Multi-view Sparse Co-clustering via Proximal Alternating Linearized Minimization

Jiangwen Sun JAVON@ENGR.UCONN.EDU
Jin Lu JIN.LU@ENGR.UCONN.EDU
Tingyang Xu TIX11001@ENGR.UCONN.EDU
Jinbo Bi JINBO@ENGR.UCONN.EDU
Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269 USA

Abstract

When multiple views of data are available for a set of subjects, co-clustering aims to identify subject clusters that agree across the different views. We explore the problem of co-clustering when the underlying clusters exist in different subspaces of each view. We propose a proximal alternating linearized minimization algorithm that simultaneously decomposes multiple data matrices into sparse row and columns vectors. This approach is able to group subjects consistently across the views and simultaneously identify the subset of features in each view that are associated with the clusters. The proposed algorithm can globally converge to a critical point of the problem. A simulation study validates that the proposed algorithm can identify the hypothesized clusters and their associated features. Comparison with several latest multi-view co-clustering methods on benchmark datasets demonstrates the superior performance of the proposed approach.

1. Introduction

Multi-view data is common in many scientific fields. A disease subtype may be recognized using both clinical symptoms (view 1) and genomic data (view 2) (Gelernter et al., 2006). Because only a very small portion of genetic variants contribute to a particular disease subtype, feature selection is indispensible in order to derive disease subtypes (clusters) that are characterized by specific clinical symptoms and selected genetic markers. Cross-species RNA analysis is used to detect clusters of genes that are co-regularized at certain developmental stages across the species (Jiang et al., 2014). Each species corresponds to a RNA data matrix (a view). The gene clusters consistent across all species can only exist in subspaces of each view (the specific set of stages when they are co-regulated). However, Feature learning for multi-view consistent clustering is an under-explored problem.

Existing multi-view data analyses include supervised/semi-supervised co-training (Blum & Mitchell, 1998; Balcan et al., 2005; Yu et al., 2011), unsupervised co-clustering (Guan et al., 2011; Sohn & Xing, 2009; Van Mechelen et al., 2004; Kumar & Daume III, 2011; Kumar et al., 2011; Ji et al., 2012) and multi-view feature learning (White et al., 2012; Wang et al., 2013), in all of which samples are characterized or viewed in multiple ways, thus creating multiple sets of input variables. Co-clustering commonly comprises two subareas: (1) biclustering (Ji et al., 2012; Dhillon et al., 2003; Shan & Banerjee, 2008), also called two-mode clustering (Van Mechelen et al., 2004), simultaneously clusters the rows and columns of a data matrix; (2) multi-view co-clustering (Culp & Michailidis, 2009; Kumar & Daume III, 2011; Kumar et al., 2011; Chaudhuri et al., 2009; Cheng et al., 2013; Cai et al., 2013; Liu et al., 2013; Sun et al., 2014) seeks groupings that are consistent across different views. The first type of co-clustering is similar to another set of algorithms (Niu et al., 2010; 2012; Guan et al., 2011) that search subspaces in the problem dimension, and each of the subspaces corresponds to a view of the data and produces a different cluster solution. Biclustering and subspace searching essentially find subspaces to define distinct clusters.

Our problem differs from the general multi-view feature learning, such as in (Wang et al., 2013), which aims to cluster subjects based on heterogeneous data, and it does not enforce the selection separately in different views. The selected features may hence come from a subset of the views (e.g. from one view). If the clusters are obtained using only features in one view, they are not truly multi-view consistent clusters. Most similar to multi-view co-clustering, our

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

problem seeks a grouping of subjects that agrees in the different views. However, existing multi-view co-clustering methods assume that all given variables in each view are equally useful to reveal an underlying partition. When clusters exist in different subspaces, an underlying partition consistent across all views may only be identified in the different subspaces of each view.

Hence, we propose a novel sparse co-clustering approach by simultaneously decomposing multiple data matrices into products of *sparse* row and column vectors. This method can be viewed as performing biclustering in each view to identify both row clusters and column clusters simultaneously but the row clusters from the different views should be the same. Figure 1 demonstrates the problem in two views with two data matrices. If rows of a data matrix represent subjects and columns represent features, we identify subject (row) clusters consistent across all the views and variables that define the subject clusters from each view.

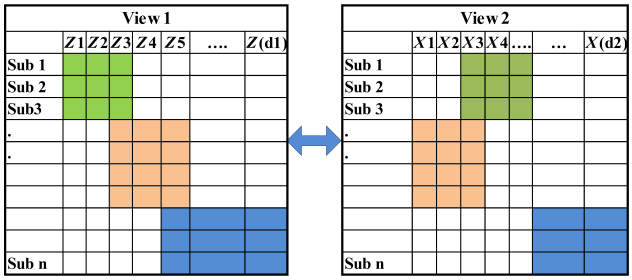

Figure 1. Sparse co-clustering: rows are grouped in the same way across the two matrices. The subjects in each row cluster are homogeneous over a subset of variables from each of the views.

The main contributions of our work are summarized as follows. (1) Unlike most existing multi-view data analytics, our method can find subspaces in individual views so that a multi-view consistent grouping can be identified. (2) A recently proposed multi-view co-clustering method (Sun et al., 2014) uses sparse singular value decomposition (SSVD) (Lee et al., 2010) and has no convergence guarantee. However, a globally convergent algorithm is developed to solve our formulation by proximal alternating linearized minimization (Bolte et al., 2014). (3) Existing SSVD based methods use the ℓ_1-norm to approximate the ℓ_0-norm in forming the sparse decomposition (Sun et al., 2013; 2014). We directly solve a formulation with the ℓ_0 penalty, which is essential for clustering.

2. Multi-view low-rank matrix approximation

Given a single data matrix $\mathbf{X}$ of size n-by-d, a subgroup of its rows and a subgroup of its columns can be simultaneously achieved by decomposing the matrix into a pair of left and right vectors that are both sparse. Let $\mathbf{u}$ of size n and $\mathbf{v}$ of size d be the left and right vector, respectively, resulted from the decomposition. Their outer product forms a sparse rank one approximation of the original matrix, i.e., $\mathbf{X} \approx \mathbf{u}\mathbf{v}^T$. Then, rows in $\mathbf{X}$ corresponding to non-zero components in $\mathbf{u}$ form a row subgroup and columns in $\mathbf{X}$ corresponding to non-zero components in $\mathbf{v}$ form a column subgroup. The resultant row and column clusters help to define each other. Subsequent clusters can be obtained by a sparse rank-one approximation of the deflated data matrix, i.e., by repeatedly solving Eq(1) with an updated $\mathbf{X}$:

$$\begin{aligned} \min_{\mathbf{u},\mathbf{v}} \quad & \|\mathbf{X} - \mathbf{u}\mathbf{v}^T\|_F^2 \\ \text{subject to} \quad & \|\mathbf{u}\|_0 \le s_u, \|\mathbf{v}\|_0 \le s_v, \end{aligned} \tag{1}$$

where the objective function measures the approximation error with a Frobenius norm of the difference matrix, and $\|\cdot\|_0$ is the ℓ_0 vector norm that returns the number of non-zeros in a vector. We assume the singular value is absorbed by the singular vectors. The hyper-parameters s_u and s_v are pre-determined to enforce the sparsity of $\mathbf{u}$ and $\mathbf{v}$, respectively. To obtain subsequent singular vectors, we update the matrix $\mathbf{X}$ by excluding subjects already identified in a row cluster. (Other choices exist, such as deflating $\mathbf{X}$ by $\mathbf{X} - \mathbf{u}\mathbf{v}^T$, which leads to overlapping row clusters.)

Now we extend this method to two or more data matrices denoted by $\mathbf{X}^k$ of size n-by-d_k, $k = 1, \cdots, m$. These m data matrices characterize the same set of subjects from m different views. We can obtain $\mathbf{u}^k$ and $\mathbf{v}^k$ for each matrix $\mathbf{X}^k$ by finding their rank-one approximations separately, but it will not guarantee the row clusters specified, respectively, by $\mathbf{u}^k$ be consistent. To make them consistent, it requires all $\mathbf{u}^k$, $k = 1, \cdots, m$, to have non-zero entries at the same positions. Note that by requiring a sparse common $\mathbf{u}$ across all views, consistent row clusters can also be identified and the related optimization problem may be easier to solve. However, it imposes a stringent constraint to limit the search space only to those that satisfy $\mathbf{u}^1 = \mathbf{u}^2 = \cdots = \mathbf{u}^m$, which can be difficult when different kinds of data are used, such as real-valued data in gene expression but discrete values in genomic markers, and rules out many potential solutions that may include the optimal row clusters. Another alternative is to minimize the pairwise differences $||\mathbf{u}^i - \mathbf{u}^j||$, which suffers from the same over-constrained problem as the exact values of the difference are not concerned. Our problem only seeks the indicators of whether or not a component of $\mathbf{u}$ is zero.

We propose to use a binary vector $\boldsymbol{\omega}$ of size n that serves as a common factor to link the different views. We multiply each component of $\mathbf{u}^k$ by the corresponding component of $\boldsymbol{\omega}$, i.e., $u_i^k = u_i^k \omega_i$. In other words, we represent each vector $\mathbf{u}^k$ by $\text{diag}(\boldsymbol{\omega})\mathbf{u}^k$ where $\text{diag}(\boldsymbol{\omega})$ is a diagonal matrix with diagonal entries equal to $\boldsymbol{\omega}$. When $\omega_i = 0$, regardless the value of the i-th components of all $\mathbf{u}^k$'s, the i-th row will be excluded from the subgroup in all views. We hence require the sparsity of $\boldsymbol{\omega}$ instead of individual $\mathbf{u}$'s in

the optimization problem as follows:

$$\begin{aligned} \min_{\boldsymbol{\omega},\mathbf{u}^k,\mathbf{v}^k,k=1,\cdots,m} \quad & \sum_{k=1}^{m} \|\mathbf{X}^k - \mathrm{diag}(\boldsymbol{\omega})\mathbf{u}^k{\mathbf{v}^k}^T\|_F^2 \\ \text{subject to} \quad & \|\boldsymbol{\omega}\|_0 \le s_\omega,\ \|\mathbf{v}^k\|_0 \le s_{v^k}, \\ & k = 1,\cdots,m, \\ & \boldsymbol{\omega} \in \mathcal{B}_n. \end{aligned} \tag{2}$$

where $\mathcal{B}_n$ is the set that contains all binary vectors of length n, s_ω and s_{v^k}'s are hyper-parameters that are predetermined to enforce sparsity of $\boldsymbol{\omega}$ and $\mathbf{v}^k$'s. Note that minimizing (2) is mathematically equivalent to minimizing $\sum_{k=1}^{m}(\|\mathbf{X}^k - \mathrm{diag}(\boldsymbol{\omega})\mathbf{u}^k{\mathbf{v}^k}^T\|_F^2 + \lambda_{v^k}\|\mathbf{v}^k\|_0) + \lambda_\omega\|\boldsymbol{\omega}\|_0$ where λ_ω and λ_{v^k}'s correspond to the optimal values of Lagrange multipliers of Eq.(2). Although this kind of problems are nonconvex and nonsmooth minimization problems, recent advances in proximal algorithms (Bolte et al., 2014; Attouch et al., 2010; 2013) allow us to derive an efficient algorithm to solve the proposed formulation.

3. A proximal alternating algorithm

The framework of proximal alternating linearized minimization (PALM) (Bolte et al., 2014) has recently been proposed to solve an optimization problem with multiple blocks of variables, and the objective function is only required to be smooth at the term that uses all variables. Typical proximal projection methods (Bauschke & Combettes, 2011) require the gradient of the smooth part of the objective function to be Lipchitz continuous with respect to all variables. In contrast, PALM only requires componentwise Lipchitz continuity, i.e., Lipchitz continuous with respect to a block of variables when others are fixed. The objective function of Eq.(2) is in C^2, i.e., second-order continuously differentiable with respect to both $\mathbf{u}$'s and $\mathbf{v}$'s, and with respect to $\boldsymbol{\omega}$ when we relax it to a real-valued vector. The regularizers based on the ℓ_0 norm are the nonsmooth parts and they only use one of the blocks at a time.

PALM alternates between optimizing each block of the variables $\boldsymbol{\omega}$, $\mathbf{u}$'s and $\mathbf{v}$'s. The central idea of PALM is to, for each block of variables, perform one gradient step on the smooth part, while a proximal step is taken on the nonsmooth part. To simplify the presentation, we denote the objective function in Eq.(2) by h and the regularizers on $\boldsymbol{\omega}$ and $\mathbf{v}$'s, respectively, by f and g's. Let $\boldsymbol{\omega}^t$, $(\mathbf{u}^k)^t$ and $(\mathbf{v}^k)^t$ be the current iterates at iteration t. For instance, to optimize $\boldsymbol{\omega}$, a linearized approximation of the objective function, which is the gradient step, is $< \boldsymbol{\omega} - \boldsymbol{\omega}^t, \nabla_{\boldsymbol{\omega}} h >$ where $\nabla_{\boldsymbol{\omega}} h$ is the partial derivatives of h with respect to $\boldsymbol{\omega}$. Then, $\mathrm{argmin}\{< \boldsymbol{\omega} - \boldsymbol{\omega}^t, \nabla_{\boldsymbol{\omega}} h > + \frac{\gamma_\omega L_{\boldsymbol{\omega}}}{2}\|\boldsymbol{\omega} - \boldsymbol{\omega}^t\|_2 :\ \|\boldsymbol{\omega}\|_0 \le s_\omega\}$ is a *well-defined* proximal map for f where $\gamma_\omega > 1$ is a constant and $L_{\boldsymbol{\omega}}$ is the Lipschitz modulis of $\nabla_{\boldsymbol{\omega}} h$. (Note all partial derivatives of h are Lipschitz continuous.) With similar notation, we now describe as follows the procedure to update the variables in iteration $t+1$.

(1) Compute $(\mathbf{u}^k)^{t+1}$ using $\boldsymbol{\omega}^t$, $(\mathbf{u}^k)^t$ and $(\mathbf{v}^k)^t$

As the update of $\mathbf{u}$'s are independent from each other, each $(\mathbf{u}^k)^{(t+1)}$ can be calculated separately. Similarly, let $\nabla_{\mathbf{u}^k} h$ be the partial derivatives of h at point $(\boldsymbol{\omega}^t, (\mathbf{u}^k)^t, (\mathbf{v}^k)^t$ with respect to $\mathbf{u}^k$, and it can be calculated as:

$$\nabla_{\mathbf{u}^k} h = \boldsymbol{\omega}^t \odot \left(\left((\boldsymbol{\omega}^t \odot (\mathbf{u}^k)^t)(\mathbf{v}^k)^{t^T} - \mathbf{X}^k\right)(\mathbf{v}^k)^t\right).$$

where $\odot$ computes the element-wise product of two vectors. The Lipschitz modulis of $\nabla_{\mathbf{u}^k} h$ is calculated by $L_{\mathbf{u}^k} = (\mathbf{v}^k)^{t^T}(\mathbf{v}^k)\|\boldsymbol{\omega}^t \odot \boldsymbol{\omega}^t\|_2$. Then we compute $(\mathbf{u}^k)^{t+1}$ by solving the following optimization problem:

$$\min_{\mathbf{u}^k} \quad < \mathbf{u}^k - (\mathbf{u}^k)^t, \nabla_{\mathbf{u}^k} h > + \frac{\gamma_u L_{\mathbf{u}^k}}{2}\|\mathbf{u}^k - (\mathbf{u}^k)^t\|_2.$$

where $\gamma_u > 1$ is a constant and note that there is no nonsmooth part due to no regularizer on $\mathbf{u}$. This problem can be easily proved to be equivalent to:

$$\min_{\mathbf{u}^k} \quad \frac{\gamma_u L_{\mathbf{u}^k}}{2}\|\mathbf{u}^k - \left((\mathbf{u}^k)^t - \frac{1}{\gamma_u L_{\mathbf{u}^k}}\nabla_{\mathbf{u}^k} h\right)\|_2,$$

which has an analytical solution as:

$$(\mathbf{u}^k)^{t+1} = (\mathbf{u}^k)^t - \frac{1}{\gamma_u L_{\mathbf{u}^k}}\nabla_{\mathbf{u}^k} h \tag{3}$$

(2) Compute $(\mathbf{v}^k)^{t+1}$ using $\boldsymbol{\omega}^t$, $(\mathbf{u}^k)^{t+1}$ and $(\mathbf{v}^k)^t$

Similarly, each $\mathbf{v}^k$ can also be computed separately. We compute the partial derivatives $\nabla_{\mathbf{v}^k} h$ and the Lipschitz modulis $L_{\mathbf{v}^k}$ as follows.

$$\nabla_{\mathbf{v}^k} h = \left((\boldsymbol{\omega}^t \odot (\mathbf{u}^k)^{t+1})(\mathbf{v}^k)^t - \mathbf{X}^k\right)^T\left(\boldsymbol{\omega}^t \odot (\mathbf{u}^k)^{t+1}\right).$$

and $L_{\mathbf{v}^k} = \left(\boldsymbol{\omega}^t \odot (\mathbf{u}^k)^{t+1}\right)^T\left(\boldsymbol{\omega}^t \odot (\mathbf{u}^k)^{t+1}\right)$. In order to obtain the update for $\mathbf{v}^k$, we solve the proximal map:

$$\begin{aligned} \min_{\mathbf{v}^k} \quad & < \mathbf{v}^k - (\mathbf{v}^k)^t, \nabla_{\mathbf{v}^k} h > + \frac{\gamma_v L_{\mathbf{v}^k}}{2}\|\mathbf{v}^k - (\mathbf{v}^k)^t\|_2 \\ & \text{subject to} \quad \|\mathbf{v}^k\|_0 \le s_{v^k}. \end{aligned}$$

Let $\delta_s(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ be an indicator function defined by:

$$\delta_s(\mathbf{x}) = \begin{cases} 0 & \|\mathbf{x}\|_1 \le s \\ +\infty & \|\mathbf{x}\|_1 > s. \end{cases} \tag{4}$$

Then, the above minimization problem can be converted to:

$$\begin{aligned} \min_{\mathbf{v}^k} \quad & < \mathbf{v}^k - (\mathbf{v}^k)^t, \nabla_{\mathbf{v}^k} h > \\ & + \frac{\gamma_v L_{\mathbf{v}^k}}{2}\|\mathbf{v}^k - (\mathbf{v}^k)^t\|_2 + \delta_{s_{v^k}}(\mathbf{v}^k). \end{aligned}$$

This problem can be proved to be equivalent to the following problem:

$$\min_{\mathbf{v}^k} \quad \frac{\gamma_v L_{\mathbf{v}^k}}{2}\|\mathbf{v}^k - \left((\mathbf{v}^k)^t - \frac{1}{\gamma_v L_{\mathbf{v}^k}}\nabla_{\mathbf{v}^k} h\right)\|_2 + \delta_{s_{v^k}}(\mathbf{v}^k). \tag{5}$$

Let

$$(\tilde{\mathbf{v}}^k)^{t+1} = (\mathbf{v}^k)^t - \frac{1}{\gamma_v L_{\mathbf{v}^k}}\nabla_{\mathbf{v}^k} h.$$

It can be shown that the optimal solution to Eq.(5) is the vector that keeps the original values in $(\tilde{\mathbf{v}}^k)^{t+1}$ at the positions whose absolute values are among the largest s_{v^k} of them. For instance, if s_{v^k} is 3, we rank the components in $(\tilde{\mathbf{v}}^k)^{t+1}$ in descending order according to their absoute values, and then choose the top three to maintain their values and set the rest to be 0. We denote the corresponding threshold by α which is the minimum value among the s_{v^k} largest absolute values in $(\tilde{\mathbf{v}}^k)^{t+1}$, and compute $(\mathbf{v}^k)^{t+1}$ as follows:

$$(\mathbf{v}^k)_i^{t+1} = \begin{cases} (\tilde{\mathbf{v}}^k)_i^{t+1} & |(\tilde{\mathbf{v}}^k)_i^{t+1}| \geq \alpha \\ 0 & |(\tilde{\mathbf{v}}^k)_i^{t+1}| < \alpha. \end{cases} \tag{6}$$

(3) Compute $(\boldsymbol{\omega}^k)^{t+1}$ using $\boldsymbol{\omega}^t$, $(\mathbf{u}^k)^{t+1}$ and $(\mathbf{v}^k)^{t+1}$

We compute the partial derivatives $\nabla_{\boldsymbol{\omega}} h$ and the Lipschitz modulis $L_{\boldsymbol{\omega}}$ as follows:

$$\nabla_{\boldsymbol{\omega}} h = \sum_k \left(\left(\left(\boldsymbol{\omega}^t \odot (\mathbf{u}^k)^{t+1}\right)(\mathbf{v}^k)^{t+1^T} - \mathbf{X}^k\right)(\mathbf{v}^k)^{t+1}\right) \odot (\mathbf{u}^k)^{t+1}.$$

and $L_{\boldsymbol{\omega}} = \|\sum_k \left((\mathbf{v}^k)^{t+1^T}(\mathbf{v}^k)^{t+1}\right)(\mathbf{u}^k)^{t+1} \odot (\mathbf{u}^k)^{t+1}\|_2$. We solve the following optimization problem for $\boldsymbol{\omega}^{t+1}$:

$$\begin{aligned} \min_{\boldsymbol{\omega}} \quad & < \boldsymbol{\omega} - \boldsymbol{\omega}^t, \nabla_{\boldsymbol{\omega}} h > + \frac{\gamma_\omega L_{\boldsymbol{\omega}}}{2}\|\boldsymbol{\omega} - \boldsymbol{\omega}^t\|_2 \\ \text{subject to} \quad & \|\boldsymbol{\omega}\|_0 \leq s_\omega. \end{aligned}$$

By introducing the indicator function δ as in Eq. (4), this problem can be converted to:

$$\min_{\boldsymbol{\omega}} \quad < \boldsymbol{\omega} - \boldsymbol{\omega}^t, \nabla_{\boldsymbol{\omega}} h > + \frac{\gamma_\omega L_{\boldsymbol{\omega}}}{2}\|\boldsymbol{\omega} - \boldsymbol{\omega}^t\|_2 + \delta_{s_\omega(\boldsymbol{\omega})},$$

which can be shown is equivalent to:

$$\min_{\boldsymbol{\omega}} \quad \frac{\gamma_\omega L_{\boldsymbol{\omega}}}{2}\|\boldsymbol{\omega} - \left((\boldsymbol{\omega})^t - \frac{1}{\gamma_\omega L_{\boldsymbol{\omega}}}\nabla_{\boldsymbol{\omega}} h\right)\|_2 + \delta_{s_\omega}(\boldsymbol{\omega}).$$

Similar to how we update $\mathbf{v}^k$, we let

$$\tilde{\boldsymbol{\omega}}^{t+1} = \boldsymbol{\omega}^t - \frac{1}{\gamma_\omega L_{\boldsymbol{\omega}}}\nabla_{\boldsymbol{\omega}} h$$

and β be the minimum value among the largest s_ω absolute values in $\tilde{\boldsymbol{\omega}}^{t+1}$. We compute $\boldsymbol{\omega}^{t+1}$ as follows:

$$\boldsymbol{\omega}_i^{t+1} = \begin{cases} \tilde{\boldsymbol{\omega}}_i^{t+1} & |\tilde{\boldsymbol{\omega}}_i^{t+1}| \geq \beta \\ 0 & |\tilde{\boldsymbol{\omega}}_i^{t+1}| < \beta. \end{cases} \tag{7}$$

Summary. Algorithm 1 summarizes all the steps in our algorithm. Row and column clusters will evolve from its outputs on $\boldsymbol{\omega}$ and $\mathbf{v}^k$. Precisely, we take rows corresponding to non-zero entries in $\boldsymbol{\omega}$ to form a row subgroup, and take columns corresponding to non-zero entries $\mathbf{v}^k$ to form a column subgroup in the k^{th} view. By repeating this procedure on updated data matrices where subjects already in a row cluster are excluded, the desired number of subject (row) clusterss can be obtained.

Algorithm 1 Multi-view rank one matrix approximation

Input: $\mathbf{X}^{k:=1,\cdots,m}$, s_ω and $s_{v_{k:=1,\cdots,m}}$
Output: $\boldsymbol{\omega}$, $\mathbf{u}_i$ and $\mathbf{v}_i$ for $i = 1, \cdots, m$
1. Initialize $\boldsymbol{\omega}^0$, and $(\mathbf{v}^k)^0$, $(\mathbf{u}^k)^0$ for all $k = 1, \cdots, m$.
2. Compute $(\mathbf{u}^k)^t$, $\forall k = 1, \cdots, m$ according to Eq. (3).
3. Compute $(\mathbf{v}^k)^t$, $\forall k = 1, \cdots, m$ according to Eq. (6).
4. Compute $\boldsymbol{\omega}^t$ according to Eq. (7)
Repeat steps 2 - 4 until convergence (e.g., until $||\boldsymbol{\omega}^{t+1} - \boldsymbol{\omega}^t|| \leq \varepsilon$, $||(\mathbf{u}^k)^{t+1} - (\mathbf{u}^k)^t|| \leq \varepsilon$, and $||(\mathbf{v}^k)^{t+1} - (\mathbf{v}^k)^t|| \leq \varepsilon$.)

Computation Cost. Algorithm 1 is efficient and scalable. At each iteration, only simple closed-form solutions need to be computed for each block of variables. When calculating the updates for the m pairs of $\mathbf{u}^k$ and $\mathbf{v}^k$, the algorithm requires a computation cost of $O(nmd)$. For updating $\boldsymbol{\omega}$, the most costly steps are the calculation of partial derivatives of h with respect to $\boldsymbol{\omega}$, which requires a computation cost of $O(nmd)$. Overall, this algorithm takes computation time of $O(nmd)$, which is in the linear order of the problem dimensions. Moreover, notice that the calculation of the update for $\mathbf{u}^k$ and $\mathbf{v}^k$ is independent from each other among views. Hence, this algorithm is readily parallelizable and can be distributed if more processors are available to further reduce the computation time.

4. Convergence analysis

Based on the results of (Bolte et al., 2014; Attouch et al., 2013), we prove that Algorithm 1 globally converges to a critical point of Eq.(2). As how Algorithm 1 is derived, it is easy to see that Eq.(2) is equivalent to minimizing an overall objective function $\Phi(\boldsymbol{\omega}, \mathbf{u}^k, \mathbf{v}^k) = h(\boldsymbol{\omega}, \mathbf{u}^k, \mathbf{v}^k) + \delta_{s_\omega}(\|\omega\|_0) + \sum_k \delta_{s_{v_k}}(\|\mathbf{v}^k\|_0)$. Theorem 1 characterizes our main result of convergence property.

Theorem 1 *Let $\mathbf{z}$ be the vector consisting of all variables in Problem (2), and $\{\mathbf{z}^t\}$ be a sequence generated by Al-*

gorithm 1. Then the sequence $\{\mathbf{z}^t\}$ *has finite length and converges to a critical point of* Φ.

The proof of Theorem 1 follows from the main result of (Bolte et al., 2014) that shows if several key properties of Φ are satisfied, then the conclusion holds . The following two lemmas establish the properties.

Lemma 1 *Φ is a semi-algebraic function, and hence satisfies the Kurdyka-Lojasiewicz (KL) property. (Please see Appendix A for the definitions).*

Proof. It has been shown in (Bolte et al., 2014) that the ℓ_0 norm $\|\mathbf{x}\|_0$ is semi-algebraic. Let d be the length of $\mathbf{x}$, the range of $\|\mathbf{x}\|_0$ is $[0, d]$, which is a semi-algebraic set. According to (Attouch et al., 2013), the indicator function of a semi-algebraic set is semi-algebraic, and any composition of semi-algebraic functions remain to be semi-algebraic. Hence, all the δ functions in Φ, i.e., δ_{s_ω} and δ_{u^k}, are semi-algebraic. As function h is a real-valued polynomial function, it is semi-algebraic as well. According to the property of semi-algebraic functions: finite sums of semi-algebraic functions remain semi-algebraic, the function Φ, as the sum of the three items, is a semi-algebraic.

In addition, it can be shown that Φ is a proper and lower semicontinuous function. Then, based on Theorem 3 in (Bolte et al., 2014): a proper, lower semicontinuous and semi-algebraic function satisfies the KL property, so Φ satisfies the KL property. ∎

Lemma 2 *The function Φ sastifies all of the following properties:*

1. $\inf_{\mathbb{R}^{(m+1)nd}} h > -\infty$, $\inf_{\mathbb{R}^n} \delta_{s_\omega} > -\infty$ *and* $\inf_{\mathbb{R}^{d^k}} \delta_{s_{v^k}} > -\infty$, *where d^k is the number of features in view k and $d = \sum d^k$.*

2. $\nabla_{\boldsymbol{\omega}} h$, $\nabla_{\mathbf{u}^k} h$ *and* $\nabla_{\mathbf{v}^k} h$ *are Lipschitz continuous with modulis $L_{\boldsymbol{\omega}}$, $L_{\mathbf{u}^k}$ and $L_{\mathbf{v}^k}$, respectively.*

3. *there exists $\lambda_{\boldsymbol{\omega}}^{-/+}$, $\lambda_{\mathbf{u}^k}^{-/+}$ and $\lambda_{\mathbf{v}^k}^{-/+}$ such that*

$$\inf L_{\boldsymbol{\omega}} \geq \lambda_{\boldsymbol{\omega}}^-, \quad \inf L_{\mathbf{u}^k} \geq \lambda_{\mathbf{u}^k}^-, \quad \inf L_{\mathbf{v}^k} \geq \lambda_{\mathbf{v}^k}^- \tag{8}$$

$$\sup L_{\boldsymbol{\omega}} \leq \lambda_{\boldsymbol{\omega}}^+, \quad \sup L_{\mathbf{u}^k} \leq \lambda_{\mathbf{u}^k}^+, \quad \sup L_{\mathbf{v}^k} \leq \lambda_{\mathbf{v}^k}^+ \tag{9}$$

4. *The entire ∇h is Lipschitz continuous on bounded subsets of $\mathbb{R}^{(m+1)nd}$.*

Proof. Property 1 is trivial due to the non-negativeness of each item in our objective function Φ.

For Property 2, notice that

$$\nabla_{\mathbf{v}^k} h(\boldsymbol{\omega}, \mathbf{u}^k, \mathbf{v}^k) = (\boldsymbol{\omega} \odot \mathbf{u}^k)^T[(\boldsymbol{\omega} \odot \mathbf{u}^k)\mathbf{v}^k - \mathbf{X}^k]$$
$$\nabla_{\mathbf{u}^k} h(\boldsymbol{\omega}, \mathbf{u}^k, \mathbf{v}^k) = [\text{diag}(\boldsymbol{\omega})(\text{diag}(\boldsymbol{\omega})\mathbf{u}^k\mathbf{v}^k - \mathbf{X}^k)\mathbf{v}^{kT}]$$
$$\nabla_{\boldsymbol{\omega}} h(\boldsymbol{\omega}, \mathbf{u}^k, \mathbf{v}^k) = \sum_i [\text{diag}(\mathbf{u}^i)(\text{diag}(\mathbf{u}^i)\boldsymbol{\omega}\mathbf{v}^i - \mathbf{X}^i)\mathbf{v}^{iT}]$$

which are all Lipschitz continuous with the respecitve Lipschitz modulis:

$$L_{\mathbf{v}^k} = \|(\boldsymbol{\omega} \odot \mathbf{u}^k)^T(\boldsymbol{\omega} \odot \mathbf{u}^k)\|_F$$
$$L_{\mathbf{u}^k} = \|(\mathbf{v}^k\mathbf{v}^{kT})(\boldsymbol{\omega} \odot \boldsymbol{\omega})\|_F$$
$$L_{\boldsymbol{\omega}} = \sum_i \|(\mathbf{v}^i\mathbf{v}^{iT})(\mathbf{u}^i \odot \mathbf{u}^i)\|_F.$$

To prove Property 3, we introduce an arbitrary positive constant μ and define

$$L'_s = \max\{L_s, \mu\}, \forall s = \mathbf{v}^k, \mathbf{u}^k, \boldsymbol{\omega}$$

This functions L'_s is still some Lipschitz moduli of ∇h_s, and it is bounded from below by μ. Hence, we have $\lambda_s^- = \mu$ for all s, and

$$\inf\{L_s\} \geq \mu, \ \forall s = \mathbf{v}^k, \mathbf{u}^k, \boldsymbol{\omega}.$$

The upper bound is discussed below.

Property 4 is satisfied directly from the Mean Value Theorem because h is in C^2. If the subsets $B \times \prod_i B^{n_i} \times \prod_i B^{m_i}$ are bounded, we similarly implement the Mean Value Theorem, and we obtain that $\sup\{L_s\}$ is bounded from above if the generated sequence $\{\mathbf{z}^t\}$ is bounded. ∎

Based on the above properties and the KL property, applying the result of (Bolte et al., 2014) yields Theorem 1.

5. Experiments

We implemented our approach using Matlab and validated it first on synthetic data that was simulated with known row and column clusters. This simulation study was particularly designed to examine whether or not our algorithm could reveal the underlying variables associated with the clusters even when a high level of noise existed. Then we evaluated our approach on two benchmark datasets with known subject clusters but unknown feature clusters. Normalized mutual information (NMI, ranging from 0 to 1) calculates the mutual information between two cluster solutions normalized by the cluster entropies. Since the true subject (row) clusters were known in our datasets, we computed NMI to measure the agreement between the true clusters and the clusters obtained by each approach in comparison. A higher NMI value indicated better performance.

We compared the proposed approach against several recent and most relevant multi-view co-clustering methods:

- **Single view sparse SVD** (Lee et al., 2010): We reported the best performance among the results by running SSVD in individual views as a baseline.
- **Co-regularized spectral** (Kumar et al., 2011): This method finds consistent row clusters across multiple views by applying spectral clustering to each view together with a co-regularization factor applied to the eigenvector representations of different views. We used the pairwise co-regularized formulation in (Kumar et al., 2011).
- **Co-trained spectral** (Kumar & Daume III, 2011): This method also finds consistent row clusters based on spectral clustering where eigenvector representations of each view are co-trained or modified by the clustering results from other views.
- **Multi-view canonical correlation analysis (CCA) clustering** (Chaudhuri et al., 2009): This method constructs lower-dimensional subspaces in each view using multiple views of data via CCA before clustering.
- **Multi-view feature learning (FL)** (Wang et al., 2013): The method projects the concatenated feature vectors to the cluster indicator matrix via a sparse projection matrix W where W is of size $n \times K$ (K total clusters), and imposes structural sparsity on W.
- **Kernel addition and Kernel product:** These two baseline methods were formulated in (Kumar et al., 2011) by summation or component-wise multiplication of two kernel matrices for use in spectral clustering. We used the same procedure as in (Kumar et al., 2011) in our experiments.

5.1. Synthetic data

The synthetic dataset consists of two views and two pairs of row and column clusters. This experiment was designed to mimic the challenging real-life problem in disease classification based on both genetic markers (view 1) and clinical symptoms (view 2). We first created two subject clusters in the genetic view and then used these clusters to create data in the clinical view.

Genetic data was downloaded from the 1000 Genome Project (Abecasis et al., 2012) and 1092 subjects were genotyped with several million genetic markers in this project. We randomly selected 1000 markers from chromosome 5 for use in our experiment. For each subject (row) cluster, 10 markers (columns) were randomly chosen to be associated. To assign subjects to a cluster, we assumed that the minor allele at each locus was the risk variant. We assigned subjects to a cluster if they had over 8 risk variants out of the 10 chosen markers. Subjects that did not belong to any of these simulated clusters were treated as in a third cluster. Hence, 247 and 167 subjects were assigned to subject *cluster 1* and *cluster 2*, respectively and 678 were in the remaining cluster.

To create the subject clusters in the clinical view (view 2), we introduced random noise to clinical features so that the associated clinical features may have different levels of association with the simulated subject clusters. The consistency of the subject clusters between this view and the genetic view varied according to the noise level. We used a parameter e to indicate the noise level. Denote r_i^j the number of risk variants associated with *cluster j* that subject i had, so $0 \leq r_i^j \leq 10$. If $r_i^j * e + N(0,1) > 7.5 * e$, we assigned subject i to *cluster j* in the clinical view. In addition to the two subject clusters that had their counterparts in the genetic view, two additional subject subgroups were created in this view to make the simulated data even more difficult and realistic. The two additional subject clusters each included 200 subjects that were randomly selected.

After the subject clusters were created for the clinical view, we simulated 10 clinical features. A subject was assigned a value of 0 or 1 for each of the features according to a probability. *Cluster 1* and *cluster 2* each was associated with 3 features in the clinical view. Subjects in each simulated subject subgroup obtained the value of 1 with probabilities of 0.6, 0.5, 0.4, respectively for the three designated features. Each of the two additional subgroups in this view was associated with two features, and subjects in each of these two clusters obtained the value of 1 on the respective two features, with probabilities of 0.6 and 0.5, respectively. A subject obtained the value of 1 with a probability of 0.1 on any other features.

To evaluate how the proposed method performs when the noise level varies, we created four datasets in the clinical view (view 2), which were generated with $e = 1, 0.8, 0.6, 0.4$, respectively. Note that when $e = 1$, the two simulated subject clusters are the most consistent across the two views. Decreased e values lead to a higher level of disagreement between the two views.

All of the compared methods were used to obtain three subject clusters, and Table 1 provides the NMI values. The proposed method has the greatest NMI value over all of the four datasets. Along with the decreasing e, NMI values obtained by the proposed approach decrease as expected, but it can still recover the true subject clusters consistent between the two views. All other methods performed poorly with similar NMI values on this difficult dataset. We also experimented with different kinds of kernels (e.g., Gaussian and linear) for those co-clustering methods that are based on a kernel. It may be partially because the kernel matrices were calculated using all features in either of the views. The poor performance may reflect the fact that 98% of the 1000 genetic markers were not relevant to the sub-

Table 1. NMI comparisons between different approaches with different effects e.

	$e = 1.0$	$e = 0.8$	$e = 0.6$	$e = 0.4$
Biclustering via SSVD	0.0821	0.1798	0.2432	0.2286
Co-regularized Spectral	0.2549	0.2477	0.2338	0.2306
Co-training Spectral	0.2062	0.1796	0.2331	0.2378
Multi-view CCA	0.1669	0.1398	0.1097	0.1559
Multi-view FL	0.1569	0.1576	0.1532	0.1211
Kernel addition	0.2587	0.2295	0.2350	0.2566
Kernel product	0.1917	0.2432	0.2302	0.2310
Proposed method	**0.6237**	**0.6226**	**0.6125**	**0.6099**

ject clusters. It may require further investigation about why these methods varied little on their performance when the noise level varied significantly. It is also important to point out that merging the variables from both views into a cluster analysis led to the worst result as shown in the row of Multi-view FL, which as we observed, was because most features were selected only from the genetic view. The genetic view with 1000 features clearly outweighed the clinical view. The resultant clusters differed only on the genetic markers, many of them were not the ones used in synthesizing the subject clusters.

Table 2. The number of features identified by the proposed method as associated with the two simulated subject clusters. TF is the number of True Features that specify a subject cluster. TPF (True Positive Features) and FPF (False Positive Features) are the numbers of features that correctly and incorrectly identified, respectively.

		View 1 (genetic)			View 2 (clinical)		
		TF	TPF	FPF	TF	TPF	FPF
cluster 1	$e = 1$	10	9	1	3	3	0
	$e = 0.8$		9	1		3	0
	$e = 0.6$		9	1		3	0
	$e = 0.4$		10	0		3	0
cluster 2	$e = 1$	10	10	0	3	3	0
	$e = 0.8$		10	0		3	0
	$e = 0.6$		10	0		3	0
	$e = 0.4$		10	0		3	0

As a significant advantage of the proposed method, the features that specify the subject clusters can be simultaneously identified during the clustering process. All other methods could not identify or select among the given variables except Multi-view FL which selected many incorrect features. We calculated the number of variables that were correctly and incorrectly identified by the proposed approach. The results are summarized in Table 2, which shows that our approach correctly identified true associated features in both views (10 in view 1 and 3 in view 2 for each subject cluster) with a very low false discovery rate. The false discovery rate was considered low given there were 1000 genetic variables.

5.2. Real-world benchmark data

Two real-world benchmark datasets with two or more views of data were used in our experiments. We give brief description of each dataset as follows.

- **UCI Handwritten digits dataset:** We downloaded the handwritten digits data from the UCI repository. The dataset consisted of 2000 examples in six views. We used the 76 Fourier coefficients of each image as view 1, and 240 pixel averages in 2×3 image windows as view 2 to report performance. This dataset was previously used to evaluate two recent multi-view clustering methods (Kumar et al., 2011) and (Kumar & Daume III, 2011) which chose two different views of data from ours for evaluation.
- **Crowd-sourcing dataset:** This dataset was downloaded from a study of Crowd-sourcing Big Data (http://web.eecs.umich.edu/∼mozafari/projects.html) (Mozafari et al., 2012). There were 584 images characterized from two views in the dataset. The first view consisted of 15,369 features that were extracted from each of the images. The second view comprised 108 features that were labels given by 27 labelers to each image to indicate the facial expression of the person in the image: neutral, happy, angry or sad.

Table 3. NMI values of different approaches on the two benchmark datasets. Numbers in parentheses are standard deviations.

	Handwritten	Crowd-sourcing
Biclustering via SSVD	0.541 (0.010)	0.409 (0.021)
Co-regularized Spectral	0.823 (0.007)	0.375 (0.024)
Co-training Spectral	0.562 (0.023)	0.133 (0.023)
Multi-view CCA	0.603 (0.106)	0.045 (0.027)
Multi-view FL	0.475 (0.008)	0.373 (0.023)
Kernel addition	0.820 (0.008)	0.274 (0.024)
Kernel product	0.814 (0.014)	0.248 (0.035)
Proposed method	**0.876** (0.006)	**0.428** (0.013)

Table 3 summarizes the NMI values of all methods on the benchmark datasets. On the Handwritten digits data, all multi-view clustering methods, except Multi-view FL, performed better than the single view clustering method. Our approach provided the best performance that was followed closely by the Co-regularized Spectral method. It is interesting to see that the two Kernel baseline methods performed pretty well on this dataset. It is probably because many features from both views were useful for cluster analysis as shown by our method that selected more than half of the features from each view. There were totally 6 views of data in this dataset. We experimented with adding more views and found that the multi-view clustering methods did not always improve performance with more views. When we added in 216 profile coefficients of each image as view 3 for instance, the performance of all methods dropped (e.g.,

our approach reported an NMI of 0.841.) It is likely that some of the views provide noisy or redundant information for the classification of digits.

On the Crowd-sourcing dataset, we observed one of the views (the labeler view) was significantly more useful than the other view (the image-feature view) for correctly grouping the images. The image features in view 1 appeared to be very noisy, so all other methods were unable to improve the NMI performance from the single-view biclustering that just used the labeler labels. The feature concatenation based method, Multi-view FL, performed relatively well because it mostly selected the features from view 2. Our method was the only one that was able to improve performance by utilizing the two views. We also noticed that the co-training based method modified the eigenvector representations of each view based on the clustering results in the other view. Hence, the wrong clustering solution in the image-feature view created undesirable effect on the data representation in the labeler view. Overall, this co-training-based method performed poorly on this dataset.

To tune the hyper-parameters of our approach, we experimented with various methods to determine s_ω, and s_v of each view. Empirically, we observed good performance when the parameter s_v is set to the number of selected features whose percentage of accumulated latent in principle component analysis (PCA) is over 90%. The initial values of vector $\mathbf{v}$ also affects the clustering performance. Although our algorithm is guaranteed to converge to a critical point, different initial point may lead to different local minimizers. We found that if we set the initial vector $\mathbf{v}$ proportionally with the first moment of PCA, our method was able to perform better than those with the initial vectors of all-ones or random-ones. For the chosen parameter values, we ran multiple trials (i.e., each trial used randomly 80% of the data) and reported mean and standard deviation (shown in parentheses) of NMI (Table 3).

6. Related work

We compare our approach with a few most relevant multi-view co-clustering methods. In (Cai et al., 2013), single-view K-means was first represented by a matrix approximation problem, then a binary matrix was used to link different views. This method aligns with our idea but it used a standard alternating optimization and did not aim to select any features. In (Tang et al., 2009; Liu et al., 2013), a linked matrix factorization (LMF) or nonnegative matrix factorization method was proposed to fuse information from multiple graphs. Both methods decompose a data/adjacency matrix into the product of two or more matrices where one of the matrices is shared or comparable across all views. They also require the decomposed matrices to be sparse. The resultant optimization problem requires a gradient descent or quasi-Newton method to optimize at each alternating step. We argue that the requirement of the same representation across all views (simialr to requiring all $\mathbf{u}$'s to be the same in our model) is a over-stringent constraint that limits the search space from the best clustering solution. The time complexity of those methods is high. For instance, the method in (Tang et al., 2009) has complexity of around $O(md(N + nd))$ where N represents the number of nonzero entries averaged over all adjacency matrices. We can clearly see that our approach is significantly faster. (Kumar et al., 2011) imposes regularization conditions to encourage pairwise similarities of subjects under the eigenvector representation to be similar across all the views. Our multi-view extension of low-rank approximation can be extended to eigenvector space/representation in a similar idea to Eq.(2). It would be interesting to further compare these methods to reveal their advantages/disadvantages.

7. Conclusion

In this paper, we have proposed a multi-view sparse clustering approach based on matrix decomposition of multiple data matrices simultaneously into sparse singular vectors. This approach links different views of data by a binary vector that is used to enforce the row clusters from all views to be consistent. Surprisingly, the resultant optimization problem is efficiently solvable using only closed-form formulas in proximal alternating minimization steps. To the best of our knowledge, our work is among the first approaches that extend *sparse* matrix decomposition to multi-view data with rigorous convergence analysis. As matrix decomposition methods are the fundamental tools for many learning tasks, the capability of extending them to learn jointly from multiple views of data will enhance many applications not only in co-clustering. For instance, unsupervised dimension reduction, such as PCA, can directly benefit from the proposed multi-view matrix decomposition approach (e.g., PCA from multi-view SVD).

There are a few directions for future work. It is possible to extend the proposed approach to the case when missing values are present in any of the views. A simple idea is to recover the missing values in one view based on information from other views. Theoretical analysis of co-clustering in general has not been fully explored. Consistency analysis of multi-view SVD-based or low-rank-matrix-approximation-based co-clustering will provide insights into the rate of convergence as the sample size increases. If partial data is labeled, generalization of the proposed framework to the semi-supervised setting will also be important. Although our algorithm is computationally efficient, more empirical evaluations on large-scale datasets might be needed to examine its speed and scalability.

Acknowledgments

This work was supported by NSF grants IIS-1320586, DBI-1356655 and NIH grant R01DA037349. Jinbo Bi was also supported by NSF grants IIS-140720 and IIS-1447711.

References

Abecasis, G. R., Auton, A., Brooks, L. D., and et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, 2012.

Attouch, Hdy, Bolte, Jrme, Redont, Patrick, and Soubeyran, Antoine. Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the kurdyka-lojasiewicz inequality. *Mathematics of operations research*, 35(2):438–457, 2010. URL www.summon.com.

Attouch, Hedy, Bolte, Jrme, and Svaiter, Benar F. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward?backward splitting, and regularized gauss?seidel methods. *Mathematical Programming*, 137(1):91–129, 2013. URL www.summon.com.

Balcan, Maria-Florina, Blum, Avrim, and Yang, Ke. Co-training and expansion: Towards bridging theory and practice. In Saul, Lawrence K., Weiss, Yair, and Bottou, Léon (eds.), *Advances in Neural Information Processing Systems 17*, pp. 89–96. MIT Press, Cambridge, MA, 2005.

Bauschke, H. H. and Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Space*. Springer, New York, 2011.

Blum, Avrim and Mitchell, Tom. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100, New York, NY, USA, 1998. ACM. ISBN 1-58113-057-0. doi: 10.1145/279943.279962.

Bolte, Jrme, Sabach, Shoham, and Teboulle, Marc. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014. URL www.summon.com.

Cai, Xiao, Nie, Feiping, and Huang, Heng. Multi-View K-Means Clustering on Big Data. *International joint conference on Artificial Intelligence*, pp. 2598–2604, 2013. ISSN 10450823.

Chaudhuri, Kamalika, Kakade, Sham M., Livescu, Karen, and Sridharan, Karthik. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 129–136, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1.

Cheng, Wei, Zhang, Xiang, Guo, Zhishan, Wu, Yubao, Sullivan, Patrick F, and Wang, Wei. Flexible and robust co-regularized multi-domain graph clustering. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, 1: 320–328, 2013. doi: 10.1145/2487575.2487582.

Culp, M. and Michailidis, G. A co-training algorithm for multi-view data with applications in data fusion. *J. Chemometr. Journal of Chemometrics*, 23(6):294–303, 2009.

Dhillon, Inderjit S., Mallela, Subramanyam, and Modha, Dharmendra S. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pp. 89–98, New York, NY, USA, 2003. ACM. ISBN 1-58113-737-0.

Gelernter, Joel, Panhuysen, Carolien, Wilcox, Marsha, Hesselbrock, Victor, Rounsaville, Bruce, Poling, James, Weiss, Roger, Sonne, Susan, Zhao, Hongyu, Farrer, Lindsay, and Kranzler, Henry R. Genomewide linkage scan for opioid dependence and related traits. *American journal of human genetics*, 78:759–769, 2006. ISSN 00029297.

Guan, Y., Dy, J., and Jordan, M. A unified probabilistic model for global and local unsupervised feature selection. In *Proceedings of the International Conference on Machine Learning 2011*, pp. 1073–1080, 2011.

Ji, Shuiwang, Zhang, Wenlu, and Liu, Jun. A sparsity-inducing formulation for evolutionary co-clustering. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pp. 334–342, New York, NY, USA, 2012. ACM.

Jiang, Z., Sun, J., Dong, H., Luo, O., Zheng, X., Obergfell, C., Tang, Y., Bi, J., R, O. Neill, Ruan, Y., Chen, J., and Tian, X. C. Transcriptional profiles of bovine in vivo pre-implantation development. *BMC Genomics*, 15(1): 756, 2014.

Kumar, Abhishek and Daume III, Hal. A co-training approach for multi-view spectral clustering. In Getoor, Lise and Scheffer, Tobias (eds.), *Proceedings of the 28th International Conference on Machine Learning*, pp. 393–400, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0619-5.

Kumar, Abhishek, Rai, Piyush, and Daume III, Hal. Co-regularized multi-view spectral clustering. In Shawe-Taylor, J., Zemel, R.S., Bartlett, P., Pereira, F.C.N., and

Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 1413–1421. 2011.

Lee, M., Shen, H., Huang, J. Z., and Marron, J. S. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–95, 2010.

Liu, Jialu, Wang, Chi, Gao, Jing, and Han, Jiawei. Multi-view clustering via joint nonnegative matrix factorization. In *Proc. of SDM*, volume 13, pp. 252–260. SIAM, 2013.

Mozafari, Barzan, Sarkar, Purnamrita, Franklin, Michael J., Jordan, Michael I., and Madden, Samuel. Active learning for crowd-sourced databases. *CoRR*, abs/1209.3686, 2012.

Niu, Donglin, Dy, Jennifer G., and Jordan, Michael I. Multiple Non-Redundant Spectral Clustering Views. In *Proceedings of International Conference on Machine Learning (ICML)*, 2010.

Niu, Donglin, Dy, Jennifer G., and Ghahramani, Zoubin. A nonparametric bayesian model for multiple clustering with overlapping feature views. In *AISTATS*, volume 22 of *JMLR Proceedings*, pp. 814–822. JMLR.org, 2012.

Shan, Hanhuai and Banerjee, Arindam. Bayesian co-clustering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pp. 530–539, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3502-9. doi: 10.1109/ICDM.2008.91. URL http://dx.doi.org/10.1109/ICDM.2008.91.

Sohn, KA and Xing, EP. A hierarchical dirichlet process mixture model for haplotype reconstruction from multi-population data. *Annals of Applied Statistics*, 3(2):791–821, 2009.

Sun, J., Bi, J., and Kranzler, H. R. Multi-view biclustering for genotype-phenotype association studies of complex diseases. *IEEE International Conference on Bioinformatics and Biomedicine (BIBM'13)*, 0:6, 2013.

Sun, J., Bi, J., and Kranzler, H. R. Multi-view singular value decomposition for disease subtyping and genetic associations. *BMC Genet*, 15:73, 2014.

Tang, Wei, Lu, Zhengdong, and Dhillon, Inderjit S. Clustering with multiple graphs. *Data Mining, IEEE International Conference on*, 0:1016–1021, 2009. ISSN 1550-4786.

Van Mechelen, Iven, Bock, Hans-Hermann, and De Boeck, Paul. Two-mode clustering methods: a structured overview. *Statistical methods in medical research*, 13 (5):363–394, 2004.

Wang, Hua, Nie, Feiping, and Huang, Heng. Multi-view clustering and feature learning via structured sparsity. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 28:352–360, 2013.

White, Martha, Yu, Yaoliang, Zhang, Xinhua, and Schuurmans, Dale. Convex multi-view subspace learning. In Bartlett, P., Pereira, F.C.N., Burges, C.J.C., Bottou, L., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1682–1690. 2012.

Yu, S., Krishnapuram, B., Bharat Rao, R., and Rosales, R. Bayesian co-training. *Journal of Machine Learning Research*, 12:2649–2680, 2011.

Multi-view Sparse Co-clustering via Proximal Alternating Linearized Minimization

Appendix A: related definitions

Definition 1 (*Semi-algebraic sets and functions*) A subset S of $\mathbb{R}^d$ is a real semi-algebraic set if there exists a finite number of real polynomial functions g_{ij}, h_{ij}: $\mathbb{R}^d \to \mathbb{R}$ such that

$$S = \bigcup_{j=1}^{p} \bigcap_{i=1}^{q} \{\mathbf{u} \in \mathbb{R}^d : g_{ij}(\mathbf{u}) = 0 \; and \; h_{ij}(\mathbf{u}) < 0\}.$$

Moreover, a function h is called semi-algebraic if its graph

$$\{(u,t) \in \mathbb{R}^{d+1} : h(u) = t\}$$

is a semi-algebraic subset of $\mathbb{R}^{d+1}$.

Semi-algebraic sets are stable under the operations of finite union, finite intersections, complementation and Cartesian product. The following are the semi-functions or the property of semi-functions that are used in the main text:

- Indicator function of semi-algebraic sets.
- Finite sums and product of semi-algebraic functions.
- Composition of semi-algebraic functions.
- Real polynomial functions.

For any subset $S \in \mathbb{R}^d$ and any point $\mathbf{x} \in \mathbb{R}^d$, we define the distance from $\mathbf{x}$ to S as follows:

$$\text{dist}(\mathbf{x}, S) := \inf\{\|\mathbf{y} - \mathbf{x}\| : \mathbf{y} \in S\}.$$

when $S \in \emptyset$, we have $\text{dist}(\mathbf{x}, S) = 0$ fro all $\mathbf{x}$. Let $\eta \in (0, +\infty]$ and Φ_η be the class of all concave and continuous functions $\psi : [0, \eta) \to R_+$ that satisfy the following conditions: (1) $\phi(0) = 0$; (2) ψ is C^1 on $(0, \eta)$ and continuous at 0; (3) for all $s \in (0, \eta) : \psi'(s) > 0$.

Definition 2 (*Kurdyka-Lojasiewicz property*) Let $\sigma : \mathbb{R} \to (-\infty, +\infty]$ be a proper and lower semicontinuous.

(i) The function σ is said to have the *Kurdyka-Lojasiewicz (KL) property* at $\bar{\mathbf{u}} \in \text{dom}\partial\sigma := \{\mathbf{u} \in \mathbb{R}^d : \partial\sigma \neq \emptyset\}$ if there exists $\eta \in (0, +\infty]$, a neighborhood $\mathbf{U}$ of $\bar{\mathbf{u}}$ and a function $\psi \in \Phi_\eta$, such that for all

$$\mathbf{u} \in \mathbf{U} \cap [\sigma(\bar{\mathbf{u}}) < \sigma(\mathbf{u}) < \sigma(\bar{\mathbf{u}}) + \eta],$$

the following equality holds

$$\psi'(\sigma(\mathbf{u}) - \sigma(\bar{\mathbf{u}}))\text{dist}(0, \partial\sigma(\mathbf{u})) \geq 1.$$

(ii) If σ satisfy KL property at each point of dom $\partial\sigma$ then σ is called KL function.