A Sparsity-Inducing Formulation for Evolutionary Co-Clustering

Shuiwang Ji
Old Dominion University
Norfolk, VA 23529
sji@cs.odu.edu

Wenlu Zhang
Old Dominion University
Norfolk, VA 23529
wzhang@cs.odu.edu

Jun Liu
Siemens Corporate Research
Princeton, NJ 08540
jun-liu@siemens.com

ABSTRACT

Traditional co-clustering methods identify block structures from static data matrices. However, the data matrices in many applications are dynamic; that is, they evolve smoothly over time. Consequently, the hidden block structures embedded into the matrices are also expected to vary smoothly along the temporal dimension. It is therefore desirable to encourage smoothness between the block structures identified from temporally adjacent data matrices. In this paper, we propose an evolutionary co-clustering formulation for identifying co-cluster structures from time-varying data. The proposed formulation encourages smoothness between temporally adjacent blocks by employing the fused Lasso type of regularization. Our formulation is very flexible and allows for imposing smoothness constraints over only one dimension of the data matrices, thereby enabling its applicability to a large variety of settings. The optimization problem for the proposed formulation is non-convex, non-smooth, and non-separable. We develop an iterative procedure to compute the solution. Each step of the iterative procedure involves a convex, but non-smooth and non-separable problem. We propose to solve this problem in its dual form, which is convex and smooth. This leads to a simple gradient descent algorithm for computing the dual optimal solution. We evaluate the proposed formulation using the Allen Developing Mouse Brain Atlas data. Results show that our formulation consistently outperforms methods without the temporal smoothness constraints.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining

General Terms

Algorithms

KDD'12, August 12–16, 2012, Beijing, China.

Keywords

Sparsity learning, evolutionary co-clustering, optimization, bioinformatics, neuroscience

1. INTRODUCTION

Clustering is one of the major techniques for unsupervised discovery of hidden structures from complex data sets. Common clustering methods include the K-means algorithm, the spectral methods, and the hierarchical clustering techniques. These approaches treat the two dimensions of the data matrix as instances and features, respectively, and group the instances into clusters based on all the features. When the data matrix is re-ordered according to the clustering results, the re-ordered matrix usually assumes a banded structure along the instance dimension, since instances in the same cluster are assumed to be similar, while those in different cluster are assumed to be dissimilar.

The classical clustering paradigm assumes that all the features are equally relevant to all instances. In many applications, certain group of instances are only similar to each other with respect to a subset of features. 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. Co-clustering, also known as bi-clustering, aims at identifying the block structures of the data matrix by clustering the rows and columns of the data matrix simultaneously into co-clusters [17, 6, 9, 10, 28]. Currently, co-clustering finds applications in many areas, including biological data analysis [25, 20], text mining [10, 9], and social studies [14].

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.

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 [29] to identify block structures from the time-varying data matrices. Meanwhile, it applies fused Lasso type of regularization [30] 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. The optimization problem for the proposed formulation is non-convex, non-smooth, and non-separable. We propose an iterative procedure to compute the solution, and 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 formulation using the Allen Developing Mouse Brain Atlas data [22, 19]. Results show that the proposed method consistently outperforms other methods by identifying blocks that are consistent with classical neuroanatomy.

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

Notations: We use boldface lower-case letters, e.g., $\mathbf{u}$, to denote vectors and upper-case letters, e.g., A, to denote matrices. $\mathbf{e}_n$ denotes a vector of all ones of length n. 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 A, its Frobenius norm is denoted as $\|A\|_F$. $\odot$ denotes component-wise multiplication, and $\otimes$ denotes the Kronecker product. The soft-thresholding operator $\mathcal{T}_\lambda$, acting on a vector $\mathbf{x}$, is defined component-wise as:

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

2. 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 [9, 36, 21]. In [9, 36], 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 [21] for co-clustering. Formally, let $B \in \mathbb{R}^{m\times n}$ be a data matrix. The first singular value and the corresponding left and right singular vectors of B can be computed as

$$\min_{s,\mathbf{p},\mathbf{q}} \|B - s\mathbf{p}\mathbf{q}^T\|_F^2, \quad (2)$$

where $s \in \mathbb{R}$ is the first singular value, and $\mathbf{p} \in \mathbb{R}^m$ and $\mathbf{q} \in \mathbb{R}^n$ 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 B [12]. Note that $\mathbf{p}$ and $\mathbf{q}$ lie in the row space and column space, respectively, of B. 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 the SVD, a sparse SVD formulation is proposed in [21]. 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:

$$\min_{s,\mathbf{p},\mathbf{q}} \frac{1}{2}\|B - s\mathbf{p}\mathbf{q}^T\|_F^2 + \lambda\|s\mathbf{p}\|_1 + \gamma\|s\mathbf{q}\|_1, \quad (3)$$

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 [29]. 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 [21] that the sparse SVD formulation can be readily employed to solve co-clustering problems. Specifically, the rows and columns of B 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. (3) 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. (3) is non-convex and non-smooth. An iterative procedure has been developed in [21] 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

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

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:

$$\min_{\tilde{\mathbf{p}}} G(\tilde{\mathbf{p}}) \equiv \frac{1}{2}\|B - \tilde{\mathbf{p}}\mathbf{q}^T\|_F^2 + \lambda\|\tilde{\mathbf{p}}\|_1, \quad (5)$$

and $\mathbf{p} = \tilde{\mathbf{p}}/s$ where $s = \|\tilde{\mathbf{p}}\|$. It can be shown that the problems in Eqs. (4) and (5) are convex and can be solved analytically.

The objective function in Eq. (4) can be written as

$$\begin{aligned} F(\tilde{\mathbf{q}}) &= \frac{1}{2}\|B - \mathbf{p}\tilde{\mathbf{q}}^T\|_F^2 + \gamma\|\tilde{\mathbf{q}}\|_1 \\ &= \frac{1}{2}\operatorname{Tr}(B^TB) - \mathbf{p}^TB\tilde{\mathbf{q}} + \frac{1}{2}\tilde{\mathbf{q}}^T\tilde{\mathbf{q}} + \gamma\|\tilde{\mathbf{q}}\|_1. \quad (6) \end{aligned}$$

Taking the subdifferential of Eq. (6) with respect to $\tilde{\mathbf{q}}$, we have

$$\partial F(\tilde{\mathbf{q}}) = -B^T\mathbf{p} + \tilde{\mathbf{q}} + \gamma\,\mathrm{SGN}(\tilde{\mathbf{q}}), \quad (7)$$

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

$$(\mathrm{SGN}(\tilde{\mathbf{q}}))_i = \begin{cases} \{1\} & \text{if } (\tilde{\mathbf{q}})_i > 0 \\ \{-1\} & \text{if } (\tilde{\mathbf{q}})_i < 0 \\ [-1, 1] & \text{if } (\tilde{\mathbf{q}})_i = 0. \end{cases} \quad (8)$$

Note that the subdifferential of a function is a set, and when the function is differentiable, the set is a singleton containing the derivative [27]. It follows from the optimality condition for unconstrained problems [27] that $\tilde{\mathbf{q}}^*$ is an optimal solution to Eq. (4) if and only if $\mathbf{0} \in \partial F(\tilde{\mathbf{q}}^*)$. Hence, it can be

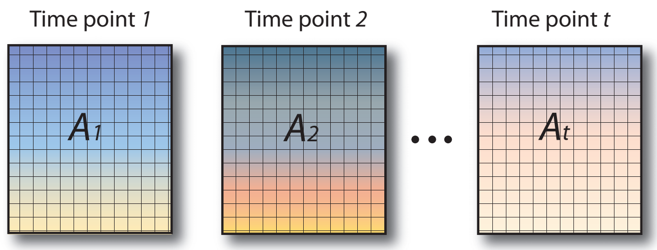

Figure 1: Illustration of the evolutionary co-clustering problem.

easily verified that the optimal $\tilde{\mathbf{q}}^*$ is given by

$$(\tilde{\mathbf{q}}^*)_i = \begin{cases} (B^T\mathbf{p}-\gamma)_i & \text{if } (B^T\mathbf{p})_i > \gamma \\ (B^T\mathbf{p}+\gamma)_i & \text{if } (B^T\mathbf{p})_i < -\gamma \\ 0 & \text{if } |(B^T\mathbf{p})_i| \le \gamma. \end{cases} \qquad (9)$$

Similarly, the optimal $\tilde{\mathbf{p}}^*$ for the optimization problem in Eq. (5) is given by

$$(\tilde{\mathbf{p}}^*)_i = \begin{cases} (B\mathbf{q}-\lambda)_i & \text{if } (B\mathbf{q})_i > \lambda \\ (B\mathbf{q}+\lambda)_i & \text{if } (B\mathbf{q})_i < -\lambda \\ 0 & \text{if } |(B\mathbf{q})_i| \le \lambda. \end{cases} \qquad (10)$$

The iterative procedure in [21] applies Eqs. (9) and (10) alternately until a locally optimal solution is reached.

3. EVOLUTIONARY CO-CLUSTERING

In the traditional co-clustering framework [17, 6, 9, 20, 28, 25], 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 as shown in Figure 1. 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 $A_i \in \mathbb{R}^{m\times n}$ for $i = 1, \cdots, t$, where t 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:

$$\min_{s_i,\mathbf{u}_i,\mathbf{v}_i} \sum_{i=1}^{t} \left\{ \frac{1}{2}\|A_i - s_i\mathbf{u}_i\mathbf{v}_i^T\|_F^2 + \lambda\|s_i\mathbf{u}_i\|_1 + \gamma\|s_i\mathbf{v}_i\|_1 \right\}$$

where $\mathbf{u}_i \in \mathbb{R}^m$ and $\mathbf{v}_i \in \mathbb{R}^n$ are associated with the rows and columns, respectively, of A_i, and s_i 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.

3.1 The Proposed Formulation

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

$$\min_{s_i,\mathbf{u}_i,\mathbf{v}_i} \sum_{i=1}^{t} \left\{ \frac{1}{2}\|A_i - s_i\mathbf{u}_i\mathbf{v}_i^T\|_F^2 + \lambda\|s_i\mathbf{u}_i\|_1 + \gamma\|s_i\mathbf{v}_i\|_1 \right\} (11)$$
$$+ \sum_{i=1}^{t-1} \left\{ \eta\|s_{i+1}\mathbf{u}_{i+1} - s_i\mathbf{u}_i\|_1 + \xi\|s_{i+1}\mathbf{v}_{i+1} - s_i\mathbf{v}_i\|_1 \right\},$$

where η and ξ and tunable parameters. In this formulation, the last two regularization terms are fused Lasso type of regularization [30], and they encourage the $\mathbf{u}_i$ and $\mathbf{v}_i$ for contiguous time points to be similar. Specifically, these regularization terms encourage the differences of contiguous $\mathbf{u}_i$ and $\mathbf{v}_i$ to be zero, thus enforcing many entries of contiguous $\mathbf{u}_i$ and $\mathbf{v}_i$ 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. (11) can be expressed equivalently as

$$\sum_{i=1}^{t} \frac{1}{2}\|A_i - s_i\mathbf{u}_i\mathbf{v}_i^T\|_F^2 + \lambda\|\tilde{\mathbf{u}}\|_1 + \gamma\|\tilde{\mathbf{v}}\|_1 + \eta\|E\tilde{\mathbf{u}}\|_1 + \xi\|F\tilde{\mathbf{v}}\|_1,$$

where $\tilde{\mathbf{u}} = (\mathbf{s}\otimes\mathbf{e}_m)\odot\mathbf{u}$, $\mathbf{s} = [s_1, s_2, \cdots, s_t]^T$, $\tilde{\mathbf{v}} = (\mathbf{s}\otimes\mathbf{e}_n)\odot\mathbf{v}$, $\mathbf{u} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \cdots, \mathbf{u}_t^T]^T \in \mathbb{R}^{mt}$, $\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \cdots, \mathbf{v}_t^T]^T \in \mathbb{R}^{nt}$, $E \in \mathbb{R}^{m(t-1)\times mt}$ and $F \in \mathbb{R}^{n(t-1)\times nt}$ are defined as

$$(E)_{ij} = \begin{cases} -1 & \text{if } j = i, \;\; i = 1, \cdots, m(t-1) \\ 1 & j = i+m, \;\; i = 1, \cdots, m(t-1) \\ 0 & \text{otherwise,} \end{cases} \qquad (12)$$

$$(F)_{ij} = \begin{cases} -1 & \text{if } j = i, \;\; i = 1, \cdots, n(t-1) \\ 1 & j = i+n, \;\; i = 1, \cdots, n(t-1) \\ 0 & \text{otherwise.} \end{cases} \qquad (13)$$

The objective function in Eq. (11) is non-convex and non-smooth. In addition, the fused Lasso regularization terms are non-separable [33, 13]. 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:

$$\min_{\tilde{\mathbf{v}}} f_\xi^\gamma(\tilde{\mathbf{v}}) \equiv \sum_{i=1}^t \frac{1}{2}\|A_i - \mathbf{u}_i\tilde{\mathbf{v}}_i^T\|_F^2 + \gamma\|\tilde{\mathbf{v}}\|_1 + \xi\|F\tilde{\mathbf{v}}\|_1. \quad (14)$$

The objective function in Eq. (14) is convex, but non-smooth and non-separable. In the following, we develop an efficient algorithm to compute the optimal $\tilde{\mathbf{v}}^*$.

3.2 A Two-Step Procedure

A central challenge for solving the optimization problem in Eq. (14) is the ℓ_1-norm and the fused Lasso regularization terms, which are 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 [13, 24] and is summarized in the following theorem:

Theorem 3.1. *Define*

$$\pi_\xi^\gamma(\mathbf{u}) = \arg\min_{\tilde{\mathbf{v}}} f_\xi^\gamma(\tilde{\mathbf{v}}). \quad (15)$$

Then for any $\gamma, \xi \geq 0$, we have

$$\pi_\xi^\gamma(\mathbf{u}) = \mathcal{T}_\gamma\left(\pi_\xi^0(\mathbf{u})\right). \quad (16)$$

Proof. We consider the case when $\gamma = 0$:

$$\pi_\xi^0(\mathbf{u}) = \arg\min f_\xi(\tilde{\mathbf{v}}), \quad (17)$$

where

$$f_\xi(\tilde{\mathbf{v}}) \equiv f_\xi^0(\tilde{\mathbf{v}}) = \sum_{i=1}^t \frac{1}{2}\|A_i - \mathbf{u}_i\tilde{\mathbf{v}}_i^T\|_F^2 + \xi\|F\tilde{\mathbf{v}}\|_1. \quad (18)$$

The subdifferentials of $f_\xi^\gamma(\tilde{\mathbf{v}})$ and $f_\xi(\tilde{\mathbf{v}})$ can be computed as

$$\begin{aligned} \partial f_\xi^\gamma(\tilde{\mathbf{v}}) &= \tilde{\mathbf{v}} - A^T\mathbf{u} + \gamma\,\mathrm{SGN}(\tilde{\mathbf{v}}) + \xi F^T\,\mathrm{SGN}(F\tilde{\mathbf{v}}), \quad (19)\\ \partial f_\xi(\tilde{\mathbf{v}}) &= \tilde{\mathbf{v}} - A^T\mathbf{u} + \xi F^T\,\mathrm{SGN}(F\tilde{\mathbf{v}}), \quad (20) \end{aligned}$$

where

$$A = \begin{pmatrix} A_1 & & & 0 \\ & A_2 & & \\ & & \ddots & \\ 0 & & & A_t \end{pmatrix} \in \mathbb{R}^{mt\times nt}. \quad (21)$$

It follows from the optimality condition for unconstrained problems [27] that

$$\mathbf{0} \in \partial f_\xi\left(\pi_\xi^0(\mathbf{u})\right).$$

Hence, there exists

$$\mathbf{y}^* \in \xi\,\mathrm{SGN}(F\pi_\xi^0(\mathbf{u})) \quad (22)$$

such that $\pi_\xi^0(\mathbf{u}) = A^T\mathbf{u} - F^T\mathbf{y}^*$. Define

$$\begin{aligned} \mathbf{a} &= \mathcal{T}_\gamma(\pi_\xi^0(\mathbf{u})), \\ \mathbf{b} &= \mathrm{sgn}(\pi_\xi^0(\mathbf{u})) \odot \min(|\pi_\xi^0(\mathbf{u})|, \gamma), \end{aligned}$$

where $\mathrm{sgn}(\mathbf{x})$ is defined component-wise as $(\mathrm{sgn}(\mathbf{x}))_i = 1$ if $x_i > 0$, $(\mathrm{sgn}(\mathbf{x}))_i = -1$ if $x_i < 0$, and 0 otherwise. It can be verified that $\mathbf{a} - A^T\mathbf{u} + \mathbf{b} + F^T\mathbf{y}^* = \mathbf{0}$ and $\mathbf{b} \in \gamma\,\mathrm{SGN}(\mathbf{a})$. It follows from the definition of $\mathbf{a}$, the fact that each row of F consisting of two nonzero elements 1 and -1, and Eq. (22) that $\mathbf{y}^* \in \xi\,\mathrm{SGN}(F\mathbf{a})$. Therefore, we have

$$\mathbf{0} = \mathbf{a} - A^T\mathbf{u} + \mathbf{b} + F^T\mathbf{y}^* \in \partial f_\xi^\gamma(\mathbf{a}).$$

This completes the proof of this theorem. □

Theorem 3.1 shows that we can solve the optimization problem in two sequential steps. Specifically, we can first solve the problem in Eq. (14) with $\gamma = 0$ to obtain the intermediate solution $\pi_\xi^0(\mathbf{u})$. Then the final optimal solution $\pi_\xi^\gamma(\mathbf{u})$ can be obtained by applying the soft thresholding operator to the intermediate solution as in Eq. (16). We now discuss how the $\gamma = 0$ case can be solved efficiently in its dual form.

3.3 The Dual Formulation

A key to the two-step procedure in Section 3.2 is to solve the optimization problem in Eq. (17), which can be rewritten in its full form as

$$\min_{\tilde{\mathbf{v}}} f_\xi(\tilde{\mathbf{v}}) \equiv \sum_{i=1}^t \frac{1}{2}\|A_i - \mathbf{u}_i\tilde{\mathbf{v}}_i^T\|_F^2 + \xi\|F\tilde{\mathbf{v}}\|_1. \quad (23)$$

We propose to solve this problem in its dual form. To this end, we introduce the dual variable

$$\mathbf{w} = [\mathbf{w}_1^T, \mathbf{w}_2^T, \cdots, \mathbf{w}_{t-1}^T]^T \in \mathbb{R}^{n(t-1)}$$

and obtain the following equivalent min-max problem:

$$\min_{\tilde{\mathbf{v}}} \max_{\|\mathbf{w}\|_\infty \leq \xi} \phi(\tilde{\mathbf{v}}, \mathbf{w}) \equiv \sum_{i=1}^t \frac{1}{2}\|A_i - \mathbf{u}_i\tilde{\mathbf{v}}_i^T\|_F^2 + \langle F\tilde{\mathbf{v}}, \mathbf{w}\rangle. \quad (24)$$

The existence of the saddle point to this min-max problem is guaranteed by the Von Neumann Lemma [26], because $\phi(\cdot,\cdot)$ is differentiable, convex in $\tilde{\mathbf{v}}$, and concave in $\mathbf{w}$. Exchanging the min and max and setting the derivative of $\phi(\tilde{\mathbf{v}}, \mathbf{w})$ with respect to $\tilde{\mathbf{v}}$ to zero, we obtain

$$\tilde{\mathbf{v}} = A^T\mathbf{u} - F^T\mathbf{w}. \quad (25)$$

Substituting Eq. (25) into Eq. (24), we obtain the following dual problem:

$$\min_{\mathbf{w}:\|\mathbf{w}\|_\infty \leq \xi} \psi(\mathbf{w}) \equiv \frac{1}{2}\|F^T\mathbf{w}\|^2 - \left\langle A^T\mathbf{u}, F^T\mathbf{w}\right\rangle - c, \quad (26)$$

where $c = \frac{1}{2}\sum_{i=1}^t \mathrm{Tr}\left((A_i - \mathbf{u}_i\mathbf{u}_i^T A_i)(A_i - \mathbf{u}_i\mathbf{u}_i^T A_i)^T\right)$. Note that we have changed max to min in Eq. (26) by negating the objective function for ease of presentation. The dual formulation in Eq. (26) is convex and smooth. Hence, it can be solved by gradient descent algorithms.

3.4 A Gradient Descent Algorithm

The dual problem in Eq. (26) is a constrained quadratic program (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 $\psi(\mathbf{w})$ is a $n(t-1) \times n(t-1)$ matrix and can be express as

$$FF^T = \begin{pmatrix} 2 & \overbrace{\cdots}^{(n-1)\,0s} & -1 & \cdots & \cdots & 0 \\ \vdots & 2 & \cdots & -1 & \cdots & 0 \\ -1 & \vdots & \ddots & \cdots & \ddots & \vdots \\ \vdots & -1 & \vdots & \ddots & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & \cdots & 2 \end{pmatrix}. \quad (27)$$

Since F is a full rank matrix, the Hessian matrix FF^T is positive definite. Thus a unique solution exists for the optimization problem in Eq. (26).

In this gradient descent algorithm, we have the following iterative update in each iteration:

$$\mathbf{w}_{k+1} = P_{\|\cdot\|_\infty \leq \xi} \left(\mathbf{w}_k - \frac{1}{\lambda_{\max}} \mathbf{g}_k \right), \tag{28}$$

where $\mathbf{g}_k = \psi'(\mathbf{w}_k) = FF^T\mathbf{w}_k - FA^T\mathbf{u}$ is the gradient of the objective function at $\mathbf{w}_k$, $\lambda_{\max}$ is the largest eigenvalue of the Hessian matrix FF^T, and

$$\left(P_{\|\cdot\|_\infty \leq \xi}(\mathbf{x})\right)_i = \begin{cases} x_i & \text{if } |x_i| \leq \xi \\ \mathrm{sgn}(x_i)\xi & \text{if } |x_i| > \xi \end{cases} \tag{29}$$

is the projection onto the feasible region. It follows from the analysis in [27] that this algorithm has a linear convergence rate as

$$\|\mathbf{w}_k - \mathbf{w}^*\|^2 \leq \left(1 - \frac{\lambda_{\min}}{\lambda_{\max}}\right)^k \|\mathbf{w}_0 - \mathbf{w}^*\|^2, \tag{30}$$

where $\mathbf{w}_0$ is the initial starting point, and $\lambda_{\min}$ denotes the smallest eigenvalue of the Hessian matrix. This algorithm can also be accelerated by the Nesterov's method [27].

3.5 Duality Gap and Convergence

The gradient descent algorithm is an iterative procedure, and thus a criterion is required to assess the convergence of the algorithm. Following [24], we define a duality gap for the min-max problem in Eq. (24) 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 $\bar{\mathbf{w}}$ be an appropriate solution computed by the gradient descent algorithm. Note that $\|\bar{\mathbf{w}}\|_\infty \leq \xi$, as it has been projected onto the feasible region in each step. Let $\bar{\mathbf{v}} = A^T\mathbf{u} - F^T\bar{\mathbf{w}}$ be the corresponding solution for the primal formulation. We can define the duality gap for Eq. (24) at $(\bar{\mathbf{v}}, \bar{\mathbf{w}})$ as

$$\mathrm{dg}(\bar{\mathbf{v}}, \bar{\mathbf{w}}) = \max_{\mathbf{w}:\|\mathbf{w}\|_\infty \leq \xi} \phi(\bar{\mathbf{v}}, \mathbf{w}) - \min_{\tilde{\mathbf{v}}} \phi(\tilde{\mathbf{v}}, \bar{\mathbf{w}}). \tag{31}$$

The following results show that the duality gap in Eq. (31) 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 3.2. *The duality gap defined in Eq. (31) can be computed as*

$$dg(\bar{\mathbf{v}}, \bar{\mathbf{w}}) = \xi\|\psi'(\bar{\mathbf{w}})\|_1 + \left\langle \bar{\mathbf{w}}, \psi'(\bar{\mathbf{w}}) \right\rangle. \tag{32}$$

In addition, we have the following results:

$$\psi(\bar{\mathbf{w}}) - \psi(\mathbf{w}^*) \quad \leq \quad dg(\bar{\mathbf{v}}, \bar{\mathbf{w}}), \tag{33}$$

$$f_\xi(\bar{\mathbf{v}}) - f_\xi(\mathbf{v}^*) \quad \leq \quad dg(\bar{\mathbf{v}}, \bar{\mathbf{w}}). \tag{34}$$

The proof of this theorem is similar to that of Theorem 3 in [24] and is thus omitted.

3.6 Regularization Parameter Interval

The regularization parameter ξ controls the temporal smoothness over $\mathbf{v}_i$. That is, when ξ is larger than a certain value $\xi_{\max}$, $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$, for $i = 1, 2, \cdots, t-1$, will be enforced to be identical. We show that such a $\xi_{\max}$ can be computed via solving a linear system of equations. To this end, we need to state the optimality condition for the problem in Eq. (26).

It follows from the optimality condition for constrained problems [27] that $\mathbf{w}^*$ ($\|\mathbf{w}^*\|_\infty \leq \xi$) is a minimizer of Eq. (26) if and only if

$$\left\langle \psi'(\mathbf{w}^*), \mathbf{w} - \mathbf{w}^* \right\rangle \geq 0, \ \ \forall\, \mathbf{w} : \|\mathbf{w}\|_\infty \leq \xi. \tag{35}$$

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 $\xi_{\max}$ can be computed via solving a linear system of equations with a special structure.

Theorem 3.3. *Let $\hat{\mathbf{w}}$ denote the unique solution of the linear system*

$$FF^T\mathbf{w} = FA^T\mathbf{u}, \tag{36}$$

and let

$$\xi_{max} = \|\hat{\mathbf{w}}\|_\infty. \tag{37}$$

Then for any $\xi \geq \xi_{max}$, we have $\tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_j$, $\forall\, i, j$.

Proof. Since the Hessian of $\psi(\cdot)$ is positive definite, the linear system in Eq. (36) has a unique solution $\hat{\mathbf{w}}$. For any $\xi \geq \xi_{\max}$, it can be easily verified that $\|\hat{\mathbf{w}}\|_\infty = \xi_{\max} \leq \xi$ and $\psi'(\hat{\mathbf{w}}) = FF^T\hat{\mathbf{w}} - FA^T\mathbf{u} = \mathbf{0}$. It follows from the optimality condition in Eq. (35) that $\hat{\mathbf{w}}$ is the optimal solution to Eq. (26) when $\xi \geq \xi_{\max}$. In addition, when $\xi \geq \xi_{\max}$, we have $\pi_\xi(\mathbf{u}) = A^T\mathbf{u} - F^T\hat{\mathbf{w}}$ from Eq. (25). It follows that

$$F\pi_\xi(\mathbf{u}) = F(A^T\mathbf{u} - F^T\hat{\mathbf{w}}) = \mathbf{0}.$$

Therefore, we have $\tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_j$, $\forall\, i, j$. □

The value of $\xi_{\max}$ can be used to guide the selection of an appropriate value for ξ in practice. We evaluate the effectiveness of ξ in the experiments and observe that the best performance is achieved when $\xi = \xi_{\max}$ on the biological data sets.

4. RELATED WORK

Simultaneous row and column clustering for identifying block structures from matrix data has been initially studied in [17]. 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 [6]. Co-clustering has also been applied in many other applications, including simultaneous clustering of words and documents [10, 9], authors and conference [32], etc. Early work on co-clustering focuses on defining an error measure and then identifying blocks that minimize this measure using heuristic search algorithms [17, 6]. These early work has recently been reformulated using matrix and optimization techniques [8]. 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 [4]. In [9, 36], 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 [21] that sparsity-inducing regularization can be employed to compute sparse singular vectors, which in turn can be used to form co-clusters.

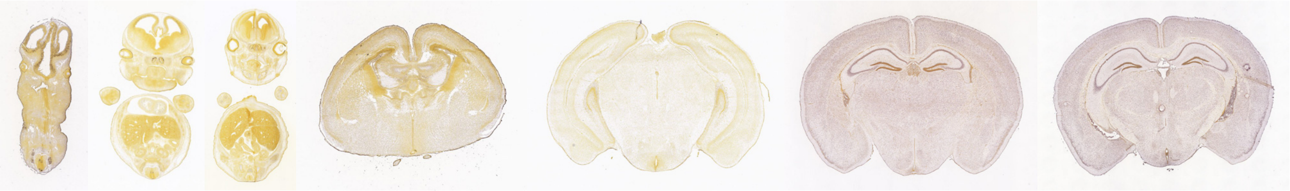

Figure 2: Sample slices of the 3D expression patterns for the gene *Neurog1* across seven developmental ages shown in the coronal view.

Table 1: Statistics of the Allen Developing Mouse Brain Atlas data used in the experiments. For each developing age, the data consist of a matrix in which one dimension corresponds to the brain voxels while the other dimension corresponds to the genes. This data set contains 7 developing ages, 4 of them are embryonic ages (denoted as E followed by the age in terms of days) and three are postnatal ages (denoted as P followed by the age). In addition, each voxel is annotated to a brain region manually. The number of genes, voxels, and brain regions for each age are summarized in this table.

Ages	E11.5	E13.5	E15.5	E18.5	P4	P14	P28
# of genes	1798	1798	1798	1798	1798	1798	1798
# of voxels	12949	17351	13454	12394	22170	25048	28333
# of regions	159	242	262	89	90	94	222

This work is also related to recent studies on mining from time-evolving data, which is becoming an increasingly important topic. Chakrabarti *et al.* [5] 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 [7], 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 [23], 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 also studied in [34].

The fused Lasso penalty was originally proposed in [30] 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 [31] and social studies [1]. 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 [13], 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 [18], 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 [24].

The formulation proposed in this work is radically different from the evolutionary clustering and matrix factorization formalisms studied in the literature [5, 7, 23, 34]. The differences lie in both the studied problems and in the adopted approaches. Specifically, the existing evolutionary methods deal with clustering problems while our work is concerned with co-clustering problem. Indeed, to the best of our knowledge, our work is the first systematic study of co-clustering on time-varying data. In addition, our work is based on the optimization framework of sparsity-inducing formulations, while the current evolutionary clustering methods is mostly motivated by matrix decomposition techniques.

5. EXPERIMENTAL EVALUATION

5.1 Experimental Setup

We evaluate the proposed evolutionary co-clustering formulation using the Allen Developing Mouse Brain Atlas data, which are publicly available[1]. This data set contains *in situ* hybridization gene expression pattern images in the developing mouse brain across 7 developmental ages (see Figure 2). 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. The statistics of the data are summarized in Table 1. There is one data matrix associated with each of the 7 developing ages. The rows of the matrices correspond to brain voxels while the columns correspond to genes. Note that 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 $\eta = 0$. This is one of the unique advantages of the proposed formulation in which the smoothness constraint can be applied to either or both dimensions. 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.

To measure the co-clustering performance, 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 classi-

[1] `http://developingmouse.brain-map.org/`

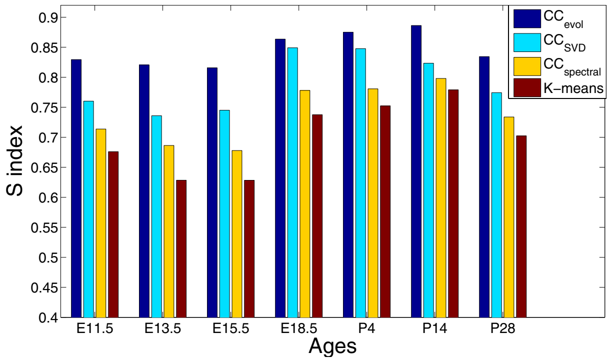

Figure 3: Performance of the proposed method ($\xi = \xi_{\max}$), denoted as $\mathrm{CC}_{\mathrm{evol}}$, in comparison with three other methods measured using the S index. $\mathrm{CC}_{\mathrm{SVD}}$ denotes the co-clustering method based on SVD proposed in [21]; $\mathrm{CC}_{\mathrm{spectral}}$ denotes the spectral co-clustering method proposed in [9]; K-means denotes the K-means method applied to gene expression vectors of each voxel.

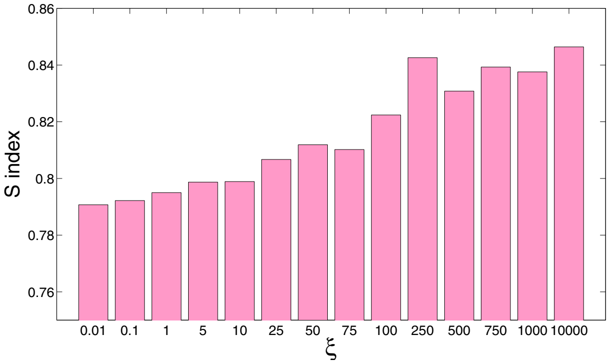

Figure 4: Performance of the proposed method as the value of the fused Lasso regularization parameter ξ increases. The performance is measured using the S index and is averaged across the 7 data sets.

cal neuroanatomy [3]. Following [2, 3], the S index is used to quantitatively measure the correspondence of the clustering results with the classical neuroanatomy reflected in the region annotations. Specifically, let $R = \{r_1, \cdots, r_N\}$ be a partition of the set of brain voxels in which each r_i comprises the set of indices of the voxels that map to that cluster (or anatomical label). The spatial overlap between a region from the annotation and the clustering result is defined as: $P_{ij} = |r_i \cap r_j|/|r_j|$. From the P_{ij} values that are computed over all pairs of brain regions and cluster result, we can then derive a global scalar index of similarity between the two partitions. Since $P_{ij} \neq P_{ji}$, X_{ij} is defined as $X_{ij} = \max\{P_{ij}, P_{ji}\}$ along with $W_{ij} = U_{ij}/\sum U_{ij}$, where $U_{ij} = \min\{|r_i|, |r_j|\}$ if $X_{ij} > 0$ and 0 otherwise. Finally, the S index is defined as $S = 1 - 4\sum_{ij} W_{ij}X_{ij}(1 - X_{ij})$. The S index lies in [0,1], and larger value indicates higher consistency between the clustering results and the annotated regions.

5.2 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 and one clustering method. The two co-clustering methods are the one based on sparse SVD in [21] and the spectral co-clustering method proposed in [9, 36]. We also compare our method with the K-means algorithm when it is applied to cluster the voxels of the data set for each age separately. Note that the evolutionary clustering methods [5, 7, 23] cannot be applied to this data set, since the brain voxels are not registered across ages.

The performance of the four methods on the seven data sets is reported in Figure 3. We observed that the best performance is achieved when $\xi = \xi_{\max}$ and report the results under this parameter setting. Detailed studies on parameter sensitivity are reported in Section 5.3. It can be observed from Figure 3 that the proposed evolutionary co-clustering method outperforms other compared methods consistently across all seven data sets, demonstrating that incorporation of the smoothness constraints between contiguous age data yield improved performance. We can also observe that co-clustering based methods consistently outperform clustering based method. This result is in accordance with the common observation that co-clustering of gene expression data usually leads to improved performance. In addition, the co-clustering method based on sparse SVD outperforms the spectral co-clustering method on all seven data sets.

5.3 Parameter Sensitivity Evaluation

In order to fully understand how the fused Lasso regularization parameter ξ affects the 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 the value for ξ from 0.01 to $\xi_{\max} = 10^4$ and report the performance on each data set in Table 2 and summarize the average performance across data sets in Figure 4. We can observe that the performance improves in general as the value for ξ increases. Indeed, the proposed formulation achieves the highest performance when $\xi = \xi_{\max}$. This demonstrate that incorporation of the fused Lasso regularization is very effective in boosting the performance.

To evaluate the effectiveness of the fused Lasso regularization in encouraging smoothness over the temporal dimension, we again vary ξ from 0.01 to 10^4 and report the ℓ_1-norm differences between temporally adjacent variable vectors in Figure 5. We can observe that, as ξ increases, the values for the fused Lasso regularization terms decrease monotonically until they reach zero, where the adjacent variables are forced to be identical.

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 6. 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. We use the duality gap as the stopping criterion in all experiments, and the error tolerance is set to 10^{-8}.

Table 2: Performance of the proposed method (measured using the S index) on the seven data sets as ξ increases from 0.01 to $\xi_{\mathbf{max}} = 10^4$. The data are publicly available at http://developingmouse.brain-map.org and the statistics are given in Table 1

ages\\ξ	10^{-2}	10^{-1}	1	5	10	25	50	75	100	250	500	750	10^3	10^4
E11.5	0.769	0.766	0.775	0.765	0.772	0.781	0.791	0.801	0.801	0.824	0.814	0.823	0.818	0.829
E13.5	0.736	0.738	0.730	0.730	0.735	0.754	0.757	0.779	0.797	0.815	0.805	0.816	0.818	0.820
E15.5	0.729	0.717	0.730	0.745	0.748	0.760	0.776	0.773	0.784	0.812	0.796	0.812	0.798	0.815
E18.5	0.840	0.843	0.835	0.842	0.848	0.845	0.854	0.839	0.847	0.866	0.842	0.857	0.856	0.863
P4	0.855	0.852	0.843	0.843	0.848	0.863	0.864	0.863	0.868	0.872	0.865	0.871	0.862	0.875
P14	0.807	0.830	0.850	0.838	0.834	0.842	0.837	0.831	0.846	0.873	0.867	0.869	0.877	0.886
P28	0.796	0.797	0.799	0.824	0.804	0.800	0.801	0.782	0.810	0.833	0.823	0.823	0.831	0.834

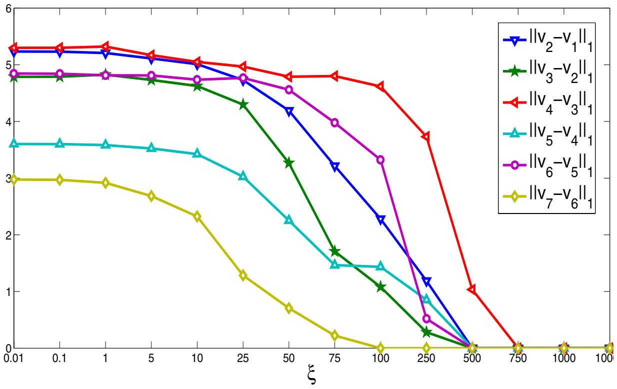

Figure 5: The values of the fused Lasso regularization terms as ξ increases.

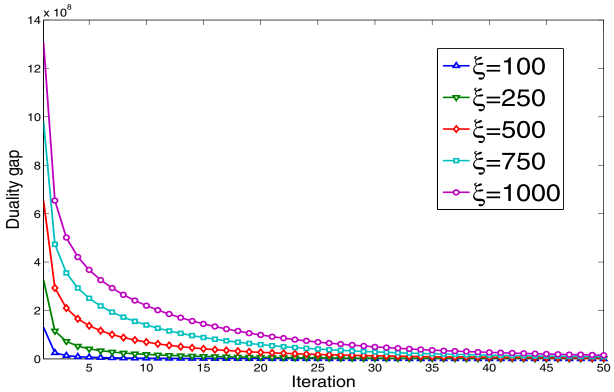

Figure 6: The duality gap in the first 50 iterations for different ξ values.

In all cases, the duality gap is reduced below the tolerance level within a relatively small number of iterations.

6. CONCLUSIONS AND DISCUSSIONS

In this paper, we propose an evolutionary co-clustering method for identifying block structures from time-evolving data. The proposed formulation employs the fused Lasso type of regularization to encourage the smoothness of the block structures, and it is applicable to scenarios in which only one sides of the blocks are required to be temporally smooth. 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 method yields consistently higher performance in comparison to other methods.

It has been shown that nonnegative matrix factorization (NMF) can be used for clustering [11] and co-clustering [35]. However, to the best of our knowledge, NMF has not been employed to perform evolutionary co-clustering. We plan to investigate how NMF [15, 16] can be adapted for co-clustering on time-evolving data. 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 convex 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.

7. ACKNOWLEDGMENTS

We thank the Allen Institute for Brain Science for making the developing mouse brain data available to us. This research is supported by NSF DBI-1147134, NSFC 60905035, and by Old Dominion University.

8. REFERENCES

[1] A. Ahmed and E. P. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106(29):11878–11883, 2009.

[2] J. W. Bohland, H. Bokil, C. B. Allen, and P. P. Mitra. The brain atlas concordance problem: Quantitative comparison of anatomical parcellations. *PLoS ONE*, 4(9):e7200, 09 2009.

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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*, 2004.

[9] 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.

[10] 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.

[11] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 126–135, 2006.

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

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

[14] 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.

[15] N. Guan, D. Tao, Z. Luo, and B. Yuan. Non-negative patch alignment framework. *IEEE Transactions on Neural Networks*, 22(8):1218–1230, 2011.

[16] N. Guan, D. Tao, Z. Luo, and B. Yuan. NeNMF: An optimal gradient method for nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 60(6):2882–2898, 2012.

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

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

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

[20] 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.

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

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

[23] 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.

[24] 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.

[25] 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.

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

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

[28] 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.

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

[30] 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.

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

[32] 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.

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

[34] 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.

[35] J. Yoo and S. Choi. Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on Stiefel manifolds. *Information Processing and Management*, 46(5):559–570, 2010.

[36] 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.