Discovering Multiple Co-Clusterings in Subspaces*

Shixin Yao[†], Guoxian Yu[†], Xing Wang[†], Jun Wang[†] Carlotta Domeniconi[‡] Maozu Guo[§]

Abstract

Multiple clustering approaches aim at exploring alternative ways of organizing a given collection of data into various clusters from different perspectives. Although multiple one-way clusterings have been studied for more than a decade, how to explore alternative two-way clusterings (or *co-clusterings*) still remains an untouched topic, and an important one from an application standpoint. To solve this interesting but yet unexplored topic, we assume the existence of alternative co-clusterings embedded in different subspaces and simultaneously pursue multiple co-clusterings therein. We initially specify a subspace indicator matrix for each feature subspace, and employ matrix tri-factorization to seek row-wise and column-wise cluster indicator matrices in each subspace. To ensure diversity, we quantify the redundancy between pairwise co-clusterings using the cluster indicator and the subspace indicator matrices. We further introduce a unified objective function to simultaneously account for the two pursues, and an alternating optimization solution to iteratively optimize cluster indicator and feature indicator matrices. Our empirical study shows that the proposed solution can explore multiple meaningful co-clusterings and generally achieves better results than state-of-the-art methods.

1 Introduction

Clustering attempts to organize data into disjoint groups called clusters. Clustering is one of the most popular exploratory data analysis techniques in unsupervised learning, and has been applied to a variety of domains, such as biomedicine, collaborative filtering, and financial analysis [1, 2, 3]. Traditional clustering methods typically compute a single partition. However, most data in the real world are rather complex (e.g., biological, multimedia, and social network data), and can be organized in a variety of meaningful clusterings. Typically, it's unknown which clustering is best suited for a given application; thus, generating different clustering results is often the best option. For example, proteins can be divided into a set of clusters based on homology information, 3D structure, or biological functions. Each criterion leads to a different clustering. To explore different clusterings of high quality from a given collection of data, *multiple clustering* has emerged as a new approach in recent years [4]. Some of the existing approaches seek clusterings in alternative to those already explored, by enforcing the new ones to be different [5, 6, 7, 8, 9]; some other solutions simultaneously pursue multiple clusterings [10, 11, 12].

Typically, existing multiple clustering methods focus on one-way clustering, i.e., they cluster samples based on their similarity computed across all the features. However, in real world applications, it's meaningful to consider a two-way clustering (*co-clustering*), in which the data matrix is clustered both sample- and feature-wise [2, 13, 14]. Such an approach enables the exploration of co-clusters relevant to both a subset of samples and a subset of features. As an example, the co-clusters discovered from gene expression data have been used as bio-markers of different cancer subtypes [15].

While sample-wise (or feature-wise) multiple clusterings have been studied extensively, how to find multiple co-clusterings remains a largely *unexplored* topic. Wang *et al.* [16] pioneered a multiple co-clustering solution (MultiCC) by repeatedly applying matrix tri-factorization on the same input data matrix, and by enforcing dissimilarity among the co-clusterings. However, multiple co-clusterings may be embedded in different subspaces (see Fig. 1 for an example), and repeatedly factorizing the original data matrix may result in multiple redundant co-clusterings lacking meaningful interpretations. Tokuda *et al.* [17] introduced a multiple co-clustering solution to account for heterogeneous marginal distributions and features via a nonparametric Bayesian mixture model; but this solution is unable to explore overlapping co-clusterings and requires the explicit modeling of different (a-priori unknown) distributions.

Given these observations, we assume that multiple co-clusterings might be embedded in different subspaces, and propose a solution called multiple co-clusterings in subspaces (MCC-SS). MCC-SS first pre-specifies a fea-

*Supported by NSFC (61873214, 61741217, 61872300, 61871020), Corresponding Author: Jun Wang (kingjun@swu.edu.cn).

[†]College of Computer and Information Sciences, Southwest University, China

[‡]Department of Computer Science, George Mason University, USA

[§]School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture

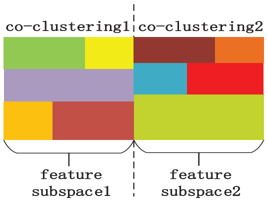


Figure 1: Alternative co-clusterings embedded in different subspaces. Each feature subspace embodies a co-clustering, which includes different co-clusters (colored boxes).

ture projection matrix to map the input data matrix into a new subspace. It then makes use of semi-nonnegative matrix tri-factorization [18] to factorize the projected data matrix into a row-cluster indicator matrix, a column-cluster indicator matrix, and a coefficient matrix between these two matrices, which contributes to a co-clustering of high quality. To ensure diversity among co-clusterings in subspaces, it quantifies redundancy between two co-clusterings using the indicator matrices and projective matrices of the respective co-clusterings, and integrates the redundancy into the matrix tri-factorization objective. Finally, it simultaneously optimizes the row and column-cluster indicator matrices, and the projective matrices of the integrated objective function to pursue multiple subspaces and multiple co-clusterings therein. We emphasize that the study of multiple co-clusterings is different from clustering ensembles [19], co-clustering ensembles [20], and multiple view clustering [21]; the latter three focus on how to derive a consensus (co)clustering result, whereas the former focuses on how to present different explainable co-clustering results of the same data. The main contributions of our work are summarized as follows:

- We study the problem of discovering multiple co-clusterings from a given collection of data, an interesting and relevant problem that has received little attention in the literature. We introduce a matrix factorization based approach (MCC-SS) to explore multiple co-clusterings embedded in subspaces.
- MCC-SS can simultaneously find different subspaces and co-clusterings therein. The discovered co-clusterings are of high quality, have a small degree of redundancy, and can be easily interpreted.
- Extensive experimental results demonstrate that MCC-SS performs significantly better than other competitive approaches [5, 9, 12, 16, 22, 23] in identifying multiple clusterings, and in discovering diverse alternative clusterings.

The remainder of this paper is organized as follows. We briefly review related work in Section 2, and then elaborate on the proposed algorithm and its optimization in Section 3. Section 4 provides the experimental results, and Section 5 discusses conclusions and future work.

2 Related Work

Our work is closely related to two branches of research, co-clustering and multiple clusterings. Co-clustering aims at discovering samples which are similar to one another with respect to a subset of features. Co-clustering can uncover interesting patterns (co-clusters) that cannot be found by traditional one-way clusterings. The exact solution to co-clustering requires enumerating all sub-matrices of the data matrix, which is NP-hard. As such, many approximation solutions have been proposed [2, 15]. Some methods transform the co-clustering problem into the task of partitioning a bipartite graph, whose nodes are samples and features [24, 25]. Other approaches assume the observed samples and features are generated from a finite mixture of underlying probability distributions, and then seek co-clusters via different statistical models [26, 27]. More recent solutions make use of matrix factorization to explore co-clusters and take the factorized low-rank matrices as the row-cluster and column-cluster indicator matrices [28, 29].

Multiple clustering approaches aim at discovering diverse clustering results, each capturing different aspects of the data. Naive solutions consist in: (1) applying a clustering algorithm on the same data using different input parameters or distance metrics; (2) applying different algorithms; or (3) applying a combination of the two [4]. These solutions can generate multiple clusterings, but the latter might be highly redundant, since no constraint is imposed on their similarity. A post-processing operation can be used to filter out clusterings which are too similar [11]. Other methods optimize the dissimilarity between the to-be-explored clustering and the already explored ones [5, 9]. Others simultaneously seek multiple clusterings by decreasing the correlation between pairwise distinct clusterings [10, 12]. A series of methods seek multiple clusterings by operating features [11, 23, 30]. To name a few, Caruana *et al.* [11] and Hu *et al.* [23] assign different weights to features and seek alternative clusterings in the resulting feature spaces. Davidson and Qi [22] use multi-link and cannot-link constraints between samples to learn a projected feature space, and then seek the alternative clustering in this space. Cui *et al.* [31] sequentially generate an alternative clustering from a new feature space, which is orthogonal to the previously used feature spaces.

All these aforementioned multiple clusterings focus on finding alternative one-way clusterings, and some can only find two alternative clusterings [12, 22]. However, in real world applications (i.e., cancer genomic data analysis [15] and e-commerce recommendation [28]), it's desirable to provide multiple, different, and explainable co-clusterings of the same data matrix. Wang *et al.* [16] pioneered a multiple co-clustering approach (MultiCC), which assumes co-clusterings are embedded in the same feature space, and consistently applies matrix tri-factorization on the same data matrix. As such, the resulting co-clusterings still suffer from high redundancy. Furthermore, this approach cannot uncover co-clusterings embedded in different subspaces. Inspired by feature transformation based multiple clusterings [22, 23, 31] and MultiCC, we introduce an approach called MCC-SS to simultaneously explore multiple co-clusterings in different feature subspaces. Compared to MultiCC, MCC-SS can further reduce redundancy by controlling the diversity in feature subspaces and can obtain more meaningful co-clusterings.

3 Methodology

3.1 Multiple Co-clusterings

Let $\mathbf{X} \in \mathbb{R}^{d \times n}$ denote a data matrix with d rows (i.e., features or variables) and n columns (i.e., samples). Co-clusterings are represented by the respective sample-wise and feature-wise co-clustering indicator matrices ($\mathbf{R}_h \in \mathbb{R}^{d \times k_h}$ and $\mathbf{C}_h \in \mathbb{R}^{n \times l_h}$, $h = 1, 2, \cdots, m$, where m is the target number of alternative co-clusterings). If feature i belongs to the $k_h^{'}$-th row-cluster of the h-th co-clustering, $\mathbf{R}_h(i, k_h^{'}) = 1$; otherwise, $\mathbf{R}_h(i, k_h^{'}) = 0$. Similarly, $\mathbf{C}_h$ is the column-cluster indicator matrix, stating that the h-th co-clustering groups the n samples into l_h column-clusters. If sample j belongs to the $l_h^{'}$-th cluster, $\mathbf{C}_h(j, l_h^{'}) = 1$; otherwise, $\mathbf{C}_h(j, l_h^{'}) = 0$. Multiple co-clustering approaches aim at finding multiple co-clusterings of good *quality* and diverse from one another.

Inspired by the success of semi-nonnegative matrix tri-factorization in co-clustering [16, 18, 28, 29], we adopt the tri-factorization formulation to find multiple co-clusterings as follows:

$$\begin{aligned} \Psi_1(\{\mathbf{R}_h\}_{h=1}^m, \{\mathbf{C}_h\}_{h=1}^m) = \frac{1}{m} \sum_{h=1}^{m} \| \mathbf{X} - \mathbf{R}_h \mathbf{S}_h \mathbf{C}_h^T \|_F^2 \\ s.t.\ \mathbf{R}_h \geq 0; \mathbf{C}_h \geq 0 \end{aligned} \tag{3.1}$$

where $\mathbf{S}_h \in \mathbb{R}^{k_h \times l_h}$ is the coefficient matrix, which enables different numbers of row-clusters and column-clusters, and the minimization of the squared error induced by matrix factorization. Traditional nonnegative matrix factorization requires the input data matrix to be nonnegative [32]. Here, $\mathbf{S}_h$ can be mix-sign, and thus Eq. (3.1) can accommodate a data matrix with negative feature values.

Eq. (3.1) implicitly assumes that multiple co-clusterings are embedded in the same feature space. However, they may not. In fact, alternative clusterings are often embedded in different feature spaces, and this observation is widely adopted in multiple clusterings [23, 31]. Given this, we assume multiple co-clusterings may also be embedded in different latent subspaces, via a series of projection coefficient matrices $\{\mathbf{P}_h \in \mathbb{R}^{d \times d_h}\}_{h=1}^m$. We then apply matrix tri-factorization on the projected data matrix $\mathbf{P}_h^T \mathbf{X}$ as follows:

$$\begin{aligned} &\Psi_2(\{\mathbf{R}_h\}_{h=1}^m, \{\mathbf{C}_h\}_{h=1}^m, \{\mathbf{P}_h\}_{h=1}^m)) \\ &= \frac{1}{m} \sum\nolimits_{h=1}^{m} \|\mathbf{P}_h^T \mathbf{X} - \mathbf{R}_h \mathbf{S}_h \mathbf{C}_h^T\|_F^2 \\ &\qquad s.t.\ \mathbf{R}_h \geq 0; \mathbf{C}_h \geq 0 \end{aligned} \tag{3.2}$$

We observe that $\mathbf{R}_h \in \mathbb{R}^{d_h \times k_h}$, indicating the h-th co-clustering, is explored in the feature space spanned by $\mathbf{R}_h$. By specifying different $\{\mathbf{P}_h \in \mathbb{R}^{d \times d_h}\}_{h=1}^m$, multiple diverse co-clusterings can be pursued, but the diversity among the co-clusterings cannot be sufficiently guaranteed.

To enforce the diversity (or non-redundancy) between multiple co-clusterings, we can quantify the similarity between two co-clusterings using the co-association information of samples and features. Specifically, let $\mathbf{A}_h^c \in \mathbb{R}^{n \times n}$ store the co-association between projected samples in the h-th feature space; $\mathbf{A}_h^c(i, j) = 1$ if two samples i and j are in the same column-cluster of the h-th co-clustering; and $\mathbf{A}_h^c(i, j) = 0$ otherwise. Then the overall sample-wise redundancy for each pairwise column-clusterings can be approximately quantified as follows:

$$\begin{aligned} \Psi_3(\{\mathbf{C}_h\}_{h=1}^m)) &= \sum_{\substack{h_1, h_2 = 1 \\ h_1 \neq h_2}}^{m} \sum_{i,j=1}^{n} (\mathbf{A}_{h_1}^c)_{ij} (\mathbf{A}_{h_2}^c)_{ij} \\ &= \sum_{\substack{h_1, h_2 = 1 \\ h_1 \neq h_2}}^{m} tr(\mathbf{C}_{h_1}^T \mathbf{C}_{h_1} \mathbf{A}_{h_2}^c) = \sum_{\substack{h_1, h_2 = 1 \\ h_1 \neq h_2}}^{m} \|\mathbf{C}_{h_1}^T \mathbf{C}_{h_2}\|_F^2 \end{aligned} \tag{3.3}$$

Eq. (3.3) quantifies the diversity for all pairs of column-clusterings: the smaller the value is, the smaller is the portion of two samples placed in the same clusters in two co-clusterings, and therefore the smaller the redundancy among column-clusterings is. The above equation still holds when $\mathbf{C}_h$ is a numerical matrix.

The original features are projected into different feature subspaces, and the subspaces may have different number of features. In addition, the projected new features have a different meaning. As such, we cannot adopt the idea of Eq. (3.3) to quantify the feature-wise redundancy between pairwise co-clusterings. To bypass this issue, we maximize the difference between $\{\mathbf{P}_h \in \mathbb{R}^{d \times d_h}\}_{h=1}^m$ to reduce the redundancy among row-clusterings as follows:

$$\begin{aligned}\Psi_4(\{\mathbf{P}_h\}_{h=1}^m)) &= \sum_{\substack{h_1,h_2=1\\h_1\neq h_2}}^{m} \sum_{i,j=1}^{d} (\mathbf{P}_{h_1}\mathbf{P}_{h_1}^T)_{ij}(\mathbf{P}_{h_2}\mathbf{P}_{h_2}^T)_{ij} \\ &= \sum_{\substack{h_1,h_2=1\\h_1\neq h_2}}^{m} \|\mathbf{P}_{h_1}^T\mathbf{P}_{h_2}\|_F^2 \end{aligned} \tag{3.4}$$

The more the orthogonality condition between $\mathbf{P}_{h_1}$ and $\mathbf{P}_{h_2}$ is satisfied, the more dissimilar the two feature subspaces are, and therefore the more dissimilar the pairwise row-clusterings are.

To this end, we integrate Eqs. (3.2) and (3.3) with Eq. (3.4) to simultaneously seek m different co-clusterings via the following unified objective function:

$$\begin{aligned} &J(\{\mathbf{R}_h\}_{h=1}^m, \{\mathbf{C}_h\}_{h=1}^m, \{\mathbf{P}_h\}_{h=1}^m)) \\ =&\frac{1}{m}\sum\nolimits_{h=1}^{m} \|\mathbf{P}_h^T\mathbf{X} - \mathbf{R}_h\mathbf{S}_h\mathbf{C}_h^T\|_F^2 \\ &+\frac{\lambda_1}{C_m^2}\sum_{\substack{h_1,h_2=1\\h_1\neq h_2}}^{m} \|\mathbf{C}_{h_1}^T\mathbf{C}_{h_2}\|_F^2 + \frac{\lambda_2}{C_m^2}\sum_{\substack{h_1,h_2=1\\h_1\neq h_2}}^{m} \|\mathbf{P}_{h_1}^T\mathbf{P}_{h_2}\|_F^2 \\ &s.t.\ \mathbf{R}_h \geq 0, \mathbf{C}_h \geq 0 \end{aligned} \tag{3.5}$$

where the two regularization parameters $\lambda_1 > 0$ and $\lambda_2 > 0$ balance the quality of the m co-clusterings, which is pursued by the matrix tri-factorization, and the diversity among the co-clusterings, which is pursued by the last two terms. Two normalization factors, $1/C_m^2$ and $1/m$, are introduced to reduce the impact of scaling.

Most feature-transformation based alternative clusterings seek the alternative clusterings in a sequential manner, and refer to the already explored clusterings [9, 23, 31]; some even require to specify the feature subspaces in advance [11, 22]. In contrast, MCC-SS can simultaneously pursue multiple subspaces and multiple alternative co-clusterings therein, and it is less affected by the quality of the already explored co-clusterings. In addition, the target number of alternative co-clusterings of MCC-SS can be specified by the user. Another underlying merit of MCC-SS is that it can jointly perform feature selection and explore different co-clusterings in the selected (possibly overlapping) feature subspaces. As a result, we can obtain clear interpretations of the co-clusterings with respect to different compositions of features.

3.2 Optimization Algorithm

$\mathbf{R}_h$ and $\mathbf{C}_h$ are both binary matrices, so it is very hard to directly optimize them. As such, we relax the entries of $\mathbf{R}_h$ and $\mathbf{C}_h$ to nonnegative numeric values. Note, the above Eqs. (3.3-3.5) still hold when $\mathbf{R}_h$ and $\mathbf{C}_h$ are numerical. Since Eq. (3.5) is not jointly convex for $\mathbf{R}_h$, $\mathbf{C}_h$, and $\mathbf{P}_h$, it is unrealistic to find the global optimal values for all the variables. Here, we solve Eq. (3.5) via the Alternating Direction Method of Multipliers (ADMM) [33], which alternatively optimizes one variable, whilst fixing the other variables.

Optimizing $\mathbf{S}_h$ with $\mathbf{R}_h$, $\mathbf{C}_h$, and $\mathbf{P}_h$ fixed: The optimization of Eq. (3.5) with respect to $\mathbf{S}_h$ is equivalent to the following objective:

$$J_S(\mathbf{S}_h) = \|\mathbf{P}_h^T\mathbf{X} - \mathbf{R}_h\mathbf{S}_h\mathbf{C}_h^T\|_F^2 \tag{3.6}$$

Letting the partial derivative $\frac{\partial \mathbf{J}_S}{\partial \mathbf{S}_h} = 0$, we can obtain the updating formula of $\mathbf{S}_h$ as follows:

$$\mathbf{S}_h = [\mathbf{R}_h^T\mathbf{R}_h]^{-1}\mathbf{R}_h^T\mathbf{P}_h^T\mathbf{X}\mathbf{C}_h[\mathbf{C}_h^T\mathbf{C}_h]^{-1} \tag{3.7}$$

Optimizing $\mathbf{R}_h$ with $\mathbf{S}_h$, $\mathbf{C}_h$, and $\mathbf{P}_h$ fixed: The optimization of Eq. (3.5) with respect to $\mathbf{R}_h$ is equivalent to the following objective:

$$J_R(\mathbf{R}^h) = \|\mathbf{P}_h^T\mathbf{X} - \mathbf{R}_h\mathbf{S}_h(\mathbf{C}_h)^T\|_F^2 \tag{3.8}$$

Because of the nonnegative constraints $\mathbf{R}_h \geq 0$, we introduce the Lagrangian multiplier $\boldsymbol{\alpha} \in \mathbb{R}^{d_h \times k_h}$ and update the above equation as follows:

$$\tilde{J}_R(\mathbf{R}_h) = \|\mathbf{P}_h^T\mathbf{X} - \mathbf{R}_h\mathbf{S}_h\mathbf{C}_h^T\|_F^2 - tr(\boldsymbol{\alpha}\mathbf{R}_h^T) \tag{3.9}$$

Letting the partial derivative $\frac{\partial \tilde{J}_R}{\partial \mathbf{R}_h} = 0$, we can get $\boldsymbol{\alpha}$ as:

$$\boldsymbol{\alpha} = -2\mathbf{A} + 2\mathbf{R}_h\mathbf{B} \tag{3.10}$$

where $\mathbf{A} = \mathbf{P}_h^T\mathbf{X}\mathbf{C}_h\mathbf{S}_h^T$, $\mathbf{B} = \mathbf{S}_h\mathbf{C}_h^T\mathbf{C}_h\mathbf{S}_h^T$. Based on the Karush-Kuhn-Tucker (KKT) complementarity condition [34] $(\boldsymbol{\alpha})_{ij}(\mathbf{R}_h)_{ij} = [-\mathbf{A} + \mathbf{R}_h\mathbf{B}]_{ij}(\mathbf{R}_h)_{ij} = 0$ we have:

$$[-\mathbf{A}^+ + \mathbf{A}^- + \mathbf{R}^h\mathbf{B}^+ - \mathbf{R}^h\mathbf{B}^-]_{ij}(\mathbf{R}_h)_{ij} = 0 \tag{3.11}$$

where $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$, $\mathbf{B} = \mathbf{B}^+ - \mathbf{B}^-$, and $\mathbf{A}_{ij}^+ = (|\mathbf{A}_{ij}| + \mathbf{A}_{ij})/2$, $\mathbf{A}_{ij}^- = (|\mathbf{A}_{ij}| - \mathbf{A}_{ij})/2$, The above equation leads to the following updating formula for $\mathbf{R}_h$:

$$(\mathbf{R}_h)_{ij} \leftarrow (\mathbf{R}_h)_{ij}\sqrt{\frac{[\mathbf{A}^+ + \mathbf{R}_h\mathbf{B}^-]_{ij}}{[\mathbf{A}^- + \mathbf{R}_h\mathbf{B}^+]_{ij}}} \tag{3.12}$$

Optimizing $\mathbf{C}_h$ with $\mathbf{S}_h$, $\mathbf{R}_h$, and $\mathbf{P}_h$ fixed: Similarly to the optimization of $\mathbf{R}_h$, the optimization of Eq. (3.5) with respect to $\mathbf{C}_h$ is equivalent to the following objective:

$$J_C(\mathbf{C}_h) = \|\mathbf{P}_h^T\mathbf{X} - \mathbf{R}_h\mathbf{S}_h\mathbf{C}_h^T\|_F^2 + \frac{\lambda_1}{C_m^2}\sum_{\substack{h_2=1\\h_2\neq h}}^{m} \|\mathbf{C}_h^T\mathbf{C}_{h_2}\|_F^2 \tag{3.13}$$

To satisfy the nonnegative constraints $\mathbf{C}_h \geq 0$, we introduce a Lagrangian multiplier $\boldsymbol{\beta} \in \mathbb{R}^{n \times l_h}$ for $\mathbf{C}_h$ and update the above equation as follows:

$$\begin{aligned} &\tilde{J}_C(\mathbf{C}_h) = \|\mathbf{P}_h^T\mathbf{X} - \mathbf{R}_h\mathbf{S}_h\mathbf{C}_h^T\|_F^2 \\ &+\frac{\lambda_1}{C_m^2}\sum_{\substack{h_2=1\\h_2\neq h}}^{m} \|\mathbf{C}_h^T\mathbf{C}_{h_2}\|_F^2 - tr(\boldsymbol{\beta}\mathbf{C}_h^T) \end{aligned} \tag{3.14}$$

Again, we take the partial derivative of $\tilde{J}_C(\mathbf{C}_h)$ with respect to $\mathbf{C}_h$ and let $\frac{\partial \tilde{J}_C}{\partial \mathbf{C}_h} = 0$. We obtain:

$$\boldsymbol{\beta} = -2\mathbf{P} + 2\mathbf{C}_h\mathbf{Q} + 2\lambda_1\boldsymbol{\Gamma}_h \tag{3.15}$$

where $\mathbf{P} = \mathbf{X}^T\mathbf{P}_h\mathbf{R}_h\mathbf{S}_h$, $\mathbf{Q} = \mathbf{S}_h^T\mathbf{R}_h^T\mathbf{R}^h\mathbf{S}^h$, $\boldsymbol{\Gamma}_h = (\sum_{h_2=1,h_2\neq h}^{m}(\mathbf{C}_{h_2}\mathbf{C}_{h_2}^T\mathbf{C}_h))/C_m^2$. Based on the KKT complementarity condition $\boldsymbol{\beta}_{ij}(\mathbf{C}_h)_{ij} = 0$, we obtain the updating rule for $\mathbf{C}_h$ as follows:

$$(\mathbf{C}_h)_{ij} \leftarrow (\mathbf{C}_h)_{ij}\sqrt{\frac{[\mathbf{P}^+ + \mathbf{C}_h\mathbf{Q}^- + \lambda_1\boldsymbol{\Gamma}_h^-]_{ij}}{[\mathbf{P}^- + \mathbf{C}_h\mathbf{Q}^+ + \lambda_1\boldsymbol{\Gamma}_h^+]_{ij}}} \tag{3.16}$$

where $\mathbf{P} = \mathbf{P}^+ - \mathbf{P}^-$, $\mathbf{Q} = \mathbf{Q}^+ - \mathbf{Q}^-$ and $\boldsymbol{\Gamma}_h = \boldsymbol{\Gamma}_h^+ - \boldsymbol{\Gamma}_h^-$ follow a similar definition as $\mathbf{A}^+$ and $\mathbf{A}^-$.

Optimizing $\mathbf{P}_h$ with $\mathbf{S}_h$, $\mathbf{C}_h$, and $\mathbf{R}_h$ fixed: The optimization of Eq. (3.5) with respect to $\mathbf{P}_h$ is equivalent to the following objective:
(3.17)

$$J_P(\mathbf{P}_h) = \|\mathbf{P}_h^T\mathbf{X} - \mathbf{R}_h\mathbf{S}_h\mathbf{C}_h^T\|_F^2 + \frac{\lambda_2}{C_m^2}\sum_{\substack{h_2=1\\h_2\neq h}}^{m}\|\mathbf{P}_h^T\mathbf{P}_{h_2}\|_F^2$$

Letting the partial derivative $\frac{\partial J_P}{\partial \mathbf{P}_h} = 0$, leads to the following updating formula for $\mathbf{P}_h$:

$$\mathbf{P}_h = (\mathbf{X}\mathbf{X}^T + \frac{\lambda_2}{C_m^2}\sum_{\substack{h_2=1\\h_2\neq h}}^{m}\mathbf{P}_{h_2}\mathbf{P}_{h_2}^T)^{-1}\mathbf{X}\mathbf{C}_h\mathbf{S}_h^T\mathbf{R}_h^T \tag{3.18}$$

By iteratively applying the above updating rules for $\mathbf{S}_h$, $\mathbf{R}_h$, $\mathbf{C}_h$, and $\mathbf{P}_h$, we can approximately solve the objective function of MCC-SS. Our empirical study shows that MCC-SS often converges after 50 iterations. The convergence trend and runtime costs of MCC-SS will be provided in the experimental section.

With respect to the time complexity of MCC-SS, it takes $O(knd_h + lnd_h)$ to update $\mathbf{S}_h$, $O(mkd_h^2)$ to update $\mathbf{R}_h$, $O(mln^2)$ to update $\mathbf{C}_h$, and $O(mn^2d_h)$ to update $\mathbf{P}_h$. Suppose t is the number of iterations for convergence, the overall complexity of MCC-SS is $O(mt(knd_h + nld_h + mkd_h^2 + mln^2 + mn^2d_h))$. MCC-SS holds similar time complexity as MultiCC.

Table 1: Characteristics of the datasets

Datasets	#Samples	#Features	#Classes
Vowel	528	10	10
Ionosphere	351	34	2
Glass	214	9	7
Vehicle	846	17	4
Crowdsourced	10545	28	6
CMUface	640	15360	20/4
Dancing	900	400	9

4 Experimental Results and Analysis

4.1 Experimental Setup With multiple clusterings we need to measure the quality and diversity of alternative clusterings. To measure quality, we adopt the widely used Silhouette Coefficient (SC) and Dunn Index (DI) as internal indexes [1]. *Larger* values of SC and DI indicate a *higher* quality clustering. To measure the redundancy between alternative clusterings, we adopt the Normalized Mutual Information (NMI) and the Jaccard Coefficient (JC) as external indexes. The *smaller* the values of NMI and JC are, the *smaller* the redundancy between alternative clusterings is. All these metrics have been used in the multiple clustering literature [4]. Due to space limitations, the formal definitions of these metrics are omitted, and can be found in [4, 9].

Given the characteristics of co-clustering and the scarcity of available multiple co-clusterings, we evaluate the performance of our MCC-SS from two angles: (1) Discovering multiple sample-wise clusterings, and compare MCC-SS with MultiCC [16] and other representative multiple clustering methods; and (2) Finding multiple co-clusterings and visualize them. Six datasets collected from the UCI machine learning repository and a Dancing image dataset are used for the experiments. These datasets are often used for multiple clusterings [10, 31, 35]; their details are summarized in Table 1.

MCC-SS needs to preset the following required parameters: the target number of alternative co-clusterings (m), the number of row-clusters (k), the number of column-clusters (l), and the dimensionality of subspaces d_h. We adopt a widely used technique to determine the number of row-clusters [36]: we first repeat k-means under each input value of k, and then measure the stability of the clusterings obtained under each k, and finally set k to the value that gives the most stable clustering results as the target number of row-clusters. Based on this technique and with $m = 2$, we finally determine the value of k as follows: $k = 5$ for Vowel, $k = 7$ for Ionosphere, $k = 3$ for Glass, $k = 3$ for Vehicle, $k = 6$ for CMUface, $k = 5$ for Dancing, and $k = 4$ for Crowdsourced. The value of l is set equal to the number of true classes of the respective datasets, as specified in Table 1. l_1 and l_2 for the CMUface are set to 20 (number of identities) and 4 (type of poses), respectively. For the first five datasets with a moderate number of features, we set $d_h = d$ (number of original features), and for the last two high-dimensional datasets, we set $d_h = 0.8d$ (Dancing) and $d_h = 0.5d$ (CMUface).

4.2 Discovering Multiple One-way Clusterings In this section, we conduct experiments to investigate the capability of MCC-SS in finding multiple one-way clusterings and compare them with the multiple clusterings explored by MultiCC [16], Dec-k-means

Table 2: Quality and Diversity of the various competing methods. ↑(↓) indicates the direction of preferred values for the corresponding measure. ●/○ indicates whether MCC-SS is statistically (according to pairwise t-test at 95% significance level) superior/inferior to the other method.

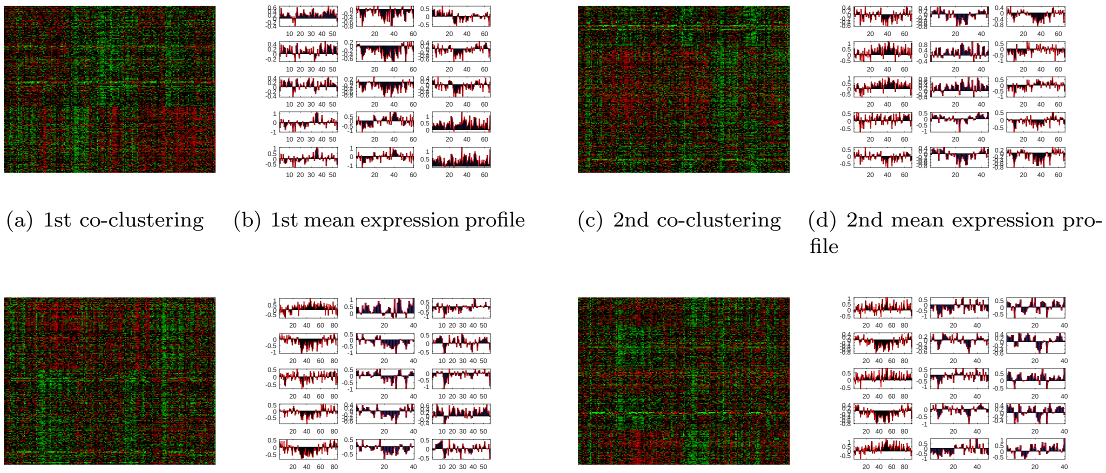
		Dec-kmeans	COALA	ADFT	MSC	OSC	MNMF	MultiCC	MCC-SS
Vowel	SC↑	0.088±0.042●	0.133±0.001●	0.233±0.015○	0.238±0.015○	0.046±0.005●	0.018±0.013●	0.243±0.021○	0.181±0.000
	DI↑	0.021±0.000●	0.079±0.000●	0.071±0.011●	0.033±0.003●	0.024±0.000●	0.019±0.003●	0.073±0.003●	0.118±0.002
	NMI↓	0.026±0.011○	0.158±0.005●	0.678±0.083●	0.458±0.023●	0.434±0.003●	0.018±0.003○	0.418±0.013●	0.037±0.000
	JC↓	0.125±0.008●	0.225±0.006●	0.353±0.054●	0.412±0.023●	0.196±0.003●	0.112±0.003●	0.142±0.013●	0.056±0.001
Ionosphere	SC↑	0.258±0.018●	0.392±0.000●	0.412±0.006●	0.401±0.002●	0.350±0.009●	0.108±0.012●	0.405±0.005●	0.741±0.011
	DI↑	0.095±0.014○	0.041±0.000○	0.077±0.006○	0.041±0.015	0.241±0.013○	0.013±0.009	0.062±0.012○	0.022±0.000
	NMI↓	0.534±0.020●	0.362±0.000●	0.809±0.016●	0.588±0.041●	0.024±0.002○	0.313±0.006●	0.115±0.005●	0.055±0.000
	JC↓	0.588±0.033●	0.502±0.000●	0.788±0.013●	0.744±0.032●	0.618±0.002●	0.338±0.005	0.448±0.012●	0.345±0.002
Glass	SC↑	0.512±0.076○	0.664±0.006○	0.566±0.010○	0.665±0.016○	0.148±0.038●	-0.131±0.076●	0.161±0.011●	0.224±0.002
	DI↑	0.047±0.019●	0.213±0.011○	0.021±0.008●	0.119±0.025○	0.017±0.000●	0.016±0.002●	0.253±0.018○	0.113±0.001
	NMI↓	0.056±0.023●	0.176±0.002●	0.866±0.018●	0.286±0.044●	0.460±0.007●	0.077±0.018●	0.037±0.008	0.032±0.000
	JC↓	0.428±0.022●	0.459±0.006●	0.874±0.018●	0.733±0.026●	0.310±0.007●	0.175±0.014●	0.078±0.011	0.077±0.000
Vehicle	SC↑	0.118±0.086●	0.661±0.001○	0.741±0.005○	0.756±0.021○	0.527±0.011○	-0.159±0.022●	0.111±0.022●	0.369±0.004
	DI↑	0.008±0.000●	0.065±0.000○	0.045±0.000○	0.035±0.002○	0.031±0.000○	0.013±0.002	0.049±0.000○	0.020±0.002
	NMI↓	0.175±0.012●	0.698±0.005●	0.998±0.031●	0.889±0.034●	0.628±0.028●	0.129±0.004●	0.146±0.002●	0.003±0.000
	JC↓	0.286±0.022●	0.722±0.004●	0.989±0.012●	0.937±0.054●	0.359±0.028●	0.242±0.004●	0.142±0.004	0.141±0.002
Crowdsourced	SC↑	0.038±0.003●	0.211±0.011●	0.196±0.008●	0.012±0.000●	0.192±0.013●	-0.011±0.000●	0.342±0.013○	0.232±0.002
	DI↑	0.027±0.001○	0.115±0.003○	0.073±0.002○	0.020±0.001○	0.042±0.001○	0.002±0.000●	0.013±0.001	0.016±0.001
	NMI↓	0.017±0.002●	0.349±0.015●	0.863±0.035●	0.111±0.006●	0.806±0.015●	0.022±0.000●	0.053±0.002●	0.013±0.000
	JC↓	0.144±0.010●	0.348±0.014●	0.538±0.032●	0.150±0.003●	0.823±0.047●	0.109±0.002●	0.155±0.001●	0.081±0.000
CMUface	SC↑	0.016±0.014●	0.044±0.002●	0.054±0.002●	0.011±0.012●	0.230±0.000○	-0.008±0.012●	0.075±0.013	0.093±0.000
	DI↑	0.088±0.033●	0.124±0.001●	0.014±0.001●	0.022±0.002●	0.203±0.000●	0.044±0.022●	0.144±0.011●	0.272±0.014
	NMI↓	0.042±0.011○	0.082±0.000○	0.662±0.019●	0.522±0.023●	0.794±0.003●	0.038±0.021○	0.028±0.003○	0.111±0.000
	JC↓	0.154±0.004●	0.166±0.000●	0.549±0.023●	0.511±0.026●	0.462±0.003●	0.142±0.006●	0.042±0.009●	0.026±0.000
Dancing	SC↑	-0.264±0.014●	-0.047±0.002●	0.680±0.006○	0.669±0.000○	0.607±0.009○	-0.198±0.004●	0.549±0.001○	0.165±0.001
	DI↑	0.043±0.000●	0.296±0.001○	0.113±0.002○	0.824±0.022○	0.069±0.000●	0.033±0.000●	0.030±0.000●	0.098±0.000
	NMI↓	0.147±0.007●	0.444±0.001●	0.860±0.002●	0.961±0.000●	0.833±0.012●	0.252±0.000●	0.218±0.004●	0.015±0.000
	JC↓	0.101±0.007●	0.072±0.001●	0.433±0.002●	0.989±0.000●	0.368±0.002●	0.122±0.002●	0.115±0.000●	0.059±0.000

[12], COALA [5], ADFT [22], MSC [23], OSC [31], and MNMF [9]. For COALA and ADFT, we use k-means to generate the first clustering ($\mathcal{C}$), and then apply their respective solutions to generate the second alternative clustering ($\mathcal{C}^*$). We downloaded the source code of MNMF and MSC, and directly implemented the other methods based on the respective original papers. Parameters were specified or optimized as suggested by the authors. Following the experimental procedure adopted by these methods, we measure clustering quality on $\mathcal{C}^*$, and measure the diversity using $\mathcal{C}$ and $\mathcal{C}^*$. Table 2 reports the average and standard deviation results of ten independent runs.

MCC-SS generally outperforms other comparing methods on different datasets and across the used metrics. Both MCC-SS and MultiCC use matrix factorization to explore multiple co-clusterings. The two alternative clusterings explored by MCC-SS have a larger diversity than those found by MultiCC, while they almost hold the same quality. This comparison supports our motivation to pursue diverse alternative clusterings in subspaces. MCC-SS, ADFT, and OSC explore multiple clusterings by mainly operating features. MCC-SS almost always outperforms the latter two in terms of diversity, but sometimes loses to them in terms of quality. This is because MCC-SS simultaneously pursues multiple clusterings, whereas AFDT and OSC pursue multiple clusterings sequentially. As such, AFDT and OSC give high priority to the quality of the first clustering, and may obtain two alternative high-quality clusterings. In terms of clustering quality, MSC performs better than MCC-SS on Glass, Vehicle, and Dancing. The reason is that MSC gives more emphasis on stability than diversity. MCC-SS has a DI which is lower than other methods on Ionosphere and Vehicle; this may be because $tr(P_h^T P_{h'})$ is not an ideal surrogate for the redundancy between clusterings, and thus may also reduce the quality of clustering results in some cases.

4.3 Discovering Multiple Co-clusterings To show that MCC-SS can explore more than two diverse co-clusterings, we apply MCC-SS on the Diffuse Large B Cell Lymphoma (DLBCL) gene expression data [37]. We preprocess this gene expression data by removing the genes that are not expressed or have a small variance, and finally obtain a data matrix with 360 genes and 180 samples (cancer patients). Particularly, we set the number of multiple co-clusterings $m = 4$, the number of gene clusters $k = 5$, the number of sample-clusters $l = 3$, $\lambda_1 = \lambda_2 = 100$, and $d_h = 360$. We visualize the results by plotting the heatmap, and the mean gene expression profile of each co-clustering in Figure 2. In the heatmap, red points indicate that the gene expression is a high expression value, while green points indicate low expression values.

From Figure 2, we can see that MCC-SS groups genes and samples into multiple red and green blocks, implying that it can find co-expression patterns of genes across specific samples. In other words, MCC-SS can find multiple diverse co-clusterings of good quality. As shown by the mean expression profiles of these co-clusterings, the co-clusters contain a different number of samples. For example, these four co-clusterings separately group 180 samples into three clusters of sizes {91, 49, 40}, {45, 67, 68}, {84, 40, 56}, and {60, 66, 54}, respectively. In addition, these co-clusters also contain a different number of features, and they divide the projected feature subspace into five column-clusters of sizes {100, 83, 68,



(a) 1st co-clustering (b) 1st mean expression profile (c) 2nd co-clustering (d) 2nd mean expression profile

(e) 3rd co-clustering (f) 3rd mean expression profile (g) 4th co-clustering (h) 4th mean expression profile

Figure 2: Heatmaps and mean gene expression profiles of co-clusters for four different co-clusterings.

52, 57}, {61, 105, 84, 53, 57}, {101, 93, 45, 39, 82}, and {75, 102, 37, 68, 78}. For the co-clusterings explored by MultiCC (results are not shown due to space limitations), the first and fourth co-clusterings explored by MultiCC are similar, and they do not manifest much diversity feature-wise.

We further use the average Co-cluster relevance Score(CS) [38] to measure the diversity between co-clusterings, and report the scores for MCC-SS and MultiCC in Table 3. A larger value of CS indicates a higher redundancy. The row-wise CS between two alternative co-clusterings is computed as follows:

(4.19)
$$CS_{h_1h_2} = \frac{1}{|\mathcal{C}^{h_1}|} \sum_{(\mathcal{R}_i,\mathcal{C}_i)\in\mathcal{C}^{h_1}} \max_{(\mathcal{R}_j,\mathcal{C}_j)\in\mathcal{C}^{h_2}} \frac{|\mathcal{R}_i \cap \mathcal{R}_j|}{|\mathcal{R}_i \cup \mathcal{R}_j|}$$

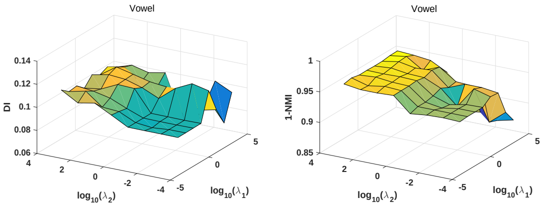
where $\mathcal{R}_i$ ($\mathcal{R}_j$) is the row-cluster of a co-cluster in $\mathcal{C}^{h_1}$ ($\mathcal{C}^{h_2}$). The column-wise CS has a similar definition. The smaller the CS is, the smaller the redundancy between the two co-clusterings is.

Both MultiCC and MCC-SS manifest low redundancy between co-clusters across co-clusterings, since both gene-wise and sample-wise scores are lower than 0.5. MCC-SS always has a much lower CS than MultiCC. This comparison further confirms that multiple co-clusterings are embedded in different subspaces, and MCC-SS can discover multiple diverse co-clusterings therein.

Table 3: Average co-cluster relevance score (CS) of four co-clusterings found by MCC-SS and MultiCC.

MCC-SS	CS_{12}	CS_{13}	CS_{14}	CS_{23}	CS_{24}	CS_{34}
Gene-wise	0.19	0.21	0.20	0.22	0.20	0.21
Sample-wise	0.22	0.23	0.22	0.21	0.20	0.22
MultiCC	CS_{12}	CS_{13}	CS_{14}	CS_{23}	CS_{24}	CS_{34}
Gene-wise	0.43	0.38	0.49	0.29	0.34	0.39
Sample-wise	0.28	0.29	0.34	0.29	0.33	0.34

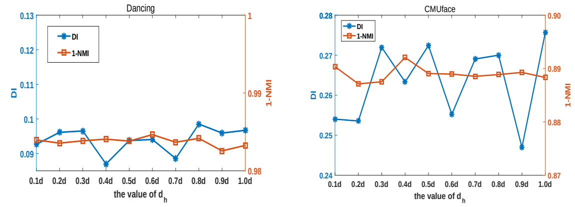
4.4 Parameter Sensitivity Analysis The regularization parameters λ_1 and λ_2 control the tradeoff between the quality and the diversity of m alternative clusterings. d_h controls the dimensionality of respective feature subspaces and also affects the diversity between alternative clusterings. We investigate parameter sensitivity by varying λ_1 and λ_2 in $[10^{-4}, 10^3]$ with $m = 2$. We take DI as the quality measure and 1-NMI as the diversity measure, and plot their values under different input values of λ_1 and λ_2 with $d_h = 10$ in Fig. 3 (Vowel). Similarly, we vary d_h from $0.1 \times d$ to d with $\lambda_1 = 100$, $\lambda_2 = 100$, and $m = 2$. For simplicity, we fix $d_{h_1} = d_{h_2}$. We show the values of DI and 1-NMI under different d_h in Figure 4 (Dancing). The other datasets give similar results.



(a) DI (b) 1-NMI

Figure 3: Quality and diversity of MCC-SS vs. λ_1 and λ_2 on the Vowel dataset.

MCC-SS has lower quality and diversity values when $\lambda_1, \lambda_2 \approx 0$, and it has significantly increased values when λ_1 and λ_2 are in other ranges. This is because



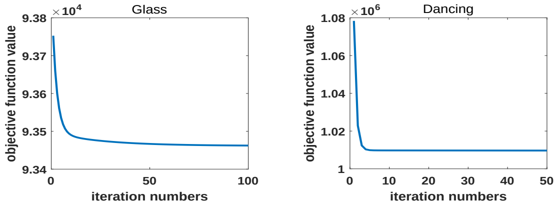
(a) Dancing (b) CMUface

Figure 4: Quality and diversity of MCC-SS vs. d_h on Dancing and CMUface datasets.

too small λ_1 or λ_2 values do not sufficiently account for the redundancy between alternative clusterings. A too large λ_1 value gives more weight to the diversity between sample-clusterings, but diverse clusterings can also be derived from different feature spaces. As such, the performance worsens under this extreme setting. λ_2 controls the redundancy between different subspaces, and MCC-SS often achieves a relatively good quality and diversity when a large λ_2 is used. This observation again indicates that multiple co-clusterings are embedded in different subspaces. Based on the above results, we adopt $\lambda_1 = \lambda_2 = 100$ for the experiments.

From Figure 4, we can see that MCC-SS is less affected by the input values of d_h than by those of λ_1 and λ_2. We also observe that $d_h \in [0.1d, 0.9d]$ sometimes gives better results than $d_h = d$. This observation corroborates the fact that multiple co-clusterings are embedded in different subspaces, and it is reasonable to explore alternative co-clusterings in different feature subspaces.

4.5 Convergence Analysis Figure 5 reveals the convergence trend on the Glass and Dancing datasets. We can observe that MCC-SS generally converges after 60 iterations, and this pattern holds on all datasets. Table 4 gives the runtimes of all methods. The experiments are conducted on a server with Linux OS 2.6.32, Intel Xeon E2675v3, and 256GB RAM; all methods are implemented in Matlab2014a.



(a) Glass (b) Dancing

Figure 5: Objective function convergence curve on Glass and Dancing datasets.

OSC is almost always the fastest and MNMF is slightly slower than OSC. These two methods are much faster than the other techniques. MCC-SS has a larger runtime than MultiCC, since is has one more matrix inversion $(\mathbf{P}_h)$, and this increases the iteration of $\mathbf{P}_h$. Dec-$k$means is based on k-means and MultiCC is based on NMF, so they have a small runtime cost. Both ADFT and MSC involve costly operations (i.e., dendrogram construction and feature weight computation). This is the main reason for their dramatic increase in computing time on high-dimensional datasets. In summary, MCC-SS holds a runtime similar to other competitive approaches, but frequently outperforms them in exploring multiple one-way and two-way clusterings.

5 Conclusions and Future Work

In this article, we study how to simultaneously find multiple co-clusterings in feature subspaces. This topic is challenging and rarely studied, but of interest and useful in practice. We introduced a matrix factorization based approach called MCC-SS to jointly explore multiple subspaces and co-clusterings therein. MCC-SS seeks multiple co-clusterings by semi-nonnegative matrix factorization and enforces diversity between the co-clusterings by minimizing redundancy between column clusterings and feature subspaces. Extensive experimental results show that MCC-SS is superior to other competitive and representative multiple clustering methods and can find many different meaningful co-clusterings. In the future, we will extend MCC-SS for integrative cancer genomic data analysis. The code of MCC-SS is available at http://mlda.swu.edu.cn/codes.php?name=MCC-SS.

References

[1] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, and C.-T. Lin, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, 2017.

[2] Y. Cheng and G. M. Church, "Biclustering of expression data." in *ISMB*, 2000, pp. 93–103.

[3] S. Li, A. Karatzoglou, and C. Gentile, "Collaborative filtering bandits," in *SIGIR*, 2016, pp. 539–548.

[4] J. Bailey, "Alternative clustering analysis: A review," in *Data Clustering: Algorithms and Applications*, A. Charu and R. Chandan, Eds. CRC Press, 2013, pp. 535–550.

[5] E. Bae and J. Bailey, "Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity," in *ICDM*, 2006, pp. 53–62.

[6] E. Bae, J. Bailey, and G. Dong, "A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings," *DMKD*, vol. 21, no. 3, pp. 427–471, 2010.

[7] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *TPAMI*, no. 2, pp. 224–227, 1979.

[8] D. Niu, J. G. Dy, and M. I. Jordan, "Multiple non-redundant spectral clustering views," in *ICML*, 2010, pp. 831–838.

Table 4: Runtimes on six datasets (seconds).

	Dec-kmeans	COALA	ADFT	MSC	OSC	MNMF	MultiCC	MCC-SS
Voewl	1.972	396.223	54.233	69.665	0.106	2.113	1.331	1.564
Ionosphere	0.625	112.556	12.134	43.654	0.095	1.336	0.552	0.765
Glass	0.962	21.356	1.963	15.336	0.076	0.335	1.765	1.622
Vehicle	2.952	2321.321	8.332	155.434	0.146	5.663	11.663	10.343
Crowdsourced	3.124	3422.231	3092.214	959.272	0.794	290.248	277.942	508.343
CMUface	1932.224	1324.211	87236.216	80495.556	124.695	175.226	2634.235	5324.752
Dancing	6.245	621.221	2580.848	1387.242	0.8524	3.196	156.234	220.332
Total	1948.104	8219.119	92985.94	83126.159	126.764	478.147	3083.722	6067.721

[9] S. Yang and L. Zhang, "Non-redundant multiple clustering by nonnegative matrix factorization," *Machine Learning*, vol. 106, no. 5, pp. 695–712, 2017.

[10] X. H. Dang and J. Bailey, "Generation of alternative clusterings using the cami approach," in *SDM*, 2010, pp. 118–129.

[11] R. Caruana, M. Elhawary, N. Nguyen, and C. Smith, "Meta clustering," in *ICDM*, 2006, pp. 107–118.

[12] P. Jain, R. Meka, and I. S. Dhillon, "Simultaneous unsupervised learning of disparate clusterings," *Statistical Analysis and Data Mining*, vol. 1, no. 3, pp. 195–210, 2008.

[13] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos, "Locally adaptive metrics for clustering high dimensional data," *DMKD*, vol. 14, no. 1, pp. 63–97, 2007.

[14] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *KDD*, 2001, pp. 269–274.

[15] J. Xie, A. Ma, A. Fennell, Q. Ma, and J. Zhao, "It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data," *Briefings in Bioinformatics*, vol. 1, p. 16, 2018.

[16] X. Wang, G. Yu, C. Domeniconi, J. Wang, Z. Yu, and Z. Zhang, "Multiple co-clusterings," in *ICDM*, 2018, pp. 1308–1313.

[17] T. Tokuda, J. Yoshimoto, Y. Shimizu, and et al., "Multiple co-clustering based on nonparametric mixture models with heterogeneous marginal distributions," *PLoS ONE*, vol. 12, no. 10, p. e0186566, 2017.

[18] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *TPAMI*, vol. 32, no. 1, pp. 45–55, 2010.

[19] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *JMLR*, vol. 3, no. 12, pp. 583–617, 2002.

[20] F. Gullo, C. Domeniconi, and A. Tagarelli, "Projective clustering ensembles," *DMKD*, vol. 26, no. 3, pp. 452–511, 2013.

[21] S. Bickel and T. Scheffer, "Multi-view clustering," in *ICDM*, vol. 4, 2004, pp. 19–26.

[22] I. Davidson and Z. Qi, "Finding alternative clusterings using constraints," in *ICDM*, 2008, pp. 773–778.

[23] J. Hu, Q. Qian, J. Pei, R. Jin, and S. Zhu, "Finding multiple stable clusterings," *KAIS*, vol. 51, no. 3, pp. 991–1021, 2017.

[24] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, "Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering," in *KDD*, 2005, pp. 41–50.

[25] J. Kawale and D. Boley, "Constrained spectral clustering using l1 regularization," in *SDM*, 2013, pp. 103–111.

[26] H. Shan and A. Banerjee, "Bayesian co-clustering," in *ICDM*, 2008, pp. 530–539.

[27] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *KDD*, 2003, pp. 89–98.

[28] Q. Gu and J. Zhou, "Co-clustering on manifolds," in *KDD*, 2009, pp. 359–368.

[29] H. Wang, F. Nie, H. Huang, and F. Makedon, "Fast nonnegative matrix tri-factorization for large-scale data co-clustering," in *IJCAI*, 2011, pp. 1553–1558.

[30] X. Wang, J. Wang, C. Domeniconi, G. Yu, G. Xiao, and M. Guo, "Multiple independent subspace clusterings," in *AAAI*, 2019, pp. 1–8.

[31] Y. Cui, X. Z. Fern, and J. G. Dy, "Non-redundant multi-view clustering via orthogonalization," in *ICDM*, 2007, pp. 133–142.

[32] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2001, pp. 556–562.

[33] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[34] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[35] B. A. Johnson and K. Iizuka, "Integrating openstreetmap crowdsourced data and landsat time-series imagery for rapid land use/land cover (lulc) mapping: Case study of the laguna de bay area of the philippines," *Applied Geography*, vol. 67, pp. 140–149, 2016.

[36] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," *Machine Learning*, vol. 52, no. 1, pp. 91–118, 2003.

[37] A. Rosenwald, G. Wright *et al.*, "The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma," *NEJM*, vol. 346, no. 25, pp. 1937–1947, 2002.

[38] A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics*, vol. 22, no. 9, pp. 1122–1129, 2006.