Biometrics 66, 1087–1095
December 2010
DOI: 10.1111/j.1541-0420.2010.01392.x

Biclustering via Sparse Singular Value Decomposition

Mihee Lee,[1] Haipeng Shen,[1,*] Jianhua Z. Huang,[2] and J. S. Marron[1]

[1]Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27599, U.S.A.

[2]Department of Statistics, Texas A&M University, College Station, Texas 77843, U.S.A.

[*]*email:* haipeng@email.unc.edu

Summary. Sparse singular value decomposition (SSVD) is proposed as a new exploratory analysis tool for biclustering or identifying interpretable row–column associations within high-dimensional data matrices. SSVD seeks a low-rank, checkerboard structured matrix approximation to data matrices. The desired checkerboard structure is achieved by forcing both the left- and right-singular vectors to be sparse, that is, having many zero entries. By interpreting singular vectors as regression coefficient vectors for certain linear regressions, sparsity-inducing regularization penalties are imposed to the least squares regression to produce sparse singular vectors. An efficient iterative algorithm is proposed for computing the sparse singular vectors, along with some discussion of penalty parameter selection. A lung cancer microarray dataset and a food nutrition dataset are used to illustrate SSVD as a biclustering method. SSVD is also compared with some existing biclustering methods using simulated datasets.

Key words: Adaptive lasso; Biclustering; Dimension reduction; High-dimension low sample size; Penalization; Principal component analysis.

1. Introduction

High dimensionality has rapidly become a common feature of data encountered in a variety of applications. Sometimes the data are high-dimension low sample size (HDLSS), for example, in fields such as text categorization, medical imaging, and microarray gene expression analysis. HDLSS offers additional statistical challenges as classical multivariate analysis fails in such settings. Unsupervised learning is playing an increasingly important role in exploring such high-dimensional datasets, whose goal is to find interpretable structures in the data.

Biclustering methods refer to a collection of unsupervised learning tools that simultaneously identify distinctive "checkerboard" patterns in data matrices, or sets of rows (or samples) and sets of columns (or variables) in the matrices that are significantly associated. Such methods are becoming increasingly popular in a variety of applications. Madeira and Oliveira (2004) offer a comprehensive survey of existing biclustering algorithms for biological data analysis. Busygin, Prokopyev, and Pardalos (2008) provide a survey of biclustering in data mining from a theoretical perspective and cover a wider range of applications. See also Shabalin et al. (2009) for a more recent development of biclustering methods.

In this article, we introduce *sparse singular value decomposition* (SSVD) as a new tool for biclustering. The application of such a tool is seen in two motivating examples we consider in the article. The first example (Section 2) is a medical application where the data matrix records the microarray gene expressions of 12,625 genes for 56 subjects who have different types of lung cancer. Researchers are interested in identifying groups of coregulated genes for different cancer types. The second example (reported in the online supplement) concerns the nutrition content of 961 different foods, measured by six nutritional variables. We expect that different subgroups of foods can be clustered together based on nutrient content.

Let $\mathbf{X}$ be a $n \times d$ data matrix whose rows may represent samples and columns may represent variables. The singular value decomposition (SVD) of $\mathbf{X}$ can be written as

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \sum_{k=1}^{r} s_k \boldsymbol{u}_k \boldsymbol{v}_k^T, \tag{1}$$

where r is the rank of $\mathbf{X}$, $\mathbf{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r)$ is a matrix of orthonormal left singular vectors, $\mathbf{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r)$ is a matrix of orthonormal right singular vectors, $\mathbf{D} = \mathrm{diag}(s_1, \ldots, s_r)$ is a diagonal matrix with positive singular values $s_1 \geq \cdots \geq s_r$ on its diagonal. SVD decomposes $\mathbf{X}$ into a summation of rank-one matrices $s_k \boldsymbol{u}_k \boldsymbol{v}_k^T$, each of which we call an *SVD layer*. In applications one usually focuses on the SVD layers corresponding to large s_k values. The rest of SVD layers corresponding to small s_ks can often be interpreted as noise and are less useful. If we take the first $K \leq r$ rank-one matrices in the summation in (1), we obtain the following rank-K approximation to $\mathbf{X}$:

$$\mathbf{X} \approx \mathbf{X}^{(K)} \equiv \sum_{k=1}^{K} s_k \boldsymbol{u}_k \boldsymbol{v}_k^T. \tag{2}$$

In fact, $\mathbf{X}^{(K)}$ gives the closest rank-K matrix approximation to $\mathbf{X}$ in the sense that $\mathbf{X}^{(K)}$ minimizes the squared Frobenius norm, i.e.,

$$\mathbf{X}^{(K)} = \underset{\mathbf{X}^* \in \mathcal{A}_K}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{X}^*\|_F^2 = \underset{\mathbf{X}^* \in \mathcal{A}_K}{\operatorname{argmin}} \operatorname{tr}\{(\mathbf{X} - \mathbf{X}^*)(\mathbf{X} - \mathbf{X}^*)^T\}, \tag{3}$$

where $\mathcal{A}_K$ is the set of all $n \times d$ matrices of rank K (Eckart and Young, 1936).

The proposed SSVD seeks a low-rank matrix approximation to $\mathbf{X}$ as that in (2), but with the requirement that the vectors $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ are *sparse*, that is, they have many zero entries. We obtain sparsity by adding sparsity-inducing penalties to the minimization objective in (3). The sparsity property implies that, the rank-one matrix $s_k \boldsymbol{u}_k \boldsymbol{v}_k^T$, now referred to as an *SSVD layer*, clearly has a checkerboard structure. This makes SSVD suitable for biclustering. Specifically, for the kth SSVD layer, those rows (or samples) with nonzero u_{ik}s are naturally clustered together, as well as those columns (or variables) with nonzero v_{jk}s. Hence the kth layer simultaneously links sets of samples and sets of variables together to reveal some desirable sample-variable association. For example, in the lung cancer application of Section 2, the first SSVD layer identifies 3205 genes (out of the original 12,625 genes) that point to a contrast between cancer types.

The rest of the article is organized as follows. We start with an illustration of our SSVD method in Section 2 by analyzing the lung cancer data studied in Liu et al. (2008). We then present the methodological details of SSVD in Section 3. Section 3.1 defines a penalized sum-of-squares criterion for obtaining SSVD layers and makes connections with variable selection methods for penalized regressions; Section 3.2 presents an iterative algorithm for efficient computation of SSVD layers; Section 3.3 gives a data-driven procedure for selecting the penalty parameters; Section 3.4 compares SSVD with the closely related sparse principal component analysis (PCA) method of Shen and Huang (2008); Section 3.5 briefly reviews several other SVD-based biclustering methods and discusses their differences with SSVD. In Section 4, we use simulation studies to evaluate SSVD and compare it with several competing methods. We then conclude in Section 5. Results from additional simulation studies, one real application of analyzing the food nutritional data of Lazzeroni and Owen (2002), and more analysis of the cancer data can be found in the Web Appendix.

2. Lung Cancer Data

In this section, we illustrate SSVD using microarray gene expression data. In general, microarray data are HDLSS, in that the expression levels of thousands of genes, d, are measured simultaneously only for a small number of subjects, n. Gene selection is a fundamental challenge in any analysis of microarray data. The goal is to identify sets of *biologically relevant* genes, for example, that are significantly expressed for certain cancer types. Below SSVD is used to simultaneously select significant genes *and* relevant subjects.

The data consist of expression levels of 12,625 genes, measured from 56 subjects. These subjects are known to be either normal subjects (Normal) or patients with one of the following three types of cancer: pulmonary carcinoid tumors (Carcinoid), colon metastases (Colon), and Small cell carcinoma (SmallCell). The data can be viewed as a $56 \times 12,625$ matrix ($\mathbf{X}$), whose rows represent the subjects, grouped together according to the cancer type, and the columns correspond to the genes. Each column of $\mathbf{X}$ is first centered before the SSVD analysis. (A subset of the data is analyzed by Liu et al. (2008) to illustrate the effectiveness of *SigClust*, a tool for assessing statistical significance of clustering.)

An important goal of our study is to simultaneously identify related gene and subject groups. For example, we are interested in looking for genes that are significantly expressed for certain types of cancer, or that can help distinguish different types of cancer. We want to point out that our SSVD method is an unsupervised learning tool in that it *does not* use the information of the available subject cancer types. We only use the cancer type information *a posterior* to interpret the analysis results and to evaluate the performance of SSVD.

We extract the first three SSVD layers $\mathbf{X}_{(k)} = \hat{s}_k \hat{\boldsymbol{u}}_k \hat{\boldsymbol{v}}_k^T$ sequentially. The reason for considering only three layers is that the first three singular values are much bigger than the rest. The technical details of SSVD will be provided in Section 3. The current analysis uses the algorithm summarized at the end of Section 3.3 that integrates model fitting and penalty parameter selection. Our algorithm usually converges within 5 to 10 iterations. Higher-order layers take longer to estimate as the corresponding eigenvalues are smaller. It takes 1162, 1163, and 2028 seconds for the three layers, respectively, using our `Matlab` program running on a Windows XP desktop with Intel® Core™2 Duo CPU P8700 of a clock speed of 2.53 Gigahertz.

Figure 1 sequentially shows the image plots of the SSVD layers. The panels are plotted using the same color scale shown in the color bar on the right: all entries of the layers are first divided by the maximum absolute value of the entries from all three layers, so that all entries afterward lie between -1 and 1. To better visualize the gene grouping, the columns of the kth layer are rearranged based on an ascending ordering of the entries of $\hat{\boldsymbol{v}}_k$. The dotted horizontal lines in each panel reveal the four cancer types of the subjects. The white vertical area corresponds to those zeroed-out genes, which reveals the effect of the sparsity regularization. (In each panel, 8000 zeroed-out genes are excluded when plotting, and the boundaries of the white areas are indicated.)

The plots illustrate very well the power of SSVD in biclustering the genes and subjects. The first observation is that SSVD automatically performs gene selection; the number of genes selected in each layer is much less than 12,625: there are, respectively, 3205, 2511, and 1221 genes involved in the three layers, corresponding to the nonzero entries of the $\hat{\boldsymbol{v}}_k$ vectors.

Furthermore, the selected genes correspond to informative grouping of the 56 subjects. The plots depict some interesting relations between the gene groups and groups of subjects. For example, the first panel suggests that the significant genes in $\hat{\boldsymbol{v}}_1$ (those nonzero entries) present a strong contrast between Normal and Carcinoid, and a milder contrast between Colon and SmallCell. More specifically, the first 1463 genes in layer 1 are mostly positively expressed for the Carcinoid and SmallCell groups, while negatively expressed for the Normal and Colon groups; on the other hand, the genes ordered between 10,883 and 12,625 show opposite expressions for the two sides of the contrast. These 3205 genes are significantly involved in the grouping of the subjects presented in layer 1.

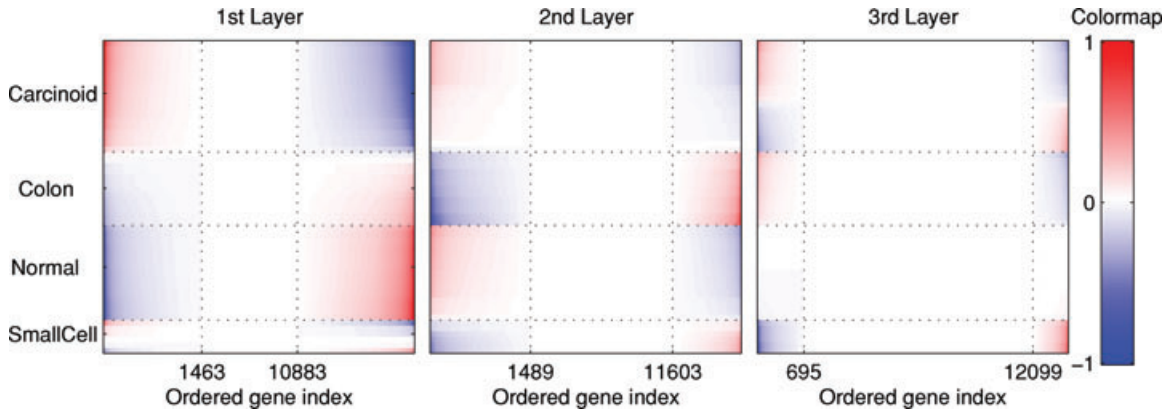

Figure 1. Lung cancer data: image plots of the first three SSVD layers $\hat{s}_k \hat{\boldsymbol{u}}_k \hat{\boldsymbol{v}}_k^T (k = 1, 2, 3)$. In each panel, the genes are rearranged according to an increasing order of the entries of $\hat{\boldsymbol{v}}_k$, and subjects are also rearranged according to the values of $\hat{\boldsymbol{u}}_k$ within each subject group. (In each panel, 8000 genes in the middle white area are excluded when plotting.) This figure appears in color in the electronic version of this article.

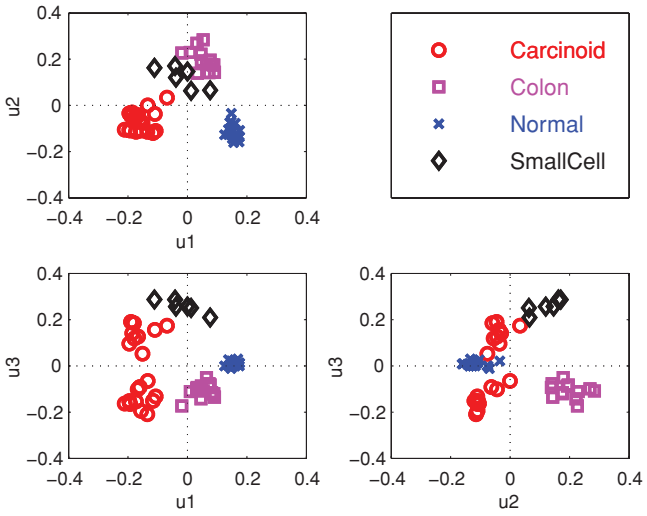

Figure 2. Lung cancer data: scatterplots of the entries of the first three left sparse singular vectors $\hat{\boldsymbol{u}}_k (k = 1, 2, 3)$. This figure appears in color in the electronic version of this article.

Interesting two-way groupings also exist in the second and third layers. Layer 2 shows a contrast between Carcinoid/Normal and Colon/SmallCell, highlighted by 2511 genes; layer 3 zeros out the Normal group and singles out the SmallCell group, using only 1221 genes.

The subject grouping/clustering can also be seen in Figure 2, which shows the scatterplots among the first three sparse left singular vectors $\hat{\boldsymbol{u}}_k$, $k = 1, 2, 3$. (For easier interpretation, the cancer types are plotted using different colors and symbols.) The first two vectors reveal three subject clusters, which coincide with the following cancer type groups: Carcinoid, Colon/SmallCell, and Normal. The next two panels reveal one interesting observation: the Carcinoid patients form two subgroups. Hence, the three vectors provide a good separation of the four cancer types, although the cancer type information is not used in our analysis.

Figure 3 depicts the viewpoint of SSVD as low-rank matrix approximation. As discussed later in Section 3, one can view the first SSVD layer as the "best" sparse rank-one approximation of the raw data matrix, the sum of the first two layers as the "best" sparse rank-two approximation, and so on. Figure 3 shows the raw data matrix and the image plots of the rank-k SSVD approximations for $k = 1, 2, 3$. For better visualization, we rearrange the genes in the following way. Each $\hat{\boldsymbol{v}}_k$ naturally groups the 12,625 genes into three groups according to the sign of the entries in $\hat{\boldsymbol{v}}_k$: negative, zero, positive. Hence, the three vectors $\hat{\boldsymbol{v}}_k (k = 1, 2, 3)$ result in 27 groups of the genes. The genes in the same group are then plotted together, and different gene groups are separated by the dotted vertical lines. (Only the first 5000 genes are plotted to better reveal the structure; the additional genes are nonsignificant based on the three $\hat{\boldsymbol{v}}_k$s.) The SSVD low-rank approximations reveal interesting checkerboard structures caused by the gene and subject grouping, and in addition the rank-three approximation (the lower-left panel) highlights very well the underlying structure of the raw data.

We now provide an illustration of how the above gene grouping can be used to derive some gene expression profile for the cancers. We first ignore the gene group that the entries of $\hat{\boldsymbol{v}}_k$ are always zero for $k = 1, 2, 3$. For each subject, we calculate the mean of the raw expression levels of the genes within each gene group, which results in a 26-dimensional mean expression vector for the subject. The left panel of Figure 4 plots the mean expression vectors of the 56 subjects, where the horizontal axis labels the 26 gene groups and the cancer types are indicated using different colors. Within each cancer type, we then average the mean expression vectors over the relevant subjects, and obtain a 26-dimensional mean expression profile, which is plotted in the right panel of Figure 4. There appears to be some clear difference among the four expression profiles. For example, only Carcinoid patients are positive for the first gene group, and only Normal patients are positive for the 11th and 12th gene groups. The results suggest that it is interesting to combine SSVD with some classifiers in an elaborate out-of-sample classification study, which, however, goes beyond the scope of the present article.

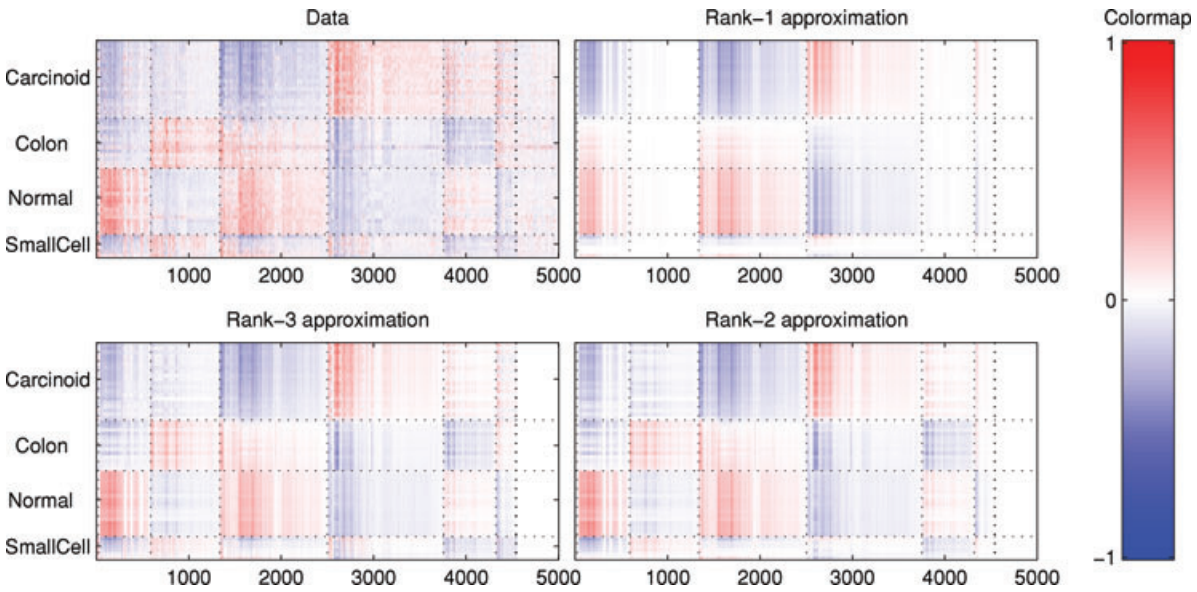

Figure 3. Lung cancer data: comparison of the raw data matrix and the best SSVD rank-k approximations ($k = 1, 2, 3$). (Only the first 5000 genes are plotted to better reveal the details.) This figure appears in color in the electronic version of this article.

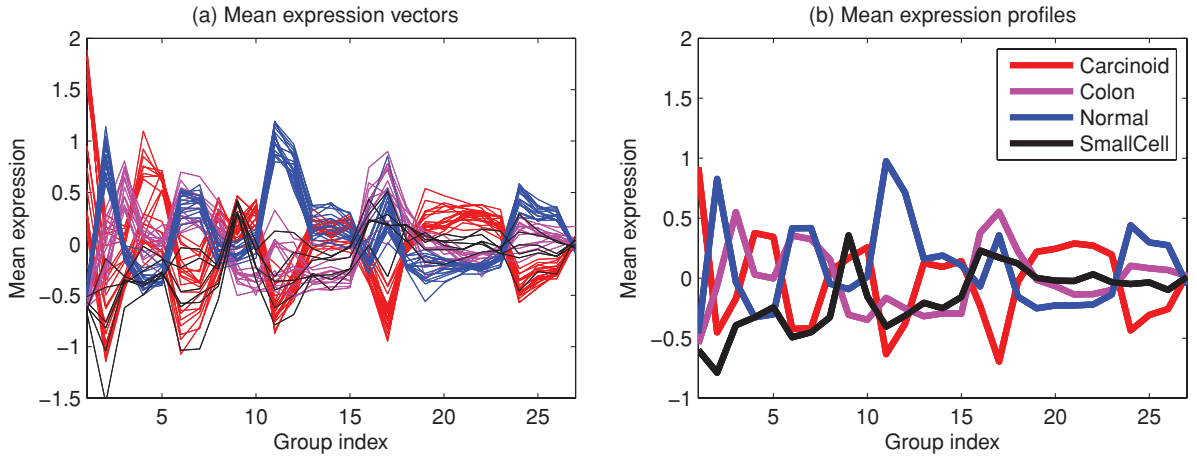

Figure 4. Lung cancer data: (a) Example mean expression vectors of the subjects. (b) Example mean expression profiles of the cancer types. This figure appears in color in the electronic version of this article.

We also analyze the data using two existing biclustering algorithms—Plaid (Lazzeroni and Owen, 2002) and RoBiC (Asgarian and Greiner, 2008). Technical details of the algorithms can be found in Section 3.5. Both methods yield less meaningful biclusters than SSVD especially in subject grouping, and give worse low-rank approximation of the data. Detailed results are presented in the Web Appendix through plots analogous to Figures 1 to 3.

3. The Method

This section presents the technical details of our SSVD procedure. Our presentation focuses on extracting the first SSVD layer; subsequent layers can be extracted sequentially from the residual matrices after removing the preceding layers.

3.1 *A Penalized Sum-of-Squares Criterion*

We note that the first SVD layer $s_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$ is the best rank-one matrix approximation of $\mathbf{X}$ under the Frobenius norm, i.e.,

$$(s_1, \boldsymbol{u}_1, \boldsymbol{v}_1) = \underset{s, \boldsymbol{u}, \boldsymbol{v}}{\operatorname{argmin}} \|\mathbf{X} - s\boldsymbol{u}\boldsymbol{v}^T\|_F^2, \tag{4}$$

where s is a positive scalar, $\boldsymbol{u}$ is a unit n-vector, and $\boldsymbol{v}$ is a unit d-vector. To obtain sparse vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, we propose to add sparsity-inducing penalties on $\boldsymbol{u}$ and $\boldsymbol{v}$ in the optimization objective in (4). Specifically, we minimize with respect to the triplet $(s, \boldsymbol{u}, \boldsymbol{v})$ the following penalized sum-of-squares criterion,

$$\|\mathbf{X} - s\boldsymbol{u}\boldsymbol{v}^T\|_F^2 + \lambda_u P_1(s\boldsymbol{u}) + \lambda_v P_2(s\boldsymbol{v}), \tag{5}$$

where $P_1(s\boldsymbol{u})$ and $P_2(s\boldsymbol{v})$ are sparsity-inducing penalty terms whose forms will be given later, and λ_u and λ_v are two

nonnegative penalty parameters that balance the goodness-of-fit measure $\|\mathbf{X} - s\boldsymbol{u}\boldsymbol{v}^T\|_F^2$ and the penalty terms. Two penalty parameters are used so that different levels of sparsity can be imposed on $\boldsymbol{u}$ and $\boldsymbol{v}$. This is a beneficial flexibility, yet little difficulty is added in terms of multiple parameter selection, as shown in Section 3.3. When $\lambda_u = \lambda_v = 0$, the criterion (5) reduces to (4), which obtains the plain SVD layers.

To motivate appropriate sparsity-inducing penalty terms in (5), we make use of the idea of lasso regression (Tibshirani, 1996). For fixed $\boldsymbol{u}$, minimization of (5) with respect to $(s, \boldsymbol{v})$ is equivalent to minimization with respect to $\tilde{\boldsymbol{v}} = s\boldsymbol{v}$ of

$$\|\mathbf{X} - \boldsymbol{u}\tilde{\boldsymbol{v}}^T\|_F^2 + \lambda_v P_2(\tilde{\boldsymbol{v}}) = \|\mathbf{Y} - (\mathbf{I}_d \otimes \boldsymbol{u})\,\tilde{\boldsymbol{v}}\|^2 + \lambda_v P_2(\tilde{\boldsymbol{v}}), \quad (6)$$

where $\mathbf{Y} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_d^T)^T \in R^{nd}$ with $\boldsymbol{x}_j$ being the jth column of $\mathbf{X}$, and $\otimes$ being the Kronecker product. The right-hand side of (6) is the minimization criterion of a penalized regression with the response variable $\mathbf{Y}$, the design matrix $\mathbf{I}_d \otimes \boldsymbol{u}$, and the regression coefficient $\tilde{\boldsymbol{v}}$. This connection with penalized regression naturally suggests use of the lasso penalty $P_2(\tilde{\boldsymbol{v}}) = \sum_{j=1}^d |\tilde{v}_j|$ in (6). Similarly, for fixed $\boldsymbol{v}$, minimization of (5) with respect to $(s, \boldsymbol{u})$ is equivalent to minimization with respect to $\tilde{\boldsymbol{u}} = s\boldsymbol{u}$ of

$$\|\mathbf{X} - \tilde{\boldsymbol{u}}\boldsymbol{v}^T\|_F^2 + \lambda_v P_1(\tilde{\boldsymbol{u}}) = \|\mathbf{Z} - (\mathbf{I}_n \otimes \boldsymbol{v})\,\tilde{\boldsymbol{u}}\|^2 + \lambda_u P_1(\tilde{\boldsymbol{u}}), \quad (7)$$

where $\mathbf{Z} = (\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)})^T \in R^{nd}$, with $\boldsymbol{x}_{(i)}^T$ being the ith row of $\mathbf{X}$. We can use the lasso penalty $P_1(\tilde{\boldsymbol{u}}) = \sum_{i=1}^n |\tilde{u}_i|$ in (7). It is important to point out that the quantities that enter the lasso penalty are $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{v}}$, not $\boldsymbol{u}$ and $\boldsymbol{v}$. The reason is that $\boldsymbol{u}$ and $\boldsymbol{v}$ are unit vectors and thus subject to scale constraints, which in turn will invalidate the use of the lasso penalty.

In this article, we consider a broader class of penalties called *adaptive lasso penalties*, as suggested by Zou (2006) and used by Zhang and Lu (2007) and Wang, Li, and Tsai (2007),

$$P_1(s\boldsymbol{u}) = s\sum_{i=1}^n w_{1,i}|u_i| \quad \text{and} \quad P_2(s\boldsymbol{v}) = s\sum_{j=1}^d w_{2,j}|v_j|, \quad (8)$$

where the $w_{1,i}$s and $w_{2,j}$s are possibly data-driven weights. When $w_{1,i} = w_{2,j} = 1$ for every i and j, we obtain the lasso penalty. Following Zou (2006), the weights $w_{2,j}$s can be chosen as

$$\boldsymbol{w}_2 \equiv (w_{2,1}, \ldots, w_{2,d})^T = |\hat{\tilde{\boldsymbol{v}}}|^{-\gamma_2},$$

where γ_2 is a known nonnegative parameter, $|\hat{\tilde{\boldsymbol{v}}}|^{-\gamma_2}$ is defined as an operation to each component of the vector $\hat{\tilde{\boldsymbol{v}}}$, and $\hat{\tilde{\boldsymbol{v}}}$ is the ordinary least squares (OLS) estimate of $\tilde{\boldsymbol{v}}$, which in this case is

$$\begin{aligned}&\left\{(\mathbf{I}_d \otimes \boldsymbol{u})^T(\mathbf{I}_d \otimes \boldsymbol{u})\right\}^{-1}(\mathbf{I}_d \otimes \boldsymbol{u})^T\mathbf{Y}\\ &\quad = \left(\boldsymbol{u}^T\boldsymbol{x}_1, \ldots, \boldsymbol{u}^T\boldsymbol{x}_d\right)^T = \mathbf{X}^T\boldsymbol{u}\end{aligned}$$

(see (6) for the forms of the responses and the design matrix of this regression). There are some natural candidates for γ_2; for example, $\gamma_2 = 0$ corresponds to the lasso fit, and $\gamma_2 = 1$ is similar to the nonnegative garotte (Breiman, 1995). Zou (2006) also suggests using $\gamma_2 = 0.5$ and 2. The Bayesian information criterion (BIC) criterion to be introduced in Section 3.3 for selecting the penalty parameters can be used to select an appropriate γ_2 from a finite set of candidate values. Similarly, $\boldsymbol{w}_1 = (w_{1,i}, \ldots, w_{1,n})^T$ can be chosen as $\boldsymbol{w}_1 = \left|\hat{\tilde{\boldsymbol{u}}}\right|^{-\gamma_1}$, where $\hat{\tilde{\boldsymbol{u}}}$ is the OLS estimator of $\tilde{\boldsymbol{u}}$, which is

$$\left\{(\mathbf{I}_n \otimes \boldsymbol{v})^T(\mathbf{I}_n \otimes \boldsymbol{v})\right\}^{-1}(\mathbf{I}_n \otimes \boldsymbol{v})^T\mathbf{Z} = \left(\boldsymbol{x}_{(1)}^T\boldsymbol{v}, \ldots, \boldsymbol{x}_{(n)}^T\boldsymbol{v}\right)^T = \mathbf{X}\boldsymbol{v},$$

and γ_1 is a nonnegative weight parameter, which can be chosen similarly as γ_2.

3.2 *An Iterative Algorithm*

The connection of SSVD to penalized regression as given in (6) and (7) suggests alternately solving the regressions (6) and (7) using the algorithm by Zou (2006) to obtain the SSVD layers. However, these regressions are of very large dimension. In this subsection, we provide a much more efficient iterative algorithm that effectively utilizes the special structure of SSVD. The steps of this algorithm consist of some simple component-wise thresholding rules.

With the adaptive lasso penalty, the minimizing objective (5) for SSVD can be written as

$$\|\mathbf{X} - s\boldsymbol{u}\boldsymbol{v}^T\|_F^2 + s\lambda_u\sum_{i=1}^n w_{1,i}|u_i| + s\lambda_v\sum_{j=1}^d w_{2,j}|v_j|, \quad (9)$$

where s is a positive scalar, $\boldsymbol{u}$ is a unit n-vector, and $\boldsymbol{v}$ is a unit d-vector. We alternately minimize (9) with respect to $\boldsymbol{u}$ and $\boldsymbol{v}$. For fixed $\boldsymbol{u}$, minimizing (9) is equivalent to minimizing

$$\begin{aligned}&\|\mathbf{X} - \boldsymbol{u}\tilde{\boldsymbol{v}}^T\|_F^2 + \lambda_v\sum_{j=1}^d w_{2,j}|\tilde{v}_j|\\ &\quad = \|\mathbf{X}\|_F^2 + \sum_{j=1}^d\left\{\tilde{v}_j^2 - 2\tilde{v}_j(\mathbf{X}^T\boldsymbol{u})_j + \lambda_v w_{2,j}|\tilde{v}_j|\right\}, \quad (10)\end{aligned}$$

where $\tilde{\boldsymbol{v}} = s\boldsymbol{v}$. Note that we can minimize (10) with respect to each $\tilde{v}$ separately. The following lemma gives a closed-form solution to such minimization problems.

LEMMA 1. *The minimizer of $\beta^2 - 2y\beta + 2\lambda|\beta|$ is $\hat{\beta} = sign(y)(|y| - \lambda)_+$. This is a simple soft-thresholding rule: if $y > \lambda$, then $\hat{\beta} = y - \lambda$; if $y < -\lambda$, then $\hat{\beta} = y + \lambda$; otherwise, $\hat{\beta} = 0$.*

Taking y to be the jth component of $\mathbf{X}^T\boldsymbol{u}$ and letting $\lambda = \lambda_v w_{2,j}/2$ in Lemma 1, we obtain that the minimizing v_j of (10) is $\tilde{v}_j = sign\{(\mathbf{X}^T\boldsymbol{u})_j\}(|(\mathbf{X}^T\boldsymbol{u})_j| - \lambda_v w_{2,j}/2)_+$. Then we separate out the scaling by letting $s = \|\tilde{\boldsymbol{v}}\|$ and $\boldsymbol{v} = \tilde{\boldsymbol{v}}/s$.

Similarly, for fixed $\boldsymbol{u}$, minimizing (9) is equivalent to minimizing

$$\begin{aligned}&\|\mathbf{X} - \tilde{\boldsymbol{u}}\boldsymbol{v}^T\|_F^2 + \lambda_u\sum_{i=1}^n w_{1,i}|\tilde{u}_i|\\ &\quad = \|\mathbf{X}\|_f^2 + \sum_{i=1}^n\left\{\tilde{u}_i^2 - 2\tilde{u}_i(\mathbf{X}\boldsymbol{v})_i + \lambda_u w_{1,i}|\tilde{u}_i|\right\}, \quad (11)\end{aligned}$$

where $\tilde{\boldsymbol{u}} = s\boldsymbol{u}$. We apply Lemma 1 again to obtain that the optimizing $\tilde{u}$ is $\tilde{u}_i = sign\{(\mathbf{X}\boldsymbol{v})_i)\}(|(\mathbf{X}\boldsymbol{v})_i| - \lambda_u w_{1,i}/2)_+$. Then we separate out the scaling by letting $s = \|\tilde{\boldsymbol{u}}\|$ and $\boldsymbol{u} = \tilde{\boldsymbol{u}}/s$. The minimization of (9) with respect to $\boldsymbol{v}$ and $\boldsymbol{u}$ is iterated until convergence.

After convergence, we set $s = \boldsymbol{u}^T \mathbf{X} \boldsymbol{v}$ and the SSVD layer is given by $s\boldsymbol{u}\boldsymbol{v}^T$. The next SSVD layer can be obtained by applying the same procedure to the residual matrix $\mathbf{X} - s\boldsymbol{u}\boldsymbol{v}^T$. The process is repeated until enough numbers of SSVD layers are obtained.

3.3 *Penalty Parameter Selection*

We define the *degree of sparsity* of the sparse singular vector $\boldsymbol{u}$ as the number of zero components in $\boldsymbol{u}$ and similarly for $\boldsymbol{v}$. The degrees of sparsity of $\boldsymbol{u}$ and $\boldsymbol{v}$ are closely related to the two penalty parameters λ_u and λ_v presented in (5) and (9). In light of (10) and the subsequent discussion, for fixed $\boldsymbol{u}$, the degree of sparsity of $\boldsymbol{v}$ is the number of $(\mathbf{X}^T \boldsymbol{u})_j$s that are bigger than $\lambda_v w_{2j}/2$. In other words, the degree of sparsity of $\boldsymbol{v}$ is a step function of the penalty parameter λ_v. The same holds between the degree of sparsity of $\boldsymbol{u}$ and λ_u. Therefore, selecting the parameters λ_u and λ_v is equivalent to selecting the degrees of sparsity.

Zou, Hastie, and Tibshirani (2007) show that, for lasso regression, the number of nonzero coefficients provides an unbiased estimate for the degree of freedom of the lasso fit, and suggest that the BIC (Schwarz, 1978) can be used to select the optimal number of nonzero coefficients. We apply this result to our setting for selecting the degrees of sparsity by making use of the connection of SSVD to penalized regression as given in (6) and (7). For the penalized regression (6) with fixed $\boldsymbol{u}$, define

$$\mathrm{BIC}(\lambda_v) = \frac{\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2}{nd \cdot \hat{\sigma}^2} + \frac{\log(nd)}{nd} \hat{df}(\lambda_v), \qquad (12)$$

where $\hat{df}(\lambda_v)$ is the degree of sparsity of $\boldsymbol{v}$ with λ_v as the penalty parameter, and $\hat{\sigma}^2$ is the OLS estimate of the error variance from the model (6). For the penalized regression (7) with fixed $\boldsymbol{v}$, define

$$\mathrm{BIC}(\lambda_u) = \frac{\|\mathbf{Z} - \hat{\mathbf{Z}}\|^2}{nd \cdot \hat{\sigma}^2} + \frac{\log(nd)}{nd} \hat{df}(\lambda_u), \qquad (13)$$

where $\hat{df}(\lambda_u)$ is the degree of freedom of $\boldsymbol{u}$ with λ_u as the penalty parameter, and $\hat{\sigma}^2$ is the OLS estimate of the error variance from the model (7).

The BICs defined above are conditional in nature. We use them by nesting the penalty parameter selection within the iterative algorithm given in the previous subsection. The conditional parameter selection prevents using the computationally more expensive simultaneous selection of two parameters. The iterative SSVD procedure that combines model fitting and penalty parameter selection is summarized below. Our experience from simulation studies and real applications suggests that the iterative algorithm typically converges within 5 to 10 iterations; the convergence of the selected penalty parameters is even faster.

The SSVD Algorithm

Step 1. Apply the standard SVD to $\mathbf{X}$. Let $\{s_{old}, \boldsymbol{u}_{old}, \boldsymbol{v}_{old}\}$ denote the first SVD triplet.

Step 2. Update:

(a) Set $\tilde{v}_j = sign\{(\mathbf{X}^T \boldsymbol{u}_{old})_j\}(|(\mathbf{X}^T \boldsymbol{u}_{old})_j| - \lambda_v w_{2,j}/2)_+$, $j = 1, \ldots, d$, where λ_v is the minimizer of $\mathrm{BIC}(\lambda_v)$ defined in (12). Let $\tilde{\boldsymbol{v}} = (\tilde{v}_1, \ldots, \tilde{v}_d)^T$, $s = \|\tilde{\boldsymbol{v}}\|$, and $\boldsymbol{v}_{new} = \tilde{\boldsymbol{v}}/s$.

(b) Set $\tilde{u}_i = sign\{(\mathbf{X}\boldsymbol{v}_{new})_i\}(|(\mathbf{X}\boldsymbol{v}_{new})_i| - \lambda_u w_{1,i}/2)_+$, $i = 1, \ldots, n$, where λ_u is the minimizer of $\mathrm{BIC}(\lambda_u)$ defined in (13). Let $\tilde{\boldsymbol{u}} = (\tilde{u}_1, \ldots, \tilde{u}_n)^T$, $s = \|\tilde{\boldsymbol{u}}\|$, and $\boldsymbol{u}_{new} = \tilde{\boldsymbol{u}}/s$.

(c) Set $\boldsymbol{u}_{old} = \boldsymbol{u}_{new}$ and repeat Steps 2(a) and 2(b) until convergence.

Step 3. Set $\boldsymbol{u} = \boldsymbol{u}_{new}, \boldsymbol{v} = \boldsymbol{v}_{new}, s = \boldsymbol{u}_{new}^T \mathbf{X} \boldsymbol{v}_{new}$ at convergence.

3.4 *Connection with Sparse Principal Component Analysis*

PCA is one of the most commonly used unsupervised learning tools (Jolliffe, 2002), especially for dimension reduction when analyzing high-dimensional data. PCA identifies a small number of linear combinations of the original variables, or principal components (PCs), that can explain most of the variation in the data. To improve the interpretability of the PCs, sparse PCA (SPCA) methods have been proposed to yield sparse PC loading vectors with many zero loadings. For example, see Jolliffe, Trendafilov, and Uddin (2003); Zou, Hastie, and Tibshirani (2006); and Leng and Wang (2009). Using the relationship between SVD and PCA, Shen and Huang (2008) propose to obtain the first sparse PC by minimizing

$$\|\mathbf{X} - \boldsymbol{u}\tilde{\boldsymbol{v}}^T\|_F^2 + \lambda P(\tilde{\boldsymbol{v}}) \quad \text{subject to} \quad \|\boldsymbol{u}\| = 1, \qquad (14)$$

and the subsequent sparse PCs by applying the same procedure to the residual matrices. It is easy to see that the SPCA criterion of Shen and Huang (2008) is a special case of our SSVD criterion (5). In fact, the SPCA criterion can be obtained by setting $\tilde{\boldsymbol{v}} = s\boldsymbol{v}$ with $s = \|\tilde{\boldsymbol{v}}\|$ and $\|\boldsymbol{v}\| = 1$, $\lambda_u = 0$, $\lambda_v = \lambda$, and $P_1(\cdot) = P(\cdot)$ in (5). However, it is important to note that SSVD and SPCA have different objectives and thus are different procedures. SSVD targets to detect block structures in the data matrix and thus is a tool for biclustering, while SPCA aims at identifying sparse PCs that can describe most data variation and thus is a tool mainly for dimension reduction, not for biclustering. Results on application of SPCA to simulated examples clearly show that SPCA is not suitable for biclustering (Section 4.1).

3.5 *Other SVD Related Biclustering Methods*

SSVD offers an SVD-based method to perform biclustering. We compare it with two existing biclustering methods, Plaid (Lazzeroni and Owen, 2002) and RoBiC (Asgarian and Greiner, 2008). Both methods assume that the data matrix can be approximated by a structure that is similar to that provided by the SVD (3).

Plaid assumes that $X_{ij} = \sum_{k=1}^K \theta_{ijk}\rho_{ik}\kappa_{jk} = \sum_{k=1}^K (\mu_k + \alpha_{ik} + \beta_{jk})\rho_{ik}\kappa_{jk}$, where K is the number of layers (or biclusters), ρ_{ik} is 1 if row i is in the kth bicluster (zero otherwise), κ_{jk} is 1 if column j is in the kth bicluster (zero otherwise), and θ_{ijk} specifies the contribution of the kth bicluster. The two-way *additive* model on θ_{ijk} estimates the effects of row i and column j. Plaid also employs an iterative procedure to obtain the layers, and the parameters are estimated by maximizing the reduction in the sum of squares. As a comparison, SSVD assumes a *multiplicative* structure within the kth bicluster; it directly balances the goodness-of-fit

measure with the sparsity, and connects naturally with the variable selection literature.

RoBiC starts with the best rank-1 approximation of the data matrix obtained from applying SVD. To obtain the first layer of biclustering, it looks at the first pair of singular vectors. The entries of the first left singular vector are ordered decreasingly in absolute value and then are fit by a hinge function (two connecting line segments). The indices to the left of the joint of the best-fitting hinge function are included in the first row cluster. The same procedure applies to the first right singular vector to yield the first column cluster. The rows and columns selected form the first bicluster. RoBiC then subtracts the values of the first bicluster from the data matrix, and repeats the same process on the residual data matrix to find the next bicluster, etc. As one can see, RoBiC uses an ad-hoc procedure to select the biclusters, and the hinge function model imposes an unnecessarily strict constraint that might limit the types of sparse structures to be detected. This is confirmed by our simulation studies in Section 4. SSVD is much more rigorous and allows more general sparse structures.

Several other biclustering methods are also based on SVD, such as the coclustering algorithm of Dhillon, Mallela, and Modha (2003); the spectral method of Kluger et al. (2003); the double conjugated clustering by Busygin, Prokopyev, and Pardalos (2005); and the hierarchical clustering of Yang, Dai, and Yan (2007). See Busygin et al. (2008) for a survey on these methods. The hierarchical clustering of Yang et al. (2007) is more akin to SSVD, which involves first extracting the few leading singular vectors, before applying hierarchical clustering *separately* to the left and right singular vectors. Note that the sparsity is not incorporated when the singular vectors are obtained, and clustering is not performed together on the samples and the variables.

4. Simulation Studies

In this section, we report two simulation studies to investigate the performance of SSVD and to compare it with the standard SVD, and two SVD-related biclustering methods—Plaid and RoBiC. We also compare SSVD with the SPCA procedure of Shen and Huang (2008) that only imposes sparse structure on one direction, denoted as either SPCA($\boldsymbol{u}$) or SPCA($\boldsymbol{v}$). The advantage of SSVD over SPCA is in discovering block structures in data matrices. Additional simulation studies can be found in the Web Appendix.

4.1 *Case 1: Rank-1 Approximation*

For the first simulation study, we consider a rank-1 true signal matrix $\mathbf{X}^*$. In particular, let $\mathbf{X}^* = s\boldsymbol{u}\boldsymbol{v}^T$ be a 100×50 rank-1 matrix with $s = 50$ and

$$\tilde{\boldsymbol{u}} = [10, 9, 8, 7, 6, 5, 4, 3, r(2, 17), r(0, 75)]^T, \quad \boldsymbol{u} = \tilde{\boldsymbol{u}}/||\tilde{\boldsymbol{u}}||,$$
$$\tilde{\boldsymbol{v}} = [10, -10, 8, -8, 5, -5, r(3, 5), r(-3, 5), r(0, 34)]^T, \quad \boldsymbol{v} = \tilde{\boldsymbol{v}}/||\tilde{\boldsymbol{v}}||,$$

where $r(a, b)$ denotes a vector of length b, whose entries are all a. Here $\boldsymbol{u}$ and $\boldsymbol{v}$ contain 25 and 16 nonzero entries, respectively. A data matrix $\mathbf{X}$ is generated as the sum of $\mathbf{X}^*$ and the noise matrix $\boldsymbol{\epsilon}$, whose elements are randomly sampled from the standard normal distribution. The simulation is repeated 100 times. The nonzero entries of $\mathbf{X}^*$ take on several distinct values, some of which are quite small. This makes the model estimation more challenging.

For SSVD, we use BIC to choose the degree of sparsity in each updating step, and use weight parameters $\gamma_1 = \gamma_2 = 2$ in deciding the adaptive weight vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. Such choice of the weight parameters has been used in Zou (2006) and is also suggested by a simulation study that is reported in the Web Appendix. For Plaid, we use the most flexible model discussed in Lazzeroni and Owen (2002) and choose the tuning parameters as suggested there. For SPCA($\boldsymbol{u}$) and SPCA($\boldsymbol{v}$), we also set the adaptive lasso weight parameter to be two, and select the penalty parameter using BIC.

Table 1 reports the estimation results, in terms of the average number of zero elements in the estimated 100 singular vectors in both directions (column 1), the average number (and proportion) of correctly identified zeros (column 2), the average number (and proportion) of correctly identified nonzeros (column 3), and the misclassification rate (column 4).

As one can see, SSVD performs much better than the competitors. For example, in terms of correctly identifying the true zero and nonzero entries, on average it only misclassifies 1.01% and 0.24% of the entries in $\boldsymbol{u}$ and $\boldsymbol{v}$, respectively, as highlighted in Table 1.

Table 1
Case 1*: Comparison of the performance among SSVD, RoBiC, Plaid, SVD, SPCA($\boldsymbol{u}$), and SPCA($\boldsymbol{v}$)*

		Avg. # of zeros (true)	Avg. # of correctly identified zeros	Avg. # of correctly identified nonzeros	Misclassification rate
SSVD	$\boldsymbol{u}$	74.73 (75)	74.36 (99.15%)	24.63 (98.52%)	**1.01**%
	$\boldsymbol{v}$	33.88 (34)	33.88 (99.65%)	16.00 (100.0%)	**0.24**%
RoBiC	$\boldsymbol{u}$	90.60 (75)	75.00 (100.0%)	9.40 (37.60%)	15.60%
	$\boldsymbol{v}$	41.46 (34)	33.98 (99.94%)	8.52 (53.25%)	15.00%
Plaid	$\boldsymbol{u}$	90.65 (75)	75.00 (100.0%)	9.35 (37.40%)	15.65%
	$\boldsymbol{v}$	20.19 (34)	17.93 (52.74%)	13.74 (85.88%)	36.66%
SVD	$\boldsymbol{u}$	0.00 (75)	0.00 (0.00%)	25.00 (100.0%)	75.00%
	$\boldsymbol{v}$	0.00 (34)	0.00 (0.00%)	16.00 (100.0%)	68.00%
SPCA($\boldsymbol{u}$)	$\boldsymbol{u}$	74.11 (75)	73.84 (98.45%)	24.73 (98.92%)	1.43%
	$\boldsymbol{v}$	0.00 (34)	0.00 (0.00%)	16.00 (100.0%)	68.00%
SPCA($\boldsymbol{v}$)	$\boldsymbol{u}$	0.00 (75)	0.00 (0.00%)	25.00 (100.0%)	75.00%
	$\boldsymbol{v}$	33.79 (34)	33.79 (99.38%)	16.00 (100.0%)	0.42%

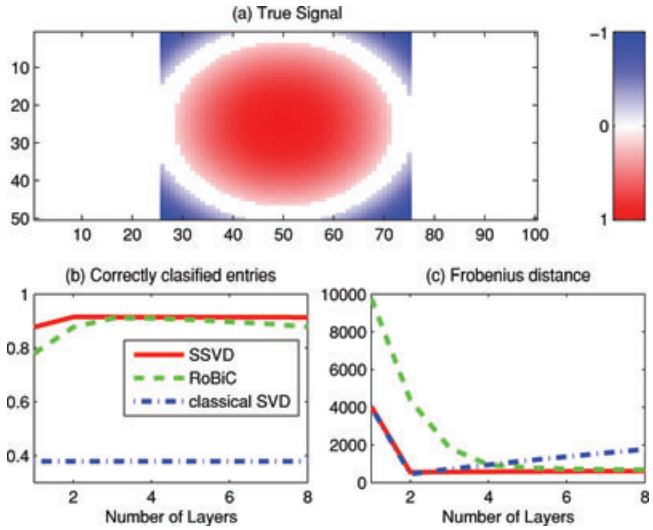

Figure 5. Case 2: (a) Image plot of the true signal matrix $\mathbf{X}^*$. (b) Proportion of the correctly identified entries. In this plot, the vertical axis shows the proportion of correctly classified entries, and the horizontal axis indicates the number of layers extracted by the various methods. (c) Frobenius distance between the true signal matrix and its estimates obtained by the three methods. This figure appears in color in the electronic version of this article.

Not surprisingly, SVD has the most trouble in detecting the underlying sparse structure. As for RoBiC, it detects too many zeroes; the problem is that the hinge model used by RoBiC has trouble separating the small (in magnitude) nonzero entries from being zero; for example, the entries of 2 in $\tilde{\boldsymbol{u}}$ or 3 and -3 in $\tilde{\boldsymbol{v}}$. The same holds true for Plaid. As for the two SPCA methods, they work fine for the penalized direction; however, they fail completely to detect the sparse structure for the unpenalized direction. This shows the advantage of SSVD over SPCA and the necessity of imposing two-way sparse structure when needed.

4.2 *Case 2: Higher Rank Approximation*

In the second simulation study, the true signal matrix $\mathbf{X}^*$ is a 50 by 100 matrix whose elements are given by $X^*_{i,j} = T_{i,j}\mathbf{1}(|T_{i,j}| > 1)$, where

$$T_{i,j} = \begin{cases} \{24^2 - (i-25)^2 - (j-50)^2\}/100, & \text{if } 26 \le j \le 75, \\ 0, & \text{otherwise.} \end{cases}$$

The image plot of $\mathbf{X}^*$ (Panel (a) of Figure 5) highlights an interesting but complex structure, where the positive entries are red, the negative ones are blue, and the zeros are white.

This setup is more complicated for the following reasons. Firstly, as evident in Figure 5, the true signal does not have a multiplicative structure, which is the underlying model assumed by SSVD; nor does $\mathbf{X}^*$ have an additive structure. Secondly, $\mathbf{X}^*$ is *almost* rank-2 in that its eigenvalues are almost zero except the first two, even though its true rank is 50.

The simulation is repeated 100 times. For each simulation run, a data matrix $\mathbf{X}$ is generated as the sum of $\mathbf{X}^*$ and a noise matrix $\boldsymbol{\epsilon}$, whose entries are randomly sampled from the standard normal distribution. We consider only SSVD, RoBiC, and the classical SVD in the current study. (SPCA is clearly not suitable for biclustering as confirmed in the previous study. Plaid is not considered here for two reasons: first, the result of Plaid tends to be similar to [or worse than] that of RoBiC; second, the existing packages for Plaid either lack an automatic procedure to output the estimation results or only output the locations of the detected biclusters without estimating the Plaid model.)

Panel (b) of Figure 5 plots the proportion of the correctly identified entries (both zero and nonzero) by each method as a function of the number of layers extracted, averaged over the 100 simulation runs. As shown in the plot, SSVD correctly identifies the highest proportion of the entries no matter how many layers are used, and the proportion increases as the number of layers increases; in addition, two layers (or a rank-2 approximation) seem to be sufficient as the proportion stabilizes afterward. On the other hand, SVD can only correctly identify the nonzero entries in $\mathbf{X}$; as for RoBiC, three or four layers are needed to peak the proportion of the correctly identified entries, and when using more layers, its detection performance starts to deteriorate as it starts to fit the noise.

Panel (c) of Figure 5 plots the Frobenius distance between the true signal matrix and its estimate, as the number of layers used in the estimation increases. The Frobenius distance measures how close each estimator is to the truth. We can see that SSVD results in the closest estimator uniformly for the numbers of layers considered. SVD performs similar to SSVD initially, before it starts to overfit the data. RoBiC results in a much larger distance than SSVD, and it needs more layers to reach the same level as SSVD.

5. Conclusion and Discussion

In this article, we modified the SVD of a data matrix by imposing sparsity on the left and right singular vectors. The sparsity implies selection of important rows and columns when forming a low rank approximation to the data matrix. Because the selection is performed both on the rows and columns, our SSVD procedure can take into account potential row–column interactions and thus provides a new tool for biclustering. The effectiveness of SSVD has been demonstrated through simulation studies and real data analysis. There are a few potential directions for future research. First, SSVD is developed as an unsupervised learning method. It is interesting to evaluate its usage as a dimension reduction tool prior to use of classification methods. Second, we have focused on using the adaptive lasso penalty. It is worthwhile to consider other sparsity-inducing penalties, such as the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001), the elastic net penalty (Zou and Hastie, 2005), the OSCAR penalty (Bondell and Reich, 2008), and the adaptive grouping penalty (Wang and Zhu, 2008). Finally, while some traditional asymptotic theory for penalized regression has been well developed (Zou, 2006), developing similar results for SSVD is wide open. For example, for biclustering, we no longer have i.i.d. samples, and there are a number of candidates for a reasonable setup for asymptotic analysis.

6. Supplementary Materials

The Web Appendix is available under the Paper Information link at the *Biometrics* website `http://www.biometrics.tibs.org`. The lung cancer data and the SSVD programs (in `R` and `Matlab`) are available from Haipeng Shen's website `http://www.unc.edu/~haipeng`.

Acknowledgements

The authors extend grateful thanks to one coeditor, one associate editor, and three reviewers for their constructive comments. ML, HS, and JSM are partially supported by NSF grant DMS-0606577. JZH is partially supported by NSF grants DMS-0606580, DMS-0907170, NCI grant CA57030, and award KUS-CI-016-04, made by King Abdullah University of Science and Technology (KAUST).

References

Asgarian, N. and Greiner, R. (2008). *Using rank-1 biclusters to classify microarray data*. Technical Report, University of Alberta, Canada.

Bondell, H. and Reich, B. (2008). Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics* **64,** 115–123.

Breiman, L. (1995). Better subset regression using the nonnegative garotte. *Technometrics* **37,** 373–384.

Busygin, S., Prokopyev, O. A., and Pardalos, P. M. (2005). Feature selection for consistent biclustering via fractional 0-1 programming. *Journal of Combinatorial Optimization* **10,** 7–21.

Busygin, S., Prokopyev, O. A., and Pardalos, P. M. (2008). Biclutering in data mining. *Computers and Operations Research* **35,** 2964–2987.

Dhillon, I. S., Mallela, S., and Modha, D. S. (2003). Information-theoretic co-clustering. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 89–98. Heidelberg, Germany: Springer.

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

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96,** 1348–1360.

Jolliffe, I. T. (2002). *Principal Component Analysis*, 2nd edition. New York: Springer-Verlag.

Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. (2003). A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics* **12,** 531–547.

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

Lazzeroni, L. and Owen, A. (2002). Plaid models for gene expression data. *Statistica Sinica* **12,** 61–86.

Leng, C. and Wang, H. (2009). On general adaptive sparse principal component analysis. *Journal of Computational and Graphical Statistics* **18,** 201–215.

Liu, Y., Hayes, D. N., Nobel, A., and Marron, J. S. (2008). Statistical significance of clustering for high dimension low sample size data. *Journal of the American Statistical Association* **103,** 1281–1293.

Madeira, S. and Oliveira, A. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics* **1,** 24–45.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics* **6,** 461–464.

Shabalin, A., Weigman, V., Perou, C., and Nobel, A. (2009). Finding large average submatrices in high dimensional data. *Annals of Applied Statistics* **3,** 985–1012.

Shen, H. and Huang, J. Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis* **99,** 1015–1034.

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

Wang, H., Li, G., and Tsai, C. (2007). Regression coefficient and autoregressive order shrinkage and selection via lasso. *Journal of the Royal Statistical Society, Series B* **69,** 63–78.

Wang, S. and Zhu, J. (2008). Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics* **64,** 440–448.

Yang, W., Dai, D., and Yan, H. (2007). Biclustering of microarray data based on singular value decomposition. *Emerging Technologies in Knowledge Discovery and Data Mining*, 194–205.

Zhang, H. H. and Lu, W. (2007). Adaptive-lasso for Cox's proportional hazard model. *Biometrika* **94,** 691–703.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* **101,** 1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B* **67,** 301–320.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics* **15,** 265–286.

Zou, H., Hastie, T., and Tibshirani, R. (2007). On the "degrees of freedom" of the lasso. *Annals of Statistics* **35,** 2173–2192.

Received December 2008. *Revised December* 2009. *Accepted December* 2009.