61522207.

References

Ailem, M.; Role, F.; and Nadif, M. 2015. Co-clustering document-term matrices by direct maximization of graph modularity. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1807–1810. ACM.

Cai, D.; He, X.; Wu, X.; and Han, J. 2008. Non-negative matrix factorization on manifold. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, 63–72. IEEE.

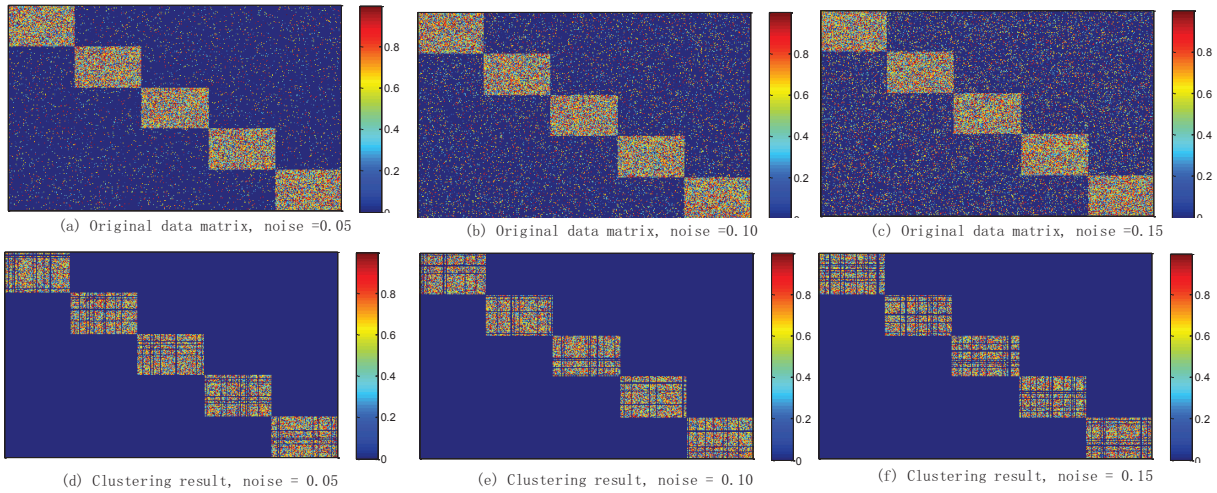


Figure 2: The figures in the first row are original data matrices with different noise rates. The figures in the second row are clustering results, and the blue lines in the diagonal blocks represent error clustered samples or features.

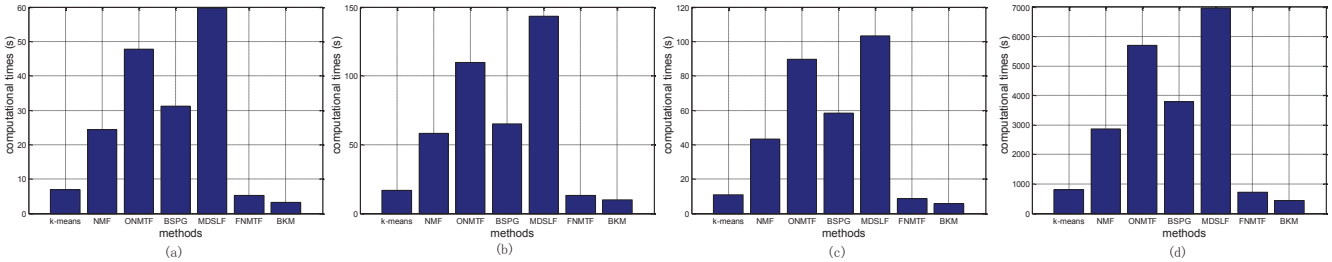


Figure 3: The average running times (s) on different real data sets. (a) WebkB4; (b) WebACE; (c) CSTR; (d) RCV1

Table 3: Clustering results of different methods measured by Accuracy/NMI/Purity on real world data sets.

Data	Metrics	k-means	NMF	ONMTF	BSGP	MDSLF	FNMTF	BKM
WebKB4	Accuracy	0.598	0.568	0.645	0.782	0.763	0.642	**0.786**
	NMI	0.467	0.427	0.442	0.447	0.442	0.431	**0.513**
	Purity	0.601	0.595	0.571	0.632	0.630	0.611	**0.687**
WebACE	Accuracy	0.526	0.514	0.725	0.763	0.753	0.537	**0.782**
	NMI	0.519	0.512	0.528	0.521	0.532	0.536	**0.611**
	Purity	0.479	0.481	0.405	0.512	0521	0.513	**0.624**
CSRT	Accuracy	0.763	0.759	0.796	0.849	0.852	0.781	**0.876**
	NMI	0.624	0.618	0.641	0.714	0.721	0.627	**0.747**
	Purity	0.612	0.587	0.617	0.692	0.703	0.614	**0.711**
RCV1	Accuracy	0.147	0.153	0.163	0.183	0.181	0.169	**0.249**
	NMI	0.262	0.261	0.264	0.321	0.316	0.263	**0.328**
	Purity	0.126	0.123	0.153	0.159	0.153	0.149	**0.162**

Charrad, M., and Ahmed, M. B. 2011. Simultaneous clustering: A survey. In *International Conference on Pattern Recognition and Machine Intelligence*, 370–375. Springer.

Cheng, G.; Han, J.; Guo, L.; Liu, Z.; Bu, S.; and Ren, J. 2015. Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 53(8):1–12.

Cheng, X.; Su, S. G. L. 2015. Co-clusterd: A distributed framework for data co-clustering with sequential updates. *IEEE Transactions on Knowledge and Data Engineering* 12(27):3231–3244.

Ding, C.; Li, T.; Peng, W.; and Park, H. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 126–135.

Table 4: Clustering results of different methods measured by Accuracy/NMI/Purity on synthetic data sets.

Data	Metrics	ONMTF		BSGP		MDSLF		FNMTF		BKM	
		sample	feature	sample	feature	sample	feature	sample	feature	sample	feature
Noise=0.05	Accuracy	0.853	0.853	0.871	0.874	0.869	0.867	0.845	0.843	**0.906**	**0.904**
	NMI	0.854	0.855	0.871	0.872	0.870	0.871	0.845	0.846	**0.909**	**0.907**
	Purity	0.858	0.858	0.872	0.871	0.865	0.864	0.847	0.846	**0.906**	**0.904**
Noise= 0.1	Accuracy	0.853	0.853	0.871	0.874	0.869	0.870	0.848	0.848	**0.906**	**0.904**
	NMI	0.854	0.855	0.871	0.872	0.863	0863	0.849	0.849	**0.909**	**0.907**
	Purity	0.858	0.858	0.872	0.871	0.861	0.857	0.850	0.851	**0.906**	**0.904**
Noise=0.15	Accuracy	0.853	0.853	0.841	0.838	0.830	0.831	0.847	0.844	**0.870**	**0.870**
	NMI	0.854	0.855	0.852	0.843	0.833	0.839	0.846	0.846	**0.891**	**0.889**
	Purity	0.858	0.858	0.831	0.833	0.827	0.825	0.851	0.850	**0.870**	**0.870**

ACM.

Ding, C.; Li, T.; and Jordan, Michael, I. 2010. Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32(1):45–55.

Gu, Q., and Zhou, J. 2009. Co-clustering on manifolds. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 359–368. ACM.

Han, J.; Zhang, D.; Hu, X.; Guo, L.; Ren, J.; and Wu, F. 2015. Background prior-based salient object detection via deep reconstruction residual. *IEEE Transactions on Circuits and Systems for Video Technology* 25(8):1309–1321.

Inderjit, S., D. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 269–274. ACM.

Lee, D.; Seung, H. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, 556–562.

Lewis, David, D. Y. Y.; Rose, Tony, G.; and Li, F. 2004. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research* 5:361–397.

Li, Z.; Wu, X.; and Chang, S. 2012. Segmentation using superpixels: A bipartite graph partitioning approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 789–796. IEEE.

Liu, W.; Li, S., and Lin, X. 2015. Spectralcspatial co-clustering of hyperspectral image data based on bipartite graph. *Multimedia Systems* 1–12.

Lu, X.; Wu, H.; and Yuan, Y. 2014. Double constrained nmf for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing* 52(5):2746–2758.

Lu, X.; Yuan; and Yan, P. 2013. Image super-resolution via double sparsity regularized manifold learning. *IEEE Transactions on Circuits and Systems for Video Technology* 23(12):2022–2033.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

Madeira, Sara, C., and Oliveira, Arlindo, L. 2004. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 1(1):24–45.

Papalexakis, E.; Sidiropoulos, N. D. B. R. 2013. From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors. In *Advances in neural information processing systems*, volume 61, 493–506.

Tanay, A.; Sharan, R.; and Shamir, R. 2005. Biclustering algorithms: A survey. *Handbook of computational molecular biology* 9(1-20):122–124.

Trivedi, S.; Pardos, Zachary, A.; Sarkozy, Gabor, N.; and Heffernan, Neil, T. 2012. Co-clustering by bipartite spectral graph partitioning for out-of-tutor prediction. *International Educational Data Mining Society*.

Yu, S. X., and Shi, J. 2003. Multiclass spectral clustering. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 313–319. IEEE.

Zhang, D.; Lin, C.; Chang, S.; and Smith, John, R. 2004. Semantic video clustering across sources using bipartite spectral clustering. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 1, 117–120. IEEE.