, IEEE Transactions on Knowledge and Data Engineering(Volume:27 , Issue: 5),May 1 2015.

Sparsity Learning Formulations for Mining Time-Varying Data

Rongjian Li, Wenlu Zhang, Yao Zhao, *Senior Member, IEEE*, Zhenfeng Zhu, and Shuiwang Ji, *Member, IEEE*

Abstract—Traditional clustering and feature selection methods consider the data matrix as static. However, the data matrices evolve smoothly over time in many applications. A simple approach to learn from these time-evolving data matrices is to analyze them separately. Such strategy ignores the time-dependent nature of the underlying data. In this paper, we propose two formulations for evolutionary co-clustering and feature selection based on the fused Lasso regularization. The evolutionary co-clustering formulation is able to identify smoothly varying hidden block structures embedded into the matrices along the temporal dimension. Our formulation is very flexible and allows for imposing smoothness constraints over only one dimension of the data matrices. The evolutionary feature selection formulation can uncover shared features in clustering from time-evolving data matrices. We show that the optimization problems involved are non-convex, non-smooth and non-separable. To compute the solutions efficiently, we develop a two-step procedure that optimizes the objective function iteratively. We evaluate the proposed formulations using the Allen Developing Mouse Brain Atlas data. Results show that our formulations consistently outperform prior methods.

Index Terms—Sparsity learning, time-varying data, co-clustering, feature selection, optimization, bioinformatics, neuroinformatics

1 Introduction

Co-clustering aims at identifying block structures of the data matrices by clustering the rows and columns simultaneously into co-clusters [18], [9], [13], [14], [36]. That is, the hidden structure of the data matrix can be more accurately described by a "checkerboard" structure in which a subset of the rows and a subset of the columns form a block. Currently, co-clustering finds applications in many areas, including biological data analysis [29], [23], text mining [14], [13], and social studies [17].

As a class of powerful methods for unsupervised pattern mining, existing co-clustering methods invariably assume that the data matrices are static; that is, they do not evolve over time. However, in many real-world domains, the processes that generated the data are time-evolving. Hence, the observed data are usually dynamic. As a consequence, the block structures embedded into the time-varying data should also evolve smoothly over time. Therefore, it is desirable to incorporate the temporal smoothness constraint into the co-clustering formalism. Similarly, current methods for feature selection in clustering assume that the data are static [47], [50]. Nevertheless, many practical problems are time-evolving, and it is desirable to select features by incorporating the temporal smooth nature of the data.

- *R. Li, W. Zhang, and S. Ji are with the Department of Computer Science, Old Dominion University, Norfolk, VA, USA, 23529. E-mails: {rli, wzhang, sji}@cs.odu.edu*
- *Y. Zhao and Z. Zhu are with the Institute of Information Science, Beijing Jiaotong University, Beijing, China, 100044. E-mails: {yzhao, zhfzhu}@bjtu.edu.cn*

In this paper, we propose an evolutionary co-clustering formulation for identifying co-clusters from time-varying data. The proposed formulation employs sparsity-inducing regularization [38] to identify block structures from the time-varying data matrices. More specifically, it applies fused Lasso type of regularization [39] to encourage temporal smoothness over the block structures identified from contiguous time points. The proposed formulation is very flexible and can be applied to encourage temporal smoothness over either one or both dimensions of the data matrices. We also study the problem of feature selection in clustering on time-varying data. By incorporating the fused Lasso regularization [39] into the framework of sparse feature selection, an evolutionary feature selection formulation is proposed for identifying clusters and shared features in time-varying data simultaneously.

We show that the two proposed formulations for evolutionary co-clustering and feature selection can be reduced to the same optimization problem, which is non-convex, non-smooth, and non-separable. We propose an iterative two-step procedure to compute the solution of the general optimization problem. Each of the iterative step involves a convex, but non-smooth and non-separable problem. To enable efficient optimization, we derive the dual form of this problem and employ a gradient descent algorithm to solve the smooth dual problem.

We evaluate the proposed formulations using the Allen Developing Mouse Brain Atlas data [25], [21], which contain high-resolution, three-dimensional gene expression patterns in the mouse brain at multiple developmental stages. Results show that the proposed evolutionary co-clustering formulation consistently out-

performs other methods by identifying blocks that are consistent with classical neuroanatomy. Meanwhile, the feature selection formulation yields a set of shared features across time points.

The rest of this paper is organized as follows. We introduce the sparse singular value decomposition method for co-clustering, and then describe the proposed evolutionary co-clustering formulation in Section 2. In Section 3, we present the proposed evolutionary feature selection formulation. We discuss some related work in Section 4 and report the experimental results in Section 5. This paper concludes in Section 6 with discussions and future work.

Notations: We use boldface lower-case letters, e.g., $\mathbf{u}$, to denote vectors and upper-case letters, e.g., $\mathbf{X}$, to denote matrices. The norm $||\cdot||$ stands for ℓ_2 norm unless stated otherwise explicitly. For a vector $\mathbf{u}$, its ℓ_1 norm, defined as the summation of the absolute values of its components, is denoted as $||\mathbf{u}||_1$. For a matrix $\mathbf{X}$, its Frobenius norm is denoted as $||\mathbf{X}||_F$. We use $\odot$ to denote the component-wise multiplication and $\otimes$ to denote the Kronecker product. The soft-thresholding operator $\mathcal{T}_\lambda$, acting on a vector $\mathbf{u}$, is defined component-wise as:

$$(\mathcal{T}_\lambda(\mathbf{u}))_i = \begin{cases} u_i - \lambda & \text{if } u_i > \lambda, \\ u_i + \lambda & \text{if } u_i < -\lambda, \\ 0 & \text{if } |u_i| \leq \lambda. \end{cases} \quad (1)$$

2 A Fused Lasso Formulation for Evolutionary Co-Clustering

In this section, we describe the sparse singular value decomposition method for co-clustering. We then propose a fused Lasso formulation for evolutionary co-clustering.

2.1 Sparse Singular Value Decomposition for Co-Clustering

The problem of co-clustering is closely related to the singular value decomposition (SVD) of the data matrices [13], [49], [24]. In [13], [49], the spectral clustering formalism is extended to derive a spectral formulation for co-clustering. In these spectral co-clustering formulations, the data are projected onto the left and the right singular vector spaces before they are concatenated and clustered to identify the co-clusters. Motivated by the relationship between SVD and co-clustering, a sparse SVD formulation is proposed in [24] for co-clustering. Formally, let $\mathbf{X} \in \mathbb{R}^{n\times p}$ be a data matrix. The first singular value and the corresponding left and right singular vectors of $\mathbf{X}$ can be computed as

$$\min_{s,\mathbf{p},\mathbf{q}} \|\mathbf{X} - s\mathbf{p}\mathbf{q}^T\|_F^2,$$

where $s \in \mathbb{R}$ is the first singular value, and $\mathbf{p} \in \mathbb{R}^n$ and $\mathbf{q} \in \mathbb{R}^p$ are the corresponding left and right singular vectors, respectively, and $\|\cdot\|_F$ denotes the matrix Frobenius norm. It is well known that the matrix $s\mathbf{p}\mathbf{q}^T$ is the optimal rank one approximation to the matrix $\mathbf{X}$ [15]. Note that $\mathbf{p}$ and $\mathbf{q}$ lie in the row space and column space, respectively, of $\mathbf{X}$. In addition, the singular vectors $\mathbf{p}$ and $\mathbf{q}$ are usually not sparse; that is, most of their components are nonzero.

Motivated by the optimal rank one approximation property of SVD, a sparse SVD formulation is proposed in [24]. Furthermore, it is shown that this sparse SVD formulation can be employed for solving co-clustering problems. Specifically, the following sparsity-inducing formulation is involved in sparse SVD:

$$\min_{s,\mathbf{p},\mathbf{q}} \frac{1}{2}\|\mathbf{X} - s\mathbf{p}\mathbf{q}^T\|_F^2 + \lambda\|s\mathbf{p}\|_1 + \gamma\|s\mathbf{q}\|_1, \quad (2)$$

where $\|\cdot\|$ denotes the vector ℓ_1-norm, and λ and γ are the regularization parameters. It is well known that the ℓ_1-norm regularization on $\mathbf{p}$ and $\mathbf{q}$ encourages sparse solutions [38]. Thus, when λ and γ are set to large values, many entries of $\mathbf{p}$ and $\mathbf{q}$ will be set of zero. The regularization parameters λ and γ control the tradeoff between the quality of the rank one approximation and the sparsity of $\mathbf{p}$ and $\mathbf{q}$, respectively.

It is shown in [24] that the sparse SVD formulation can be readily employed to solve co-clustering problems. Specifically, the rows and columns of $\mathbf{X}$ corresponding to nonzero entries of $\mathbf{p}$ and $\mathbf{q}$, respectively, can be naturally interpreted to form a co-cluster. If multiple co-clusters are desired, subsequent co-clusters can be identified by removing the rank one approximation from the data matrix and solving the optimization problem in Eq. (2) using the residual matrix. It is shown that this sparse SVD method outperforms prior co-clustering methods by identifying distinctive gene expression profiles corresponding to various pathological conditions from a microarray gene expression data set.

The optimization problem in Eq. (2) is non-convex and non-smooth. An iterative procedure has been developed in [24] to compute the solution. In this procedure, one of the vector variables is fixed while the other one is optimized, and this process is alternated between the two vector variables until it converges to a locally optimal solution. Specifically, when $\mathbf{p}$ is fixed, $\mathbf{q}$ can be computed by solving

$$\min_{\tilde{\mathbf{q}}} F(\tilde{\mathbf{q}}) \equiv \frac{1}{2}\|\mathbf{X} - \mathbf{p}\tilde{\mathbf{q}}^T\|_F^2 + \gamma\|\tilde{\mathbf{q}}\|_1, \quad (3)$$

where $\tilde{\mathbf{q}} = s\mathbf{q}$. After $\tilde{\mathbf{q}}$ is obtained, we have $s = \|\tilde{\mathbf{q}}\|$ and $\mathbf{q} = \tilde{\mathbf{q}}/s$. Similarly, when $\mathbf{q}$ is fixed, the following problem is involved:

$$\min_{\tilde{\mathbf{p}}} G(\tilde{\mathbf{p}}) \equiv \frac{1}{2}\|\mathbf{X} - \tilde{\mathbf{p}}\mathbf{q}^T\|_F^2 + \lambda\|\tilde{\mathbf{p}}\|_1, \quad (4)$$

and $\mathbf{p} = \tilde{\mathbf{p}}/s$ where $s = \|\tilde{\mathbf{p}}\|$. It can be shown that the problems in Eqs. (3) and (4) are convex and can be solved analytically.

The objective function in Eq. (3) can be written as

$$\begin{aligned} F(\tilde{\mathbf{q}}) &= \frac{1}{2}\|\mathbf{X} - \mathbf{p}\tilde{\mathbf{q}}^T\|_F^2 + \gamma\|\tilde{\mathbf{q}}\|_1 \\ &= \frac{1}{2}\operatorname{Tr}(\mathbf{X}^T\mathbf{X}) - \mathbf{p}^T\mathbf{X}\tilde{\mathbf{q}} + \frac{1}{2}\tilde{\mathbf{q}}^T\tilde{\mathbf{q}} + \gamma\|\tilde{\mathbf{q}}\|_1. \end{aligned} \quad (5)$$

Taking the subdifferential of Eq. (5) with respect to $\tilde{\mathbf{q}}$, we have

$$\partial F(\tilde{\mathbf{q}}) = -\mathbf{X}^T\mathbf{p} + \tilde{\mathbf{q}} + \gamma\,\mathrm{SGN}(\tilde{\mathbf{q}}),$$

where $\mathrm{SGN}(\cdot)$ is defined component-wise as

$$(\mathrm{SGN}(\tilde{\mathbf{q}}))_i = \begin{cases} \{1\} & \text{if } (\tilde{\mathbf{q}})_i > 0 \\ \{-1\} & \text{if } (\tilde{\mathbf{q}})_i < 0 \\ [-1,1] & \text{if } (\tilde{\mathbf{q}})_i = 0. \end{cases}$$

Note that the subdifferential of a function is a set, and when the function is differentiable, the set is a singleton containing the derivative [34]. It follows from the optimality condition for unconstrained problems [34] that $\tilde{\mathbf{q}}^*$ is an optimal solution to Eq. (3) if and only if $\mathbf{0} \in \partial F(\tilde{\mathbf{q}}^*)$. Hence, it can be easily verified that the optimal $\tilde{\mathbf{q}}^*$ is given by

$$(\tilde{\mathbf{q}}^*)_i = \begin{cases} (\mathbf{X}^T\mathbf{p} - \gamma)_i & \text{if } (\mathbf{X}^T\mathbf{p})_i > \gamma \\ (\mathbf{X}^T\mathbf{p} + \gamma)_i & \text{if } (\mathbf{X}^T\mathbf{p})_i < -\gamma \\ 0 & \text{if } |(\mathbf{X}^T\mathbf{p})_i| \le \gamma. \end{cases} \quad (6)$$

Similarly, the optimal $\tilde{\mathbf{p}}^*$ for the optimization problem in Eq. (4) is given by

$$(\tilde{\mathbf{p}}^*)_i = \begin{cases} (\mathbf{X}\mathbf{q} - \lambda)_i & \text{if } (\mathbf{X}\mathbf{q})_i > \lambda \\ (\mathbf{X}\mathbf{q} + \lambda)_i & \text{if } (\mathbf{X}\mathbf{q})_i < -\lambda \\ 0 & \text{if } |(\mathbf{X}\mathbf{q})_i| \le \lambda. \end{cases} \quad (7)$$

The iterative procedure in [24] applies Eqs. (6) and (7) alternately until a locally optimal solution is reached.

2.2 Evolutionary Co-Clustering

In the traditional co-clustering framework [18], [9], [13], [23], [36], [29], we assume that the data matrix is time-invariant; that is, it does not evolve along the temporal dimension. In many application domains, each data matrix is usually associated with a particular time point, and it evolves smoothly over time. For example, in the developing mouse brain gene expression analysis, the spatial gene expression patterns at a particular developing time point is captured by a data matrix in which one dimension corresponds to the genes and the other dimension corresponds to the spatial locations. Since gene regulation acts sequentially, the expression patterns usually evolves smoothly over time, thereby resulting a series of time-stamped data matrices, one for each sampled developing time point. A simple approach for mining these time-evolving data matrices is to treat the data matrices at different time points separately. This approach, however, ignores the time-dependent nature of the underlying process, thereby yielding results that are not amenable to domain interpretation. In this paper, we propose an evolutionary co-clustering formulation for uncovering patterns from time-evolving data matrices. The proposed formulation encourages smooth changes in the row and/or column patterns over time, thereby capturing the time-evolving nature of the underlying process faithfully. The proposed framework is very flexible and can be applied to applications in which only one dimension of the data matrices evolves.

Given a set of time-evolving data matrices $\mathbf{X}_t \in \mathbb{R}^{n\times p}$ for $t = 1, \cdots, N$, where N is the number of sampled time points, we are interested in identifying block structures from each of the data matrices. A simple approach is to compute the sparse SVD for each data matrix separately, leading to the following optimization problem:

$$\min_{s_t,\mathbf{u}_t,\mathbf{v}_t} \sum_{t=1}^{N} \left\{ \frac{1}{2}\|\mathbf{X}_t - s_t\mathbf{u}_t\mathbf{v}_t^T\|_F^2 + \lambda\|s_t\mathbf{u}_t\|_1 + \gamma\|s_t\mathbf{v}_t\|_1 \right\},$$

where $\mathbf{u}_t \in \mathbb{R}^n$ and $\mathbf{v}_t \in \mathbb{R}^p$ are associated with the rows and columns, respectively, of $\mathbf{X}_t$, and s_t is the corresponding singular value. However, this approach decouples the data matrices for contiguous time points and ignores the temporal evolving nature of the underlying process that generated the data matrices.

To incorporate the temporal smoothness constraints into the co-clustering framework, we propose the following sparsity-inducing evolutionary co-clustering formulation:

$$\begin{aligned} \min_{s_t,\mathbf{u}_t,\mathbf{v}_t} & \sum_{t=1}^{N} \left\{ \frac{1}{2}\|\mathbf{X}_t - s_t\mathbf{u}_t\mathbf{v}_t^T\|_F^2 + \lambda\|s_t\mathbf{u}_t\|_1 + \gamma\|s_t\mathbf{v}_t\|_1 \right\} \\ & + \sum_{t=1}^{N-1} \left\{ \eta\|s_{t+1}\mathbf{u}_{t+1} - s_t\mathbf{u}_t\|_1 + \xi\|s_{t+1}\mathbf{v}_{t+1} - s_t\mathbf{v}_t\|_1 \right\}, \end{aligned} \quad (8)$$

where η and ξ and tunable parameters. In this formulation, the last two regularization terms are fused Lasso type of regularization [40], and they encourage the $\mathbf{u}_t$ and $\mathbf{v}_t$ for contiguous time points to be similar. Specifically, these regularization terms encourage the differences of contiguous $\mathbf{u}_t$ and $\mathbf{v}_t$ to be zero, thus enforcing many entries of contiguous $\mathbf{u}_t$ and $\mathbf{v}_t$ to be identical. These fused Lasso type of regularization naturally incorporates the time-evolving nature of the data matrices by encouraging the block structures for contiguous time points to be similar. Note that we can also encourage only the rows or the columns of the block structures to be similar by setting either ξ or η to zero.

The objective function in Eq. (8) can be expressed equivalently as

$$\sum_{t=1}^{N} \frac{1}{2}\|\mathbf{X}_t - s_t\mathbf{u}_t\mathbf{v}_t^T\|_F^2 + \lambda\|\tilde{\mathbf{u}}\|_1 + \gamma\|\tilde{\mathbf{v}}\|_1 + \eta\|\mathbf{E}\tilde{\mathbf{u}}\|_1 + \xi\|\mathbf{F}\tilde{\mathbf{v}}\|_1,$$

where $\tilde{\mathbf{u}} = (\mathbf{s}\otimes\mathbf{e}_n)\odot\mathbf{u}$, $\mathbf{s} = [s_1, s_2, \cdots, s_N]^T$, $\tilde{\mathbf{v}} = (\mathbf{s}\otimes\mathbf{e}_p)\odot\mathbf{v}$, $\mathbf{u} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \cdots, \mathbf{u}_N^T]^T \in \mathbb{R}^{nN}$, $\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \cdots, \mathbf{v}_N^T]^T \in \mathbb{R}^{pN}$, $\mathbf{E} \in \mathbb{R}^{n(N-1)\times nN}$ and $\mathbf{F} \in \mathbb{R}^{p(N-1)\times pN}$ are defined as

$$(\mathbf{E})_{ij} = \begin{cases} -1 & \text{if } j = i,\ \ i = 1, \cdots, n(N-1) \\ 1 & j = i + n,\ \ i = 1, \cdots, n(N-1) \\ 0 & \text{otherwise,} \end{cases}$$

$$(\mathbf{F})_{ij} = \begin{cases} -1 & \text{if } j = i,\ \ i = 1, \cdots, p(N-1) \\ 1 & j = i + p,\ \ i = 1, \cdots, p(N-1) \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

The objective function in Eq. (8) is non-convex and non-smooth. In addition, the fused Lasso regularization terms are non-separable [43], [16]. We propose an iterative procedure to compute $\mathbf{u}$ and $\mathbf{v}$. Specifically, we optimize $\mathbf{u}$ by fixing $\mathbf{v}$ and then optimize $\mathbf{v}$ by fixing $\mathbf{u}$. This iterative process is repeated until convergence. In the following, we discuss the detailed procedure of computing $\mathbf{v}$ when $\mathbf{u}$ are fixed. The other case can be derived in a similar way. Specifically, when $\mathbf{u}$ are fixed, $\tilde{\mathbf{v}}$ can be computed by solving the following optimization problem:

$$\min_{\tilde{\mathbf{v}}} f_\xi^\gamma(\tilde{\mathbf{v}}) \equiv \sum_{i=1}^{t} \frac{1}{2} \|A_i - \mathbf{u}_i \tilde{\mathbf{v}}_i^T\|_F^2 + \gamma \|\tilde{\mathbf{v}}\|_1 + \xi \|\mathbf{F}\tilde{\mathbf{v}}\|_1. \quad (10)$$

The objective function in Eq. (10) is convex, but non-smooth and non-separable. In Section 4, we develop an efficient algorithm to compute the optimal $\tilde{\mathbf{v}}^*$.

3 Evolutionary Feature Selection in Clustering

In this section, we describe the problem of feature selection in clustering. We then propose an evolutionary feature selection formulation for clustering time-varying data.

3.1 Feature Selection in Clustering

Given a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ containing n samples and p features, we want to group the rows of $\mathbf{X}$ into clusters. If we use $\mathbf{X}_j$ to denote the j-th column (feature) of $\mathbf{X}$, the objective functions of many clustering methods can be expressed in terms of the columns of $\mathbf{X}$ as follows [47]:

$$\min_{\Theta} \sum_{j=1}^{p} f_j(\mathbf{X}_j, \Theta),$$

where f_j is a function only related to the feature $\mathbf{X}_j$, and Θ represents a partition of the data set. For instance, if we consider the K-means clustering method, f_j will be the summation of within-cluster distances for feature $\mathbf{X}_j$ as

$$f_j(\mathbf{X}_j, \Theta) = c - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j}$$

where $\Theta = \{C_1, \cdots, C_K\}$ is a partition of the samples in K clusters, c is a constant related to the data, n_k denotes the number of samples in cluster C_k, $d_{i,i',j}$ is the dissimilarity between the i-th and the i'-th samples with respect to the j-th feature.

When the squared Euclidean distance $d_{i,i',j} = |\mathbf{X}_{i,j} - \mathbf{X}_{i',j}|^2$ is used, a sparse clustering method was proposed in [47]. Instead of treating features equally, the sparse clustering method gives each feature a weight. This leads to the following optimization problem:

$$\min_{\mathbf{w};\Theta} \sum_{j=1}^{p} w_j f_j(\mathbf{X}_j, \Theta),$$

where $\mathbf{w} = (w_1, \cdots, w_p)^T$ denotes the weight vector for the features. In this formulation, the different contributions of features to the overall objective function are reflected in the weight vector. Furthermore, if additional constraints on the weight vector are imposed, the weight vector can be encouraged to be sparse (i.e., containing zero elements) [38], thereby leading to feature selection [47]. Specifically, the following optimization problem is involved in the sparse clustering method in [47]:

$$\begin{aligned} \min_{\mathbf{w};\Theta} \quad & \sum_{j=1}^{p} w_j f_j(\mathbf{X}_j, \Theta) \\ \text{s.t.} \quad & ||\mathbf{w}||^2 \le 1, \quad ||\mathbf{w}||_1 \le s, \quad w_j \ge 0, \end{aligned}$$

where s is a tuning parameter.

In [47] a two-step procedure is used to solve this optimization problem. In the first step, the weight vector $\mathbf{w}$ is fixed, and the optimization reduces to a weighted K-means problem. In the second step, a vector $\mathbf{a}$ is formed, where each element $a_j = f_j(\mathbf{X}_j, \Theta)$ is the within-cluster distance for the j-th feature based on the clustering results obtained from the first step. This gives rise to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\mathbf{w}^T \mathbf{a} \\ \text{s.t.} \quad & ||\mathbf{w}||^2 \le 1, \quad ||\mathbf{w}||_1 \le s, \quad w_j \ge 0. \end{aligned}$$

This problem is of Lasso type and the solution can be computed by applying the soft thresholding operator in Eq. (1) as

$$\mathbf{w} = \frac{\mathcal{T}_\lambda(\mathbf{a})}{||\mathcal{T}_\lambda(\mathbf{a})||},$$

for some λ determined by s. The weight vector $\mathbf{w}$ is updated and this loop will be iterated until the change of $\mathbf{w}$ is very small.

It is intuitively easy to understand that some features will be given zero weight after a few iterations, and thus they will not affect the clustering results. Specifically, if there are only minor differences among samples for the j-th feature, a_j obtained from the first step will be close to zero. According to the shrinkage effect of the soft thresholding operator, the corresponding weight w_j will be updated to a smaller value. In the next iteration, since the standard K-means will be applied on data scaled by $\mathbf{w}$, the contributions of this feature will diminish gradually. Therefore, features that are invariant among samples will be eliminated eventually.

3.2 Evolutionary Feature Selection in Clustering

In the above feature selection framework, the data matrix is considered static and does not evolve over time. In

many application domains, the data matrices evolve over time, and thus the data matrices at different time points are correlated with each other. Each of them captures a snapshot of an evolving process that generated the data. A simple approach for mining these time-evolving data matrices is to analyze them at different time points separately. In this way, however, the time-dependent nature of the underlying process is ignored and the results are not amenable to domain interpretation.

In this section, we propose an evolutionary feature selection formulation for uncovering shared features from time-evolving data matrices. The proposed formulation encourages smooth changes of the features over time, thereby capturing the time-evolving nature of the underlying process faithfully. Formally, given a sequence of data matrices $\mathbf{X}_t, t = 1, \cdots, N$, where N is the number of time points. A simple idea is to apply sparse K-means separately to each data matrix, leading to the following optimization problem:

$$\begin{aligned} \min_{\tilde{\mathbf{w}};\tilde{\Theta}} \quad & \sum_{t=1}^{N}\sum_{j=1}^{p}(\mathbf{w}_t)_j f_j((\mathbf{X}_t)_j, \Theta_t) \\ & ||\mathbf{w}_t||^2 \leq 1, \quad ||\mathbf{w}_t||_1 \leq s_t, \quad t = 1, \cdots, N, \end{aligned}$$

where $\tilde{\mathbf{w}} = (\mathbf{w}_1^T, \cdots, \mathbf{w}_N^T)^T \in \mathbb{R}^{Np}$, $\mathbf{w}_t \in \mathbb{R}^p$ is weight vector corresponding to the data matrix $\mathbf{X}_t \in \mathbb{R}^{n\times p}$, $\tilde{\Theta} = \{\Theta_1, \cdots, \Theta_N\}$, and $\{s_1, \cdots, s_N\}$ are the tuning parameters controlling the feature selection at different time points.

In order to encourage the selection of shared features among time-varying data matrices, we introduce a fused Lasso term on the successive differences of the weight vectors. This leads to the following optimization problem:

$$\begin{aligned} \min_{\tilde{\mathbf{w}};\tilde{\Theta}} \quad & \sum_{t=1}^{N}\sum_{j=1}^{p}(\mathbf{w}_t)_j f_j((\mathbf{X}_t)_j, \Theta_t) \\ & ||\mathbf{w}_t||^2 \leq 1, \quad ||\mathbf{w}_t||_1 \leq s_t, \quad t = 1\cdots, N, \\ & ||\mathbf{w}_t - \mathbf{w}_{t-1}||_1 \leq s', \quad t = 2, \cdots, N, \end{aligned} \tag{11}$$

where s' is a tuning parameter to encourage weight vectors at contiguous time points to be similar. Specifically, with the fused Lasso regularization, $\mathbf{w}_t - \mathbf{w}_{t-1}$ will be enforced to be close to zero if the tuning parameter s' is small enough. In this case, $\mathbf{w}_t$ and $\mathbf{w}_{t-1}$ will be almost the same, thereby leading to the selection of shared features across time points.

Following [47], we develop a two-step procedure for solving the optimization problem in Eq. (11). In the first step, the weight vector $\tilde{\mathbf{w}}$ is fixed, and we optimize the clustering $\tilde{\Theta}$. This leads to a set of decoupled clustering problems in which each feature is associated with a weight. The can be solved by applying commonly used algorithms such as K-means to scaled data matrices using $\tilde{\mathbf{w}}$ as the weights. In the second step, the clustering results $\tilde{\Theta}$ from the first step are fixed, and the optimization problem in Eq. (11) is reduced to a fused Lasso type problem as

$$\begin{aligned} \min_{\tilde{\mathbf{w}}} \quad & -\tilde{\mathbf{w}}^T\tilde{\mathbf{a}} \\ & ||\mathbf{w}_t||^2 \leq 1, \quad ||\mathbf{w}_t||_1 \leq s_t, \quad t = 1\cdots, N, \\ & ||\mathbf{w}_t - \mathbf{w}_{t-1}||_1 \leq s', \quad t = 2, \cdots, N, \end{aligned} \tag{12}$$

where $\tilde{\mathbf{a}} = (\mathbf{a}_1^T, \cdots, \mathbf{a}_N^T)^T$, $\mathbf{a}_t \in \mathbb{R}^p$ is the within-cluster dissimilarity vector, and its element is defined as $(\mathbf{a}_t)_j = f_j((\mathbf{X}_t)_j, \Theta_t)$.

We can transform the problem in Eq. (12) to an equivalent unconstrained optimization problem:

$$\min_{\tilde{\mathbf{w}}} \quad \alpha||\tilde{\mathbf{w}}||^2 - \tilde{\mathbf{w}}^T\tilde{\mathbf{a}} + \sum_{t=1}^{N}\lambda_t||\tilde{\mathbf{w}}_t||_1 + \lambda'||\mathbf{F}\tilde{\mathbf{w}}||_1, \tag{13}$$

where $\mathbf{F}$ is defined in Eq.(9), the coefficient of the ℓ_2-norm term α depends on the data and needs to be determined. Although there is no closed form solution for α, we can devise an approximation scheme to estimate its value. To this end, we propose to solve an unconstraint optimization problem first as

$$\min_{\tilde{\mathbf{w}}} \quad \alpha||\tilde{\mathbf{w}}||^2 - \tilde{\mathbf{w}}^T\tilde{\mathbf{a}}. \tag{14}$$

The solution to the problem in Eq. (14) can be expressed as $\tilde{\mathbf{w}}^\star = \frac{\tilde{\mathbf{a}}}{2\alpha}$. Then we set $||\tilde{\mathbf{w}}^\star|| = 1$, and obtain $\alpha^\star = \frac{||\tilde{\mathbf{a}}||}{2}$. Finally, we substitute this $\alpha^\star$ into Eq. (13) and get the following problem:

$$\min_{\tilde{\mathbf{w}}} \frac{1}{2}||\tilde{\mathbf{w}} - \tilde{\mathbf{u}}||^2 + \sum_{t=1}^{N}\lambda_t||\mathbf{w}_t||_1 + \lambda'||\mathbf{F}\tilde{\mathbf{w}}||_1, \tag{15}$$

where $\tilde{\mathbf{u}} = \frac{\tilde{\mathbf{a}}}{||\tilde{\mathbf{a}}||}$. The formulation in Eq. (15) is similar to the problem that we need to solve in the second step of the evolutionary co-clustering procedure. In Eq. (15), we still use the notations λ_t and λ' instead of their exact forms, $\frac{2\lambda_t}{||\tilde{\mathbf{a}}||}$ and $\frac{2\lambda'}{||\tilde{\mathbf{a}}||}$ to simplify the notation, since the parameters λ_t and λ' can be scaled to make these two forms equivalent. Note that the parameters λ_t can be different for the sequences of data sets. For simplicity, we set them to the same value in our experiments.

4 An Efficient Algorithm

The evolutionary co-clustering is for identifying the hidden block structures in the data matrices along the temporal dimension. Meanwhile, the evolutionary feature selection method is designed to uncover the shared features from time-evolving data matrices. We show that both problems can be formulated as solving fused Lasso regularized objective functions. Specifically, a common optimization problem that needs to be solved in the evolutionary co-clustering and feature selection formulations in Sections 2 and 3 has the following form:

$$\min_{\tilde{\mathbf{w}}} f_{\lambda_2}^{\lambda_1} \equiv L(\tilde{\mathbf{w}}) + \lambda_1||\tilde{\mathbf{w}}||_1 + \lambda_2||\mathbf{F}\tilde{\mathbf{w}}||_1, \tag{16}$$

where $L(\tilde{\mathbf{w}})$ is a convex smooth loss function. In particular, $L(\tilde{\mathbf{w}})$ is $\sum_{t=1}^{N}\frac{1}{2}\|\mathbf{X}_t - \mathbf{u}_t\tilde{\mathbf{w}}_t^T\|_F^2$ for the evolutionary co-clustering model and $\frac{1}{2}||\tilde{\mathbf{w}} - \tilde{\mathbf{u}}||^2$ for the evolutionary

feature selection formulation. This optimization problem is similar to the fused Lasso signal approximator [28], [27], and we develop an efficient procedure for solving it in the following.

4.1 A Two-Step Algorithm

A central challenge for solving the optimization problem in Eq. (16) is to deal with the ℓ_1-norm and the fused Lasso regularization term, which is non-smooth and non-separable. A key property that leads to an efficient algorithm to this problem is that the ℓ_1-norm term and the fused Lasso term can be solved sequentially in two steps, giving rise to a two-step procedure. This result is originally given in [16], [28] and is summarized in the following theorem:

Theorem 4.1: Define

$$\pi_{\lambda_2}^{\lambda_1} = \arg\min_{\tilde{\mathbf{w}}} f_{\lambda_2}^{\lambda_1}(\tilde{\mathbf{w}}). \tag{17}$$

Then for any $\lambda_1, \lambda_2 \geq 0$, we have

$$\pi_{\lambda_2}^{\lambda_1} = \mathcal{T}_{\lambda_1}\left(\pi_{\lambda_2}^{0}\right). \tag{18}$$

The proof of this theorem is similar to that of Theorem 3 in [28] and is thus omitted.

Theorem 4.1 shows that we can solve the optimization problem in two sequential steps. Specifically, we can first solve the problem in Eq. (16) with $\lambda_1 = 0$ to obtain the intermediate solution $\pi_{\lambda_2}^{0}$. Then the final optimal solution $\pi_{\lambda_2}^{\lambda_1}$ can be obtained by applying the soft thresholding operator to the intermediate solution as in Eq. (18). We now discuss how the $\lambda_1 = 0$ case can be solved efficiently in its dual form.

4.2 A Dual Formulation

A key to the two-step procedure mentioned above is to solve the optimization problem rewritten in its full form as

$$\min_{\tilde{\mathbf{w}}} f_{\lambda_2}(\tilde{\mathbf{w}}) \equiv L(\mathbf{w}) + \lambda_2 \|\mathbf{F}\tilde{\mathbf{w}}\|_1. \tag{19}$$

We propose to solve this problem in its dual form. Since the ℓ_1 norm is non-differentiable, we obtain the following equivalent min-max problem:

$$\min_{\tilde{\mathbf{w}}} \max_{\|\tilde{\mathbf{z}}\|_\infty \leq \lambda_2} \phi(\tilde{\mathbf{w}}, \tilde{\mathbf{z}}) \equiv L(\tilde{\mathbf{z}}) + \langle \mathbf{F}\tilde{\mathbf{w}}, \tilde{\mathbf{z}} \rangle. \tag{20}$$

The existence of saddle point to this min-max problem is guaranteed by the Von Neumann Lemma [33], because $\phi(\cdot,\cdot)$ is differentiable, convex in $\tilde{\mathbf{w}}$, and concave in $\tilde{\mathbf{z}}$. After exchanging the order of min and max and setting the derivative of $\phi(\tilde{\mathbf{w}}, \tilde{\mathbf{z}})$ with respect to $\tilde{\mathbf{w}}$ to zero, we obtain an equation to describe the relationship between the primal and dual variables as

$$\nabla L(\tilde{\mathbf{w}}) + \mathbf{F}^T\tilde{\mathbf{z}} = 0, \tag{21}$$

where $\nabla L(\tilde{\mathbf{w}})$ denotes the gradient of the smooth function $L(\tilde{\mathbf{w}})$ with respect to $\tilde{\mathbf{w}}$. By substituting Eq. (21) into Eq. (20), we obtain a dual optimization problem in terms of $\tilde{\mathbf{z}}$. For ease of presentation, we change max to min after the substitution by negating the objective function.

In the case of evolutionary co-clustering, the form of the dual problem can be written as

$$\min_{\|\tilde{\mathbf{z}}\|_\infty \leq \lambda_2} \varphi(\tilde{\mathbf{z}}) \equiv \frac{1}{2}\|\mathbf{F}^T\tilde{\mathbf{z}}\|^2 - \left\langle \tilde{\mathbf{X}}^T\mathbf{u}, \mathbf{F}^T\tilde{\mathbf{z}} \right\rangle - c, \tag{22}$$

where

$$\tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{X}_1 & & & 0 \\ & \mathbf{X}_2 & & \\ & & \ddots & \\ 0 & & & \mathbf{X}_t \end{pmatrix} \in \mathbb{R}^{nN \times pN},$$

$c = \frac{1}{2}\sum_{t=1}^{N} \mathrm{Tr}\left((\mathbf{X}_t - \mathbf{u}_t\mathbf{u}_t^T\mathbf{X}_t)(\mathbf{X}_t - \mathbf{u}_t\mathbf{u}_t^T\mathbf{X}_t)^T\right)$. In the case of evolutionary feature selection, the dual problem can be written as

$$\min_{||\tilde{\mathbf{z}}||_\infty \leq \lambda_2} \varphi(\tilde{\mathbf{z}}) \equiv \frac{1}{2}||\mathbf{F}^T\tilde{\mathbf{z}}||^2 - < \mathbf{F}\tilde{\mathbf{z}}, \tilde{\mathbf{u}} > . \tag{23}$$

The dual formulations in Eqs. (23) and (22) are convex and smooth. Hence, they can be solved by gradient decent algorithms.

4.3 A Gradient Algorithm

The dual problems in Eqs. (22) and (23) are constrained quadratic programs (QP) and can be solved by general QP solvers. However, direct application of general QP solvers would ignore the special structure of this problem, incurring excessive computational cost. In this paper, we propose to solve this dual formulation by a gradient descent algorithm, since the objective function is differentiable. Note that the Hessian of $\varphi(\tilde{\mathbf{z}})$ in Eqs. (22) and (23) is a $p(N-1) \times p(N-1)$ matrix and can be express as

$$\mathbf{F}\mathbf{F}^T = \begin{pmatrix} 2 & \overbrace{\cdots}^{(p-1)\,0s} & -1 & \cdots & \cdots & 0 \\ \vdots & 2 & \cdots & -1 & \cdots & 0 \\ -1 & \vdots & \ddots & \cdots & \ddots & \vdots \\ \vdots & -1 & \vdots & \ddots & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & \cdots & 2 \end{pmatrix}.$$

Since the Hessian matrix is positive definite, the following iterative process is guaranteed to converge to the solution:

$$\tilde{\mathbf{z}}^{(k+1)} = P_{||\cdot||\leq\lambda_2}\left(\tilde{\mathbf{z}}^{(k)} - \frac{1}{\mathrm{eign}_{\max}}\tilde{\mathbf{g}}^{(k)}\right),$$

where $\tilde{\mathbf{g}}^{(k)} = \nabla\varphi(\tilde{\mathbf{z}}^{(k)})$, $\mathrm{eign}_{\max}$ is the largest eigenvalue of the Hessian matrix and

$$\left(P_{||\cdot||\leq\lambda_2}(\mathbf{x})\right)_i = \begin{cases} x_i & \text{if } |x_i| \leq \lambda_2, \\ \mathrm{sgn}(x_i)\lambda_2 & \text{if } |x_i| > \lambda_2. \end{cases}$$

From the analysis in [34], this algorithm has a linear convergence rate as

$$||\tilde{\mathbf{z}}^{(k)} - \tilde{\mathbf{z}}^{\star}||^2 \leq \left(1 - \frac{\text{eign}_{\min}}{\text{eign}_{\max}}\right)^k ||\tilde{\mathbf{z}}^{(0)} - \tilde{\mathbf{z}}^{\star}||^2,$$

where $\tilde{\mathbf{z}}^{\star}$ denotes the optimal solution, and $\tilde{\mathbf{z}}^{(0)}$ is the starting point of this iterative process. Since $\mathbf{F}$ is a full rank matrix, the Hessian matrix $\mathbf{F}\mathbf{F}^T$ is positive definite. Thus a unique solution exists. This algorithm can also be accelerated by the Nesterov's method [34].

4.4 Convergence and Stopping Criterion

The gradient descent algorithm is an iterative procedure, and thus a criterion is required to assess the convergence of the algorithm. Following [28], we define a duality gap for the min-max problem in Eq. (20) and derive a simple equation for computing the duality gap in each iteration. We use this duality gap as the stopping criterion in our experiments, and the gradient descent algorithm returns when the duality gap is smaller than 10^{-8}.

Let $\bar{\mathbf{z}}$ be an appropriate solution computed by the gradient descent algorithm. Note that $\|\bar{\mathbf{z}}\|_\infty \leq \lambda_2$, as it has been projected onto the feasible region in each step. Let $\bar{\mathbf{w}}$ be the corresponding solution for the primal formulation. We can define the duality gap for Eq. (20) at $(\bar{\mathbf{w}}, \bar{\mathbf{z}})$ as

$$\text{dg}(\bar{\mathbf{w}}, \bar{\mathbf{z}}) = \max_{\|\tilde{\mathbf{z}}\|_\infty \leq \lambda_2} \phi(\bar{\mathbf{w}}, \tilde{\mathbf{z}}) - \min_{\tilde{\mathbf{w}}} \phi(\tilde{\mathbf{w}}, \bar{\mathbf{z}}). \quad (24)$$

The following results show that the duality gap in Eq. (24) is an upper bound for the errors in both the primal and the dual formulations. In addition, it can be computed easily by a simple equation.

Theorem 4.2: The duality gap defined in Eq. (24) can be computed as

$$\text{dg}(\bar{\mathbf{w}}, \bar{\mathbf{z}}) = \lambda_2 \|\nabla\varphi(\bar{\mathbf{z}})\|_1 + \langle \bar{\mathbf{w}}, \nabla\varphi(\bar{\mathbf{z}})\rangle . \quad (25)$$

In addition, we have the following results:

$$\varphi(\bar{\mathbf{z}}) - \varphi(\tilde{\mathbf{z}}^*) \leq \text{dg}(\bar{\mathbf{w}}, \bar{\mathbf{z}}), \quad (26)$$

$$f_{\lambda_2}(\bar{\mathbf{w}}) - f_{\lambda_2}(\tilde{\mathbf{w}}^*) \leq \text{dg}(\bar{\mathbf{w}}, \bar{\mathbf{z}}). \quad (27)$$

The proof of this theorem is similar to that of Theorem 3 in [28] and is thus omitted.

4.5 Regularization Parameter Interval

The regularization parameter λ_2 controls the temporal smoothness over $\mathbf{w}_i$. That is, when λ_2 is larger than a certain value $\lambda_{\max}$, $\mathbf{w}_t$ and $\mathbf{w}_{t+1}$, for $t = 1, 2, \cdots, N-1$, will be enforced to be identical. We show that such a $\lambda_{\max}$ can be computed via solving a system of equations. To this end, we need to state the optimality condition for the problems in Eqs. (22) and (23).

It follows from the optimality condition for constrained problems [34] that $\tilde{\mathbf{z}}^*$ ($\|\tilde{\mathbf{z}}^*\|_\infty \leq \lambda_2$) is a minimizer of Eq. (23) or (22) if and only if

$$\langle \nabla\varphi(\tilde{\mathbf{z}}^*), \tilde{\mathbf{z}} - \tilde{\mathbf{z}}^*\rangle \geq 0, \ \ \forall \tilde{\mathbf{z}} : \|\tilde{\mathbf{z}}\|_\infty \leq \lambda_2.$$

This is the well-known variational inequality, and it gives the optimality condition for constrained optimization problems. Based on the above result, we show that $\lambda_{\max}$ can be computed via solving a system of equations with a special structure.

Theorem 4.3: Let $\hat{\mathbf{z}}$ denote the unique solution of the system

$$\nabla\varphi(\tilde{\mathbf{z}}) = 0,$$

and let

$$\lambda_{\max} = \|\hat{\mathbf{z}}\|_\infty.$$

Then for any $\lambda_2 \geq \lambda_{\max}$, we have $\tilde{\mathbf{w}}_i = \tilde{\mathbf{w}}_j$, $\forall\, i, j$.
The proof of this theorem is similar to that of Theorem 3.3 in [22] and is thus omitted.

The value of $\lambda_{\max}$ can be used to guide the selection of an appropriate value for λ_2 in practice. We evaluate the effectiveness of λ_2 in the experiments on the biological data sets.

5 Related Work

Simultaneous row and column clustering for identifying block structures from matrix data has been initially studied in [18]. Recent surge of interests in co-clustering is motivated by biological applications, which aim at identifying subset of genes co-expressed in a subset of samples from microarray gene expression data [9]. Co-clustering has also been applied in many other applications, including simultaneous clustering of words and documents [14], [13], authors and conference [42], etc. Early work on co-clustering focuses on defining an error measure and then identifying blocks that minimize this measure using heuristic search algorithms [18], [9]. These early work has recently been reformulated using matrix and optimization techniques [11], [4]. Following the spectral clustering formalism, it has been shown recently that co-clustering is closely related to the singular value decomposition (SVD) of the data matrix [6]. In [13], [49], co-clustering is formulated as a bipartite graph cut problem, and the data are projected onto the left and right singular vector spaces before they are concatenated and clustered to identify row and column co-clusters. It is shown in [24] that sparsity-inducing regularization can be employed to compute sparse singular vectors, which in turn can be used to form co-clusters. In [12], a framework for simultaneous co-clustering and predictive learning is proposed.

This work is also related to recent studies on mining from time-evolving data. Chakrabarti *et al.* [7] first proposed the concept of evolutionary clustering and extended the K-means and the hierarchical clustering algorithms for uncovering smooth patterns from time-evolving data matrices. In [10], the spectral clustering formalism is systematically extended to the evolutionary setting by incorporating a temporal cost into the objective function, leading to a suite of formulations for evolutionary spectral clustering. In [26], the nonnegative matrix factorization is employed for soft clustering, and a

temporal cost is included for mining from time-evolving data. Evolutionary nonnegative matrix factorization is studied in [44], and the idea of adaptively estimating the smoothness parameter is proposed in [48]. The broad area of evolutionary network analysis is reviewed in [1].

The fused Lasso penalty was originally proposed in [40] for encouraging smoothness over related coefficients in regression problems. This type of penalty is very attractive and has been applied for encouraging smoothness over spatial and temporal smoothness in many applications, including biological data analysis [41] and social studies. A critical challenge in employing the fused Lasso formalism is that this class of penalty is non-smooth and non-separable and thus is very challenging to optimize. In [16], a modified coordinate descent algorithm is developed to solve the fused Lasso formulation. However, this algorithm is not guaranteed to give the exact solution. In [19], a path algorithm is proposed to solve the fused Lasso signal approximator. Instead of solving the original primal problem, Liu *et al.* developed a dual formulation for the fused Lasso signal approximator and devised a gradient descent algorithm for computing the dual solution [28]. Similar formulations and algorithms have been studied in the compressive sensing literature [20], [8].

The problem of feature selection in clustering has been studied in [47], [50], [2], [45], [31]. These studies mostly focus on clustering static data matrices. In the literature, the evolutionary clustering [7], [10], [26] paradigm is related, but different from, the currently studied evolutionary feature selection formalism. Specifically, the smoothness constraints are imposed on the sample dimension in evolutionary clustering, while similar constraints are imposed on the feature dimension in evolutionary feature selection. Consequently, the clustering results are expected to evolve smoothly in evolutionary clustering, while the selected features are shared across time points in evolutionary feature selection.

6 Experimental Evaluation

6.1 Experimental Setup

We evaluate the proposed evolutionary co-clustering formulation and evolutionary feature selection formulation using the Allen Developing Mouse Brain Atlas data [3], [37]. This data set contains *in situ* hybridization gene expression pattern images in the developing mouse brain across seven developmental ages E11.5, E13.5, E15.5, E18.5, P4, P14, and P28. The 3D images are registered to a reference atlas separately for each age, and a regular grid is applied to partition the 3D brain space into voxels. The expression energy within each voxel is given as a numerical value. There is one data matrix associated with each of the seven developing ages. The rows of the matrices correspond to brain voxels while the columns correspond to genes. The reference atlas ontology is organized into a hierarchy, and we up-propagate the annotations to Level 3 and Level 5 in the experiments. It is well-known that the developing mouse brain is divided into grid-like patterns along the longitudinal and transversal dimensions [46], [35], and identification of genes co-expressed in these domains might elucidate the genetic mechanisms governing the mouse brain development. The transversal and longitudinal dimensions correspond to the Level 3 and Level 5 ontology, respectively. Table 1 shows the statistics on the number of genes, voxels and brain regions for each data set on Level 3 and Level 5 annotations respectively.

To measure the performance of our proposed methods, we consider the annotated brain region of each voxel as its class and compare the clustering results with the region labels of voxels, since it has been shown that the results of gene expression data clustering are largely consistent with classical neuroanatomy [5]. Following [30], the normalized mutual information (NMI) and Rand index are used to quantitatively measure the correspondence of the clustering results with the classical neuroanatomy reflected in the region annotations. We use the duality gap as the stopping criterion for the gradient descent algorithm and the error tolerance is set to 10^{-8} in the experiments. Overall, the proposed formulations are efficient to solve on a regular desktop PC, but we do not provide detailed timing results due to space constraints.

6.2 Co-Clustering Performance Evaluation

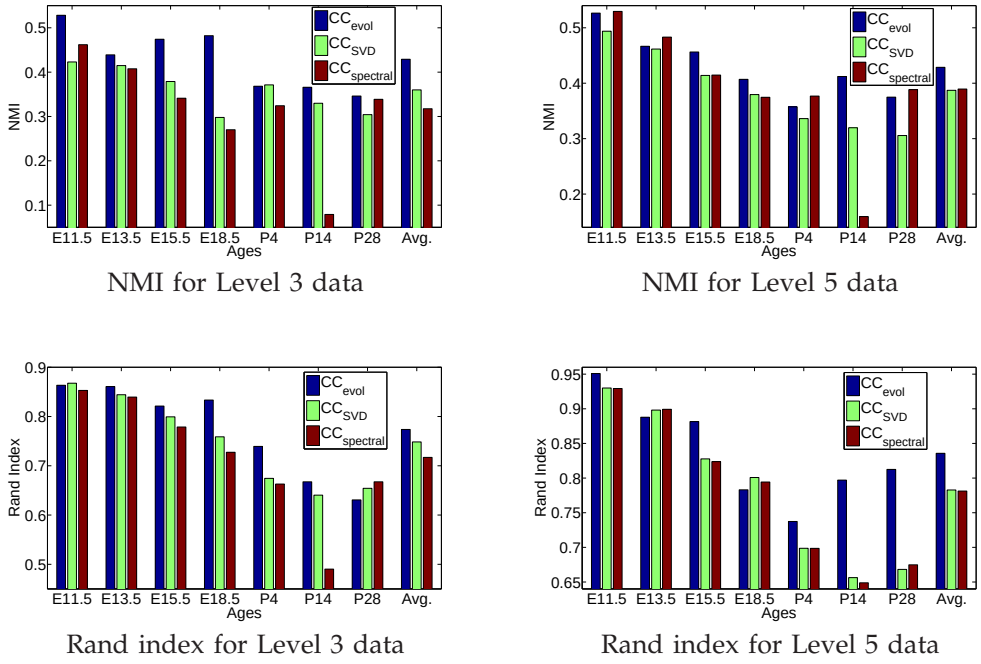
To evaluate the performance of the proposed evolutionary co-clustering method, we compare the proposed method with two other co-clustering methods; namely the one based on sparse SVD in [24] and the spectral co-clustering method proposed in [13], [49]. Note that the evolutionary clustering methods [7], [10], [26] cannot be applied to this data set, since the brain voxels are not registered across ages and the data for each age contain different number of voxels. Hence, we only apply the fused Lasso regularization over the columns (genes); that is, we set $\eta = 0$ in Eq. (8). This is one of the unique advantages of the proposed formulation in which the smoothness constraint can be applied to either or both dimensions.

The performance of the three methods on the seven data sets is reported in Figure 1. We observe that the best performance is achieved when $\xi = 0.05 \times \lambda_{\max}$ where $\lambda_{\max}$ is defined in Eq. (4.3) and report the results under this parameter setting. Detailed studies on parameter sensitivity are reported in the following. It can be observed from Figure 1 that incorporation of the smoothness constraints between contiguous age data yield improved performance.

In order to fully understand how the fused Lasso regularization parameter affects performance, we conduct a series of experiments and report the results in the following. We first investigate how the performance changes as the value for ξ changes. To this end, we vary

TABLE 1
Statistics about the mouse brain data at annotation Level 3 and Level 5.

		E11.5	E13.5	E15.5	E18.5	P4	P14	P28
Number of genes	Level 3	1724	1724	1724	1724	1724	1724	1724
	Level 5	1724	1724	1724	1724	1724	1724	1724
Number of voxels	Level 3	7122	13194	12148	12045	21845	24180	28023
	Level 5	7106	13191	12148	12045	21845	24180	28023
Number of regions	Level 3	20	20	20	20	20	19	20
	Level 5	82	77	76	65	64	71	74



NMI for Level 3 data | NMI for Level 5 data

Rand index for Level 3 data | Rand index for Level 5 data

Fig. 1. Performance of the proposed method ($\xi = 0.05 \times \lambda_{\max}$), denoted as CC_{evol}, for the Level 3 data and Level 5 data in comparison with two other methods measured using NMI and Rand index respectively. CC_{SVD} denotes the co-clustering method based on SVD proposed in [24]; $CC_{spectral}$ denotes the spectral co-clustering method proposed in [13].

the value for ξ from $0.001 * \lambda_{\max}$ to $\lambda_{\max}$ and report the performance on each data set and summarize the average performance across data sets in Tables 2 and 4 for Level 3 data sets and Tables 3 and 5 for Level 5 data sets, respectively. We can observe that the performance is dependent on the choice of the parameter value. This demonstrate that incorporation of the fused Lasso regularization is effective in boosting the performance.

To evaluate the effectiveness of the fused Lasso regularization in encouraging smoothness over the temporal dimension, we report the ℓ_1-norm differences between temporally adjacent variable vectors with different values of ξ in Figure 2. We can observe that, as ξ increases, the values for the fused Lasso regularization terms decrease monotonically for Level 3 data until they reach zero, where the adjacent variables are forced to be identical. The values for Level 5 data also decreased to zero with the increasing of ξ after some fluctuations when ξ is very small.

We also evaluate the effectiveness of the defined duality gap in determining the convergence of the gradient descent algorithm. To this end, we plot the values of the duality gap in the first 50 iterations of the gradient descent algorithm under multiple ξ values in Figure 3. We can observe that the duality gap decreases monotonically in all cases. In addition, as the value of ξ increases, the duality gap approaches zero at a slower speed. This is because more computations are required to fuse adjacent variables when the value for ξ increases. In all cases, the duality gap is reduced below the tolerance level within a relatively small number of iterations.

6.3 Evolutionary Feature Selection in Clustering

To evaluate the proposed evolutionary feature selection formulation, we compare it with two other clustering methods; namely the K-means and the sparse K-means methods in [47] on the Allen Developing Mouse Brain Atlas data. To study the effect of the fused Lasso regularization parameter, we conduct a series of experiments and report the performance measured using NMI and

TABLE 2
Performance of the proposed method on the the Level 3 Allen Developing Mouse Brain Atlas data measured using NMI. The regularization parameter is set to $\xi =$ percentage $\times \lambda_{\max}$, and the "percentage" is increased from 0.001 to 1. CC_{SVD} denotes the co-clustering method based on SVD proposed in [24]; $CC_{spectral}$ denotes the spectral co-clustering method proposed in [13].

Data	CC_{SVD}	$CC_{spectral}$	0.001	0.005	0.01	0.05	0.1	0.5	1
E11.5	0.4230	0.4619	0.4983	0.5108	0.4947	**0.5282**	0.4835	0.2844	0.2637
E13.5	0.4148	0.4076	**0.4694**	0.4438	0.4672	0.4390	0.4463	0.3701	0.3557
E15.5	0.3789	0.3412	0.4770	0.4609	**0.4795**	0.4742	0.4218	0.3890	0.3878
E18.5	0.2978	0.2701	0.4498	0.4394	0.4435	**0.4822**	0.3816	0.3665	0.3952
P4	0.3713	0.3243	0.3087	0.3345	0.3902	0.3682	0.3353	0.3812	**0.4275**
P14	0.3298	0.0791	0.4186	0.3904	0.3607	0.3659	0.3490	0.3422	**0.4259**
P28	0.3042	0.3387	0.3521	0.3487	0.3382	0.3461	0.3087	0.3204	**0.4147**
Avg.	0.3600	0.3175	0.4248	0.4184	0.4249	**0.4291**	0.3894	0.3505	0.3815

TABLE 3
Performance of the proposed method on the the Level 5 Allen Developing Mouse Brain Atlas data measured using NMI. The regularization parameter is set to $\xi =$ percentage $\times \lambda_{\max}$, and the "percentage" is increased from 0.001 to 1. CC_{SVD} denotes the co-clustering method based on SVD proposed in [24]; $CC_{spectral}$ denotes the spectral co-clustering method proposed in [13].

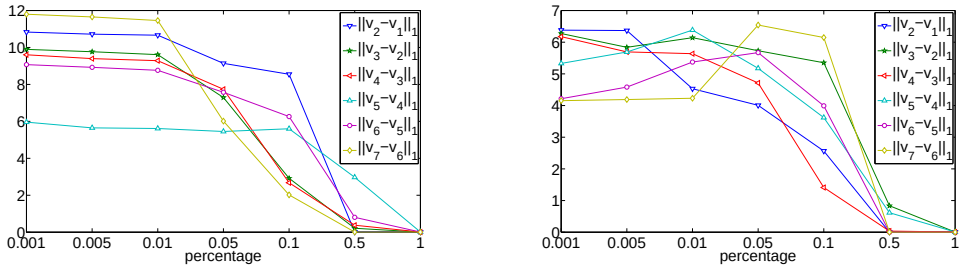
Data	CC_{SVD}	$CC_{spectral}$	0.001	0.005	0.01	0.05	0.1	0.5	1
E11.5	0.4939	0.5296	0.5153	0.5205	**0.5358**	0.5265	0.4921	0.4408	0.4403
E13.5	0.4616	0.4832	**0.5192**	0.5063	0.4836	0.4667	0.4459	0.4487	0.4385
E15.5	0.4140	0.4147	0.4477	0.4336	0.4556	**0.4563**	0.4360	0.4491	0.4384
E18.5	0.3795	0.3746	0.3855	0.4023	0.4159	0.4070	0.4222	**0.4302**	0.4203
P4	0.3361	0.3768	0.3193	0.2800	0.3081	0.3574	0.3571	**0.4203**	0.4089
P14	0.3197	0.1593	**0.4193**	0.4006	0.3969	0.4122	0.3901	0.3974	0.3912
P28	0.3056	**0.3884**	0.3533	0.3304	0.3649	0.3749	0.3824	0.3792	0.3621
Avg.	0.3872	0.3895	0.4228	0.4105	0.4230	**0.4287**	0.4180	0.4236	0.4142

TABLE 4
Performance of the proposed method on the the Level 3 Allen Developing Mouse Brain Atlas data measured using Rand index. The regularization parameter is set to $\xi =$ percentage $\times \lambda_{\max}$, and the "percentage" is increased from 0.001 to 1. CC_{SVD} denotes the co-clustering method based on SVD proposed in [24]; $CC_{spectral}$ denotes the spectral co-clustering method proposed in [13].

Data	CC_{SVD}	$CC_{spectral}$	0.001	0.005	0.01	0.05	0.1	0.5	1
E11.5	0.8676	0.8529	0.8807	**0.8833**	0.8749	0.8634	0.8641	0.8825	0.8757
E13.5	0.8442	0.8394	0.8626	0.8648	0.8740	0.8607	**0.8760**	0.8446	0.8410
E15.5	0.7993	0.7787	**0.8514**	0.8142	0.8376	0.8212	0.8212	0.7872	0.7732
E18.5	0.7588	0.7274	**0.8573**	0.8371	0.8511	0.8334	0.8363	0.7777	0.7712
P4	0.6744	0.6629	0.5960	0.6708	0.6503	**0.7393**	0.6829	0.6889	0.7095
P14	0.6404	0.4902	0.7015	**0.7078**	0.6708	0.6674	0.6312	0.6854	0.7052
P28	0.6542	0.6674	0.6395	0.6449	0.6610	0.6307	0.6082	0.6112	**0.6695**
Avg.	0.7484	0.7170	0.7698	**0.7747**	0.7742	0.7737	0.7600	0.7539	0.7636

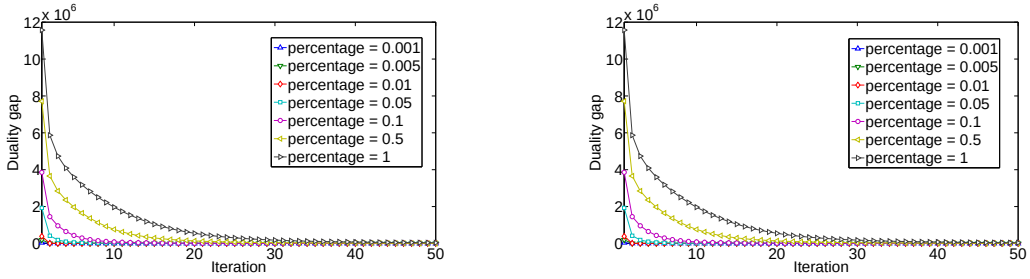
TABLE 5
Performance of the proposed method on the the Level 5 Allen Developing Mouse Brain Atlas data measured using Rand index. The regularization parameter is set to $\xi =$ percentage $\times \lambda_{\max}$, and the "percentage" is increased from 0.001 to 1. CC_{SVD} denotes the co-clustering method based on SVD proposed in [24]; $CC_{spectral}$ denotes the spectral co-clustering method proposed in [13].

Data	CC_{SVD}	$CC_{spectral}$	0.001	0.005	0.01	0.05	0.1	0.5	1
E11.5	0.9300	0.9293	0.9377	0.9428	0.9413	**0.9509**	0.9475	0.9377	0.9306
E13.5	0.8981	0.8993	**0.9214**	0.9009	0.8992	0.8878	0.8953	0.8921	0.8965
E15.5	0.8277	0.8237	0.8246	0.7957	0.7992	0.8815	**0.8883**	0.8185	0.8076
E18.5	0.8008	0.7943	0.7677	0.7766	0.7704	0.7831	**0.8819**	0.8219	0.7880
P4	0.6987	0.6986	0.5565	0.5102	0.5518	0.7374	0.7325	**0.7524**	0.7367
P14	0.6563	0.6489	0.7942	**0.8033**	0.7929	0.7971	0.7875	0.7638	0.7534
P28	0.6682	0.6748	0.8014	0.7790	0.7816	**0.8125**	0.7862	0.7185	0.7217
Avg.	0.7828	0.7813	0.8005	0.7869	0.7909	0.8358	**0.8456**	0.8150	0.8049



Level 3 data Level 5 data

Fig. 2. The values of the fused Lasso regularization terms as ξ increases. The x-axis denotes the "percentage" that is used to determine the value of ξ by $\xi = \text{percentage} \times \lambda_{\max}$.



The values of duality gap for the Level 3 data The values of duality gap for the Level 5 data

Fig. 3. The duality gap for the first 50 iterations under different ξ values. The x-axis denotes the "percentage" that is used to determine the value of ξ by $\xi = \text{percentage} \times \lambda_{\max}$.

Rand index for different λ' in Eq. (15). We select λ' to be some percentage of $\lambda_{\max}$ defined in Eq.(4.3). The percentage is varied from 0.001 to 1 and the performance is reported in Tables 6 and 8 for the Level 3 data sets, and Tables 7 and 9 for the Level 5 data sets, respectively. We can see from these results that our proposed method outperforms the other two compared methods for multiple different regularization parameter values. The best average performance is achieved at $\lambda' = 0.05 \times \lambda_{\max}$ for most cases. More importantly, the evolutionary feature selection method yields a set of shared features across developmental ages. These features correspond to genes in our data sets. Hence, our method identifies a set of genes that act continuously in multiple developmental ages. These genes might play important roles in the mouse brain development. We will analyze their functional and developmental roles in the future.

7 Conclusions and Discussions

In this paper, we propose evolutionary co-clustering and feature selection formulations for mining time-evolving data. The proposed formulations employ the fused Lasso type of regularization to encourage smoothness across time points. The resulting optimization problem is non-convex, non-smooth, and non-separable, and we employ an iterative procedure to compute the solution. Each step of the iterative procedure involves a convex problem. We derive the dual form of this problem and employ a gradient descent algorithm to compute the dual optimal solution. Experimental results on the Allen Developing Mouse Brain Atlas data show that the proposed methods yield consistently higher performance in comparison to other methods.

In this paper, we solve the dual form of the convex problem in each iteration. In the literature, coordinate descent and path algorithms have been developed to solve the fused Lasso signal approximator. We will explore and compare other alternative methods for solving this problem. This paper focuses on evaluating the proposed method on the mouse brain gene expression data, but this method can be applied to many other domains. We plan to apply our method to other data sets in the future. The selection of the fused Lasso regularization parameter is an important but challenging task. It has been shown that the stability selection is a promising way to tune the regularization parameters [32]. We plan to apply stability selection to tune the parameters in the future. Our current work does not consider tuning the smoothness parameter adaptively in order to incorporate different levels of smoothness at different time points [48]. We plan to extend our formulations to such scenarios in the future.

TABLE 6
Performance of the proposed method on the the Allen Developing Mouse Brain Atlas Level 3 data sets measured using NMI. The "percentage" is increased from 0.001 to 1, and $\lambda' = \text{percentage} \times \lambda_{\max}$. SK-means denotes the sparse K-means method.

Data	K-means	SK-means	0.001	0.005	0.01	0.05	0.1	0.5	1
E11.5	0.4757	0.4551	0.4609	0.4565	0.4906	0.4775	0.4713	**0.4923**	0.4852
E13.5	0.3491	0.3825	0.3841	**0.4158**	0.3827	0.3887	0.3746	0.3865	0.3935
E15.5	0.3592	0.3523	**0.3847**	0.3504	0.3720	0.3499	0.3354	0.3523	0.3779
E18.5	0.3645	**0.3797**	0.3490	0.3759	0.3333	0.3425	0.3436	0.3465	0.3252
P4	0.3727	0.3444	**0.3983**	0.3961	0.3756	0.3715	0.3866	0.3520	0.3784
P14	0.3560	0.3890	0.3221	**0.4097**	0.3554	0.3863	0.3536	0.3613	0.3694
P28	**0.3869**	0.3499	0.3560	0.3577	0.3503	0.3698	0.3484	0.3267	0.3040
Avg.	0.3806	0.3790	0.3793	**0.3946**	0.3800	0.3837	0.3733	0.3740	0.3762

TABLE 7
Performance of the proposed method on the the Allen Developing Mouse Brain Atlas Level 5 data sets measured using NMI. The "percentage" is increased from 0.001 to 1, and $\lambda' = \text{percentage} \times \lambda_{\max}$. SK-means denotes the sparse K-means method.

Data	K-means	SK-means	0.001	0.005	0.01	0.05	0.1	0.5	1
E11.5	0.5660	0.5802	0.5779	0.5696	**0.5805**	0.5803	0.5749	0.5659	0.5717
E13.5	0.5341	0.5266	0.5238	0.5458	0.5363	**0.5510**	0.5307	0.5271	0.5314
E15.5	0.4844	0.5019	0.4900	0.4871	0.5084	**0.5242**	0.5052	0.5046	0.4984
E18.5	0.4743	0.4751	0.4675	0.4598	0.4742	**0.4909**	0.4659	0.4538	0.4476
P4	0.4143	0.4311	0.4176	0.4275	0.4185	0.4388	**0.4430**	0.4250	0.4325
P14	0.3913	0.4028	0.3968	0.4066	0.4075	**0.4184**	0.4040	0.4018	0.4016
P28	0.3924	0.3886	0.3966	0.3955	0.4009	0.3921	0.3962	**0.4067**	0.3986
Avg.	0.4652	0.4723	0.4672	0.4703	0.4752	**0.4851**	0.4743	0.4693	0.4688

TABLE 8
Performance of the proposed method on the the Allen Developing Mouse Brain Atlas Level 3 data sets measured using Rand index. The "percentage" is increased from 0.001 to 1, and $\lambda' = \text{percentage} \times \lambda_{\max}$. SK-means denotes the sparse K-means method.

Data	K-means	SK-means	0.001	0.005	0.01	0.05	0.1	0.5	1
E11.5	0.8153	0.8339	0.8371	0.8359	0.8490	0.8615	0.8621	**0.8638**	0.8429
E13.5	0.8239	0.8034	**0.8267**	0.8070	0.8201	0.8179	0.8077	0.8135	0.8205
E15.5	0.7767	0.7925	0.7897	0.7683	0.7921	0.7880	0.7834	**0.7997**	0.7862
E18.5	0.7499	0.7621	**0.7738**	0.7576	0.7568	0.7685	0.7623	0.7534	0.7527
P4	0.6727	**0.6806**	0.6687	0.6783	0.6662	0.6768	0.6621	0.6730	0.6760
P14	0.6480	0.6244	**0.6501**	0.6220	0.6208	0.6264	0.6283	0.6346	0.6345
P28	**0.6627**	0.6559	0.6390	0.6602	0.6429	0.6493	0.6611	0.6526	0.6470
Avg.	0.7356	0.7361	0.7405	0.7328	0.7354	**0.7415**	0.7381	0.7415	0.7371

TABLE 9
Performance of the proposed method on the the Allen Developing Mouse Brain Atlas Level 5 data sets measured using Rand index. The "percentage" is increased from 0.001 to 1, and $\lambda' = \text{percentage} \times \lambda_{\max}$. SK-means denotes the sparse K-means method.

Data	K-means	SK-means	0.001	0.005	0.01	0.05	0.1	0.5	1
E11.5	0.9297	0.9298	0.9288	0.9300	0.9284	0.9291	0.9307	0.9298	**0.9310**
E13.5	0.8979	0.8983	**0.8992**	0.8985	0.8985	0.8959	0.8967	0.8977	0.8971
E15.5	0.8309	0.8280	**0.8324**	0.8286	0.8308	0.8282	0.8292	0.8296	0.8283
E18.5	0.8016	0.8017	0.8016	0.8002	0.8023	**0.8035**	0.8016	0.8028	0.8033
P4	0.7008	0.7017	0.7028	0.7025	0.7018	**0.7046**	0.7005	0.7034	0.7003
P14	0.6563	0.6569	0.6586	0.6566	**0.6596**	0.6593	0.6574	0.6564	0.6561
P28	0.6693	0.6711	0.6727	0.6694	0.6716	**0.6733**	0.6706	0.6695	0.6703
Avg.	0.7838	0.7839	**0.7852**	0.7837	0.7847	0.7848	0.7838	0.7842	0.7838

ACKNOWLEDGMENTS

This work was supported by NSF DBI-1147134, DBI-1356621, NSF CAREER Award DBI-1350258, National Basic Research Program of China (No.2012CB316400), Program for Changjiang Scholars and Innovative Research Team in University (No.IRT201206), and Program for New Century Excellent Talents in University (No.13-0661).

REFERENCES

[1] C. Aggarwal and K. Subbian. Evolutionary network analysis: A survey. *ACM Computing Surveys*, 47(1):10, 2014.

[2] S. Alelyani, J. Tang, and H. Liu. Feature selection for clustering: A review. *Data Clustering: Algorithms and Applications, Editor: Charu Aggarwal and Chandan Reddy, CRC Press*.

[3] Allen Institute for Brain Science. Allen Developing Mouse Brain Atlas [Internet], 2012.

[4] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

[5] J. W. Bohland and *et al.* Clustering of spatial gene expression patterns in the mouse brain and comparison with classical neuroanatomy. *Methods*, 50(2):105–112, 2010.

[6] S. Busygin, O. Prokopyev, and P. M. Pardalos. Biclustering in data mining. *Computers and Operations Research*, 35:2964–2987, September 2008.

[7] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 554–560, 2006.

[8] C. Chen, J. Huang, L. He, and H. Li. Preconditioning for accelerated iteratively reweighted least squares in structured sparsity reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2713–2720, 2013.

[9] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.

[10] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. On evolutionary spectral clustering. *ACM Transactions on Knowledge Discovery from Data*, 3:17:1–17:30, December 2009.

[11] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra. Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 114–125, 2004.

[12] M. Deodhar and J. Ghosh. SCOAL: A framework for simultaneous co-clustering and learning from complex data. *ACM Trans. Knowl. Discov. Data*, 4(3):11:1–11:31, 2010.

[13] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274, 2001.

[14] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98, 2003.

[15] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[16] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, 2007.

[17] E. Giannakidou, V. Koutsonikola, A. Vakali, and Y. Kompatsiaris. Co-clustering tags and social data sources. In *Proceedings of the 2008 The Ninth International Conference on Web-Age Information Management*, pages 317–324, 2008.

[18] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[19] H. Höfling. A path algorithm for the fused Lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

[20] J. Huang, S. Zhang, H. Li, and D. Metaxas. Composite splitting algorithms for convex optimization. *Computer Vision and Image Understanding*, 115(12):1610–1622, 2011.

[21] S. Ji. Computational network analysis of the anatomical and genetic organizations in the mouse brain. *Bioinformatics*, 27(23):3293–3299, 2011.

[22] S. Ji, W. Zhang, and J. Liu. A sparsity-inducing formulation for evolutionary co-clustering. In *Proceedings of the Eighteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 334–342, 2012.

[23] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13(4):703–716, 2003.

[24] M. Lee, H. Shen, J. Z. Huang, and J. S. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66:1087–1095, 2010.

[25] E. S. Lein *et al.* Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445(7124):168–176, 2007.

[26] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data*, 3:8:1–8:31, April 2009.

[27] J. Liu, S. Ji, and J. Ye. SLEP: Sparse Learning with Efficient Projections. *Arizona State University*, 2009. http://www.public.asu.edu/~jye02/Software/SLEP/.

[28] J. Liu, L. Yuan, and J. Ye. An efficient algorithm for a class of fused Lasso problems. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–332, 2010.

[29] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1:24–45, January 2004.

[30] C. Manning and P. Raghavan *et al.* *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[31] D. Mavroeidis and E. Marchiori. Feature selection for k-means clustering stability: theoretical analysis and an algorithm. *Data Mining and Knowledge Discovery*, 28(4):918–960, 2014.

[32] N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[33] A. Nemirovski. Efficient methods in convex programming, 1994. Lecture Notes.

[34] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2003.

[35] L. Puelles and J. L. Rubenstein. Forebrain gene expression domains and the evolving prosomeric model. *Trends in neurosciences*, 26(9):469–476, 2003.

[36] C. Soneson and M. Fontes. A method for visual identification of small sample subgroups and potential biomarkers. *Annals of Applied Statistics*, 5(3):2131–2149, 2011.

[37] S. M. Sunkin, L. Ng, C. Lau, T. Dolbeare, T. L. Gilbert, C. L. Thompson, M. Hawrylycz, and C. Dang. Allen Brain Atlas: an integrated spatio-temporal portal for exploring the central nervous system. *Nucleic Acids Research*, 2012.

[38] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1):267–288, 1996.

[39] R. Tibshirani and M. Saunders. Sparsity and smoothness via the fused lasso. *Journal of Royal Statistics Society B*, 67(1):91–108, 2005.

[40] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[41] R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, 9(1):18–29, 2008.

[42] H. Tong, S. Papadimitriou, P. S. Yu, and C. Faloutsos. Proximity tracking on time-evolving bipartite graphs. In *Proceedings of the SIAM International Conference on Data Mining*, pages 704–715, 2008.

[43] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, June 2001.

[44] F. Wang, H. Tong, and C.-Y. Lin. Towards evolutionary nonnegative matrix factorization. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[45] L. Wasserman, M. Azizyan, and A. Singh. Feature selection for high-dimensional clustering. *arXiv preprint arXiv:1406.2240*, 2014.

[46] C. Watson, G. Paxinos, and L. Puelles. *The Mouse Nervous System*. Academic Press, 2011.

[47] D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

[48] K. S. Xu, M. Kliger, and A. O. Hero Iii. Adaptive evolutionary clustering. *Data Mining and Knowledge Discovery*, 28(2):304–336, 2014.

[49] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Bipartite graph partitioning and data clustering. In *Proceedings of the tenth International Conference on Information and Knowledge Management*, pages 25–32, 2001.

[50] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu. Advancing feature selection research-asu feature selection repository. *School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe*, 2010.

Rongjian Li received the BS and MS degrees in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2007 and 2012, respectively. Currently, he is working towards the PhD degree in the Department of Computer Science at Old Dominion University. His research interests include machine learning, data mining, and bioinformatics.

Wenlu Zhang received the BS degree in Computer Science from Information Engineering University, Zhengzhou, China, in 2008, the MS degree in Computer Science from City College of New York, New York, NY, in 2010. Currently, she is a PhD candidate of Computer Science at Old Dominion University. Her research interests include machine learning, data mining, and bioinformatics.

Yao Zhao received the BS degree from Fuzhou University, China, in 1989, and the ME degree from Southeast University, Nanjing, China, in 1992, both from the Radio Engineering Department, and the PhD degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), China, in 1996. He became an associate professor at BJTU in 1998 and became a professor in 2001. From 2001 to 2002, he was a senior research fellow with the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. He is currently the director of the Institute of Information Science, BJTU. His current research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding. He serves on the editorial boards of several international journals, including as associate editors of IEEE Transactions on Cybernetics, IEEE Signal Processing Letters, and an area editor of Signal Processing: Image Communication (Elsevier), etc. He was named a distinguished young scholar by the National Science Foundation of China in 2010, and was elected as a Chang Jiang Scholar of Ministry of Education of China in 2013. He is a senior member of the IEEE and a fellow of IET.

Zhenfeng Zhu received the PhD degree in pattern recognition and intelligence system from the Institute of Automation, Chinese Academy of Sciences, in 2005. He is currently an associate professor of the Institute of Information Science, Beijing Jiaotong University. He has been a visiting scholar at the Department of Computer Science and Engineering, Arizona State University, during 2010. His research interests include image and video understanding, computer vision, and machine learning.

Shuiwang Ji received the PhD degree in computer science from Arizona State University, Tempe, Arizona, in 2010. Currently, he is an assistant professor of Computer Science, Old Dominion University, Norfolk, Virginia. His research interests include machine learning, data mining, computational neuroscience, and bioinformatics. He received the National Science Foundation CAREER Award in 2014. He is currently an Associate Editor for BMC Bioinformatics, IEEE Transactions on Neural Networks and Learning Systems, and Neurocomputing. He is a member of IEEE, IEEE Computer Society, and IEEE Computational Intelligence Society.