A Sparse Singular Value Decomposition Method for High-Dimensional Data

Dan YANG, Zongming MA, and Andreas BUJA

We present a new computational approach to approximating a large, noisy data table by a low-rank matrix with sparse singular vectors. The approximation is obtained from thresholded subspace iterations that produce the singular vectors simultaneously, rather than successively as in competing proposals. We introduce novel ways to estimate thresholding parameters, which obviate the need for computationally expensive cross-validation. We also introduce a way to sparsely initialize the algorithm for computational savings that allow our algorithm to outperform the vanilla singular value decomposition (SVD) on the full data table when the signal is sparse. A comparison with two existing sparse SVD methods suggests that our algorithm is computationally always faster and statistically always at least comparable to the better of the two competing algorithms. Supplementary materials for the article are available in an online appendix. An R package ssd implementing the algorithms introduced in this article is available on CRAN.

**Key Words:** Cross-validation; Denoising; Low-rank matrix approximation; Penalization; Power iterations; Principal component analysis; Thresholding.

1. INTRODUCTION

Interest in multivariate analysis has surged in recent years, partly due to the increasing importance of large data problems in areas such as genomics, imaging, and financial markets, among others. With concern over data size as the main driver of research, traditional multivariate methods have been in need of updating in at least two areas: computational efficiency and statistical performance in high dimensions. This, however, may lead to difficulties in that modifying multivariate methods for statistical performance in high dimensions may make them less computationally efficient because of a need to estimate regularization parameters in a data-driven way. Then again, in this article, we present a multivariate method that has two-fold merit in that it has provable asymptotic optimality properties and at the same time it can be faster than classical methods when the data have certain sparsity properties. The goal of the present article is to deal with computations and methodology, whereas theoretical optimality results are deferred to a companion article.

Dan Yang is Assistant Professor (E-mail: dyang@stat.rutgers.edu), Zongming Ma is Assistant Professor (E-mail: zongming@wharton.upenn.edu), and Andreas Buja is Professor (E-mail: buja.at.wharton@gmail.com), Department of Statistics, University of Pennsylvania, Philadelphia, PA 19104.

© 2014 American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America

*Journal of Computational and Graphical Statistics*, Volume 23, Number 4, Pages 923–942

DOI: 10.1080/10618600.2013.858632

Color versions of one or more of the figures in the article can be found online at [www.tandfonline.com](http://www.tandfonline.com).
In what follows we motivate our model and method in contradistinction to principal component analysis (PCA), to which we now turn. Traditionally, PCA has been understood as eigendecomposition of the covariance matrix of the columns/variables, the rows being interpreted as iid samples from a multivariate distribution. To gain an understanding of PCA in terms of statistical performance, there has been a need to postulate models in which PCA estimates the relevant parameters. Chief among such models is the “spiked covariance” model that assumes the $p \times p$ covariance matrix $\Sigma$ of the variables to have the form

$$\Sigma = V D^2 V^T + \sigma^2 I,$$

where the matrix $V$ is of size $p \times r$ with orthonormal columns and $r \ll p$, thereby representing the idea that few dimensions of variation stand out from a background of noise. This model has been used by Baik and Silverstein (2006), Paul (2007), Paul and Johnstone (2007), Johnstone and Lu (2009), and Ma (2013), among others.

A peculiarity of the spiked covariance model is that it assumes the variables to have the same units, else the single noise variance $\sigma^2$ for all variables would not be meaningful:

$$\text{var}[X_{i,j}] = (VD^2V)^{j,j} + \sigma^2. \tag{A}$$

(A more general model that allows arbitrary units for the variables would have to use a general diagonal matrix $\Lambda = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p)$ to describe the “unique” noise variances: $\Sigma = VD^2V^T + \Lambda^2$.) The assumption of identical measurement scales makes sense if the model is targeted at “functional data analysis” where the variables are evaluations of a stochastic function at $p$ discrete locations of an underlying domain such as space or time or frequency. In functional data analysis, therefore, all variables transform identically under a change of units.

The model considered in this article differs from the spiked covariance model in that it is a mean model rather than a covariance model. We will assume the variables $X_{i,j}$ can be modeled as follows:

$$X = UDV^T + Z, \quad Z_{i,j} \sim \text{iid}, \quad E[Z_{i,j}] = 0, \quad \text{var}[Z_{i,j}] = \sigma^2, \tag{1}$$

where $U$, $V$, $D$, and $\sigma^2$ are fixed but unknown parameters, $U$ is of size $n \times r$, $V$ is of size $p \times r$, both with $r$ orthonormal columns, and $r \ll \min(n, p)$, representing the idea that the mean structure $E[X] = UDV^T$ has few dimensions that rise above iid noise $Z_{i,j}$. We may call this the “spiked mean model.”

Like the spiked covariance model, the spiked mean model is also naturally directed at functional data analysis, but in a two-way sense: both, the columns and the rows, are interpreted functionally as discrete locations in separate domains. For example, in spatio-temporal data the rows would represent locations $s_i$ in space and the columns points $t_j$ in time. The data matrix arises therefore as a set of evaluations of a bivariate random function $X(s, t)$ on a grid $(s_i, t_j)$: $X_{i,j} = X(s_i, t_j)$. Similarly, the columns $U_{(i,k)}$ and $V_{(j,k)}$ of the singular matrices $U$ and $V$ are evaluations of singular functions: $U_{i,k} = U_{(k)(s_i)}$ and $V_{j,k} = V_{(k)(t_j)}$. The spiked mean model should be thought of as arising from a low-rank structure for the mean function $\mu(s, t) = E[X(s, t)]: \mu(s, t) = \sum_{k=1}^{\infty} D_k U_{(k)(s)} V_{(k)(t)}.$

While the “obvious” method for estimating the parameters of the spiked covariance model is an eigendecomposition of the empirical covariance method, the obvious analog for the parameters of the spiked mean model is a singular value decomposition (SVD) of the full data matrix. In practice, the true rank $r$ is not known in either case, hence both procedures need to be augmented with an estimation method for the rank $r$ of the
true denoised covariance matrix $VD^2V'$ and for the true denoised mean matrix $UDV'$, respectively.

A critical statistical issue that has arisen in large datasets is that in very high dimensional settings, that is, for very large $n$ and $p$, the eigendecomposition of the empirical covariance matrix and the SVD of the data matrix behave poorly as estimators in their respective spiked models. This has been shown for the eigendecomposition in the spiked covariance model by Nadler (2009), Paul (2007), and Johnstone and Lu (2009), and for the SVD in the spiked mean model by Shabalin and Nobel (2010). The reason is that in high dimensions the noise can overwhelm the signal to such an extent that the traditional empirical estimates of eigenvectors and singular vectors are not even near the ballpark of the underlying truth and can therefore be entirely misleading. Compounding the problems in large datasets are the difficulties of computing accurate eigendecompositions and SVDs at affordable cost. Obtaining statistically viable estimates of eigenvectors and eigenspaces for the spiked covariance model in high-dimensional data has been the focus of a considerable literature; a representative but incomplete list of references is Lu (2002), Zou, Hastie, and Tibshirani (2006), Paul (2007), Paul and Johnstone (2007), Shen and Huang (2008), Johnstone and Lu (2009), Shen, Shen, and Marron (2013), and Ma (2013). On the other hand, overcoming similar problems for the spiked mean model has been the subject of far less work, pertinent articles being Witten, Tibshirani, and Hastie (2009), Lee et al. (2010), Huang, Shen, and Buja (2009), and Allen, Grosenick, and Taylor (2011).

In high dimensions, statistical estimation from matrix data is not possible without the assumption of strong structure in the data. This is so in the simpler case of vector data from Gaussian sequence models (Johnstone 2011), but even more so for matrix data from either spiked covariance or spiked mean models. Both require assumptions of low rank $r$ as well as assumptions of sparsity or smoothness. Of the latter two, sparsity has slightly greater generality because certain types of smoothness can be reduced to sparsity through suitable basis changes (Johnstone 2011). In the spiked mean model, by imposing sparseness on singular vectors, one may be able to “sharpen” the structure in data and thereby expose “checkerboard” patterns that convey biclustering structure, that is, joint clustering in the row and column domains of the data (Lee et al. 2010; Sill et al. 2011). Going one step further, Witten and Tibshirani (2010) used sparsity to develop a novel form of hierarchical clustering.

A few recent articles propose sparsity approaches to the high-dimensional SVD problem: Witten, Tibshirani, and Hastie (2009) introduced a matrix decomposition that constrains the $l_1$ norm of the singular vectors to impose sparsity on the solutions. Lee et al. (2010) used penalized LS for rank-one matrix approximations with $l_1$ norms of the singular vectors as additive penalties. Both methods use iterative procedures to solve different optimization problems. (We will give more details about these two methods in Section C of the Appendix.) Allen, Grosenick, and Taylor (2011) is a Lagrangian version of Witten, Tibshirani, and Hastie (2009) where the errors are permitted to have a known type of dependence and/or heteroscedasticity. These articles focus on estimating the first rank-one term given by $d_1, \hat{u}_1, \hat{v}_1$ by either constraining the $l_1$ norm of $\hat{u}_1$ and $\hat{v}_1$ or adding it as a penalty. To estimate $d_2, \hat{u}_2, \hat{v}_2$ for a second rank-one term, they subtract the first term $d_1 \hat{u}_1 \hat{v}_1' \hat{v}_1$ from the data matrix $X$ and repeat the procedure on the residual matrix. There exists further related work on sparse matrix factorization, for example, by Zheng et al. (2007),
Mairal et al. (2010), and Bach, Mairal, and Ponce (2008), but these do not have the form of an SVD. In our simulations and data examples, we use the proposals by Witten, Tibshirani, and Hastie (2009) and Lee et al. (2010) for comparison.

Our approach is to estimate the subspaces spanned by the $r$ leading singular vectors simultaneously. As a result, our method yields sparse singular vectors that are orthogonal, unlike the proposals by Witten, Tibshirani, and Hastie (2009) and Lee et al. (2010). In terms of statistical performance, simulations show that our method is competitive with the better performing of the two proposals, which is generally Lee et al. (2010). In terms of computational speed, our method is faster by at least a factor of two compared with the more efficient of the two proposals, which is generally Witten, Tibshirani, and Hastie (2009). Thus, we show that the current state of the art in sparse SVDs is “inadmissible” if measured by the two metrics “statistical performance” and “computational speed”: our method closely matches the better statistical performance and provides it at a fraction of the better computational performance. In fact, by making use of sparsity at the initialization stage, our method can also beat the conventional SVD in terms of speed if there is indeed sparsity in the data.

Finally, our method is grounded in asymptotic theory that comprises minimax results that we describe in a companion article (Yang, Ma, and Buja 2011); a summary is available in an online appendix. A signature of the theory is that it is not concerned with optimization problems but with a class of iterative algorithms that form the basis of the methodology proposed here. The algorithms are “sparcifications” of the well-known SVD iterations that update left singular vectors/subspaces from right singular, and vice versa. The theory derives asymptotic minimax results directly for the outputs from these iterations without the need to interpret them as optimization. As do most asymptotic theories in this area, ours rely on Gaussianity of noise, which is the major aspect that needs robustification when turning theory into methodology with a claim to practical applicability. Essential aspects of our proposal therefore relate to lesser reliance on the Gaussian assumption.

The rest of the article is organized as follows. Section 2 describes our method for computing sparse SVDs. Section 3 applies our and the competing methods to real data examples. Finally, Section 4 discusses the results and open problems. In the Appendix, Section B states the minimax result under normal assumptions and Section C shows simulation results to compare the performance of our method with that of Witten, Tibshirani, and Hastie (2009) and Lee et al. (2010).

2. METHODOLOGY

In this section, we give a detailed description of the proposed sparse SVD method.

To start, consider the noiseless case. Our sparse SVD procedure is motivated by the simultaneous orthogonal iteration algorithm (Golub and Van Loan 1996, chap. 8), which is a straightforward generalization of the power method for computing higher-dimensional invariant subspaces of symmetric matrices. For an arbitrary rectangular matrix $\Xi$ of size $n \times p$ with SVD $\Xi = UDV'$, one can find the subspaces spanned by the first $r$ ($1 \leq r \leq \min(n, p)$) left and right singular vectors by iterating the pair of mappings $V \mapsto U$ and $U \mapsto V$ with $\Xi$ and $\Xi'$ (its transpose), respectively, each followed by orthonormalization,
until convergence. More precisely, given a right starting frame $V^{(0)}$, that is, a $p \times r$ matrix with $r$ orthonormal columns, the SVD subspace iterations repeat the following four steps until convergence:

$$
\begin{align*}
(1) \text{Right-to-Left Multiplication:} & \quad U^{(k),\text{mul}} = \Xi V^{(k-1)} \\
(2) \text{Left Orthonormalization with QR Decomposition:} & \quad U^{(k)} R^{(k)} = U^{(k),\text{mul}} \\
(3) \text{Left-to-Right Multiplication:} & \quad V^{(k),\text{mul}} = \Xi' U^{(k)} \\
(4) \text{Right Orthonormalization with QR Decomposition:} & \quad V^{(k)} R^{(k)} = V^{(k),\text{mul}}
\end{align*}
$$

The superscript $(k)$ indicates the $k$th iteration, and $\text{mul}$ the generally nonorthonormal intermediate result of multiplication. For $r = 1$, the QR decomposition step reduces to normalization. If $\Xi$ is symmetric, the second pair of steps is the same as the first pair, hence the original orthogonal iteration algorithm for symmetric matrices is a special case of the above algorithm.

The problems our approach addresses are the following: for large noisy matrices in which the significant structure is concentrated in a small subset of the matrix $X$, the classical algorithm outlined above produces estimates with large variance due to the accumulation of noise from the majority of structureless cells (Shabalin and Nobel 2010). In addition to the detriment for statistical estimation, involving large numbers of structureless cells in the calculations adds unnecessary computational cost to the algorithm. Thus, shaving off cells with little apparent structure has the promise of both statistical and computational benefits. This is indeed borne out in the following proposal for a sparse SVD algorithm.

### 2.1 THE FIT-SSVD ALGORITHM: “FAST ITERATIVE THRESHOLDING FOR SPARSE SVDS”

Unsurprisingly, the algorithm proposed here involves some form of thresholding, be it soft or hard or something in between. All thresholding schemes reduce small coordinates in the singular vectors to zero, and additionally such schemes may or may not shrink large coordinates as well. Any thresholding reduces variance at the cost of some bias, but if the sparsity assumption is not too unrealistic, the variance reduction will vastly outweigh the bias inflation. The obvious places for inserting thresholding steps are right after the multiplication steps. If thresholding reduces a majority of entries to zero, the computational cost for the subsequent multiplication and QR decomposition steps is much reduced as well. The iterative procedure we propose is schematically laid out in Algorithm 1.

In what follows we discuss the thresholding function and convergence criterion of Algorithm 1. Subsequently, in Sections 2.2–2.4, we describe other important aspects of the algorithm: the initialization of the orthonormal matrix, the target rank, and the adaptive choice of threshold levels.

**Thresholding function.** At each thresholding step, we perform entry-wise thresholding. In our modification of the subspace iterations (2), we allow any thresholding function $\eta(x, \gamma)$ that satisfies $|\eta(x, \gamma) - x| \leq \gamma$ and $\eta(x, \gamma)1_{|x| \leq \gamma} = 0$, which includes soft-thresholding with $\eta_{\text{soft}}(x, \gamma) = \text{sign}(x)(|x| - \gamma)_+$, hard-thresholding with $\eta_{\text{hard}}(x, \gamma) =$
Input:
1. Observed data matrix $X$.
2. Target rank $r$.
3. Thresholding function $\eta$.
4. Initial orthonormal matrix $V^{(0)} \in \mathbb{R}^{p \times r}$.
5. Algorithm $f$ to calculate the threshold level $\gamma = f(X, U, V, \hat{\sigma})$ given (a) the data matrix $X$, (b) current estimates of left and right singular vectors $U$, $V$, and (c) an estimate of the standard deviation of noise $\hat{\sigma}$.

(Algorithm 3 is one choice.)

Output: Estimators $\hat{U} = U^{(\infty)}$ and $\hat{V} = V^{(\infty)}$.

1. Set $\hat{\sigma} = 1.4826 \text{MAD}(\text{as.vector}(X))$.

repeat
2. Right-to-Left Multiplication: $U^{(k), \text{mul}} = XV^{(k-1)}$.
3. Left Thresholding: $U^{(k), \text{thr}} = (u_{il}^{(k), \text{thr}})$, with $u_{il}^{(k), \text{thr}} = \eta(u_{il}^{(k), \text{mul}}, \gamma_{ul})$, where $\gamma_{ul} = f(X, U^{(k-1)}, V^{(k-1)}, \hat{\sigma})$.
4. Left Orthonormalization with QR Decomposition: $U^{(k), \text{mul}} = U^{(k), \text{thr}}$.
5. Left-to-Right Multiplication: $V^{(k), \text{mul}} = X'U^{(k)}$.
6. Right Thresholding: $V^{(k), \text{thr}} = (v_{jl}^{(k), \text{thr}})$, with $v_{jl}^{(k), \text{thr}} = \eta(v_{jl}^{(k), \text{mul}}, \gamma_{vl})$, where $\gamma_{vl} = f(X', V^{(k-1)}, U^{(k)}, \hat{\sigma})$.
7. Right Orthonormalization with QR Decomposition: $V^{(k), \text{mul}} = V^{(k), \text{thr}}$.
until Convergence;

Algorithm 1: FIT-SSVD.

$x_1|x| > \gamma$, as well as the thresholding function used in SCAD (Smoothly Clipped Absolute Deviation) (Fan and Li 2001). The parameter $\gamma$ is called the threshold level. In Algorithm 1, we apply the same threshold level $\gamma_{ul}$ (or $\gamma_{vl}$) to all the elements in the $l$th column of $U^{(k), \text{mul}}$ (or $V^{(k), \text{mul}}$, resp.). For more details on threshold levels, see Section 2.4.

Convergence criterion. We stop the iterations once subsequent updates of the orthonormal matrices are very close to each other. In particular, for any matrix $H$ with orthonormal columns (i.e., $H' H = I$), let $P_H = H H'$ be the associated projection matrix. We stop after the $k$th iteration if $\max\{\|P_{U^{(k)}} - P_{U^{(k-1)}}\|_2^2, \|P_{V^{(k)}} - P_{V^{(k-1)}}\|_2^2\} \leq \epsilon$, where $\epsilon$ is a prespecified tolerance level, chosen to be $\epsilon = 10^{-8}$ for the rest of this article. ($\|A\|_2$ denotes the spectral norm of $A$.)

2.2 Initialization Algorithm for FIT-SSVD

In Algorithm 1, we need a starting frame $V^{(0)}$ such that the subspace it spans has no dimension that is orthogonal to the subspace spanned by the true $V$. Most often used is the $V$ frame provided by the ordinary SVD. However, due to its denseness, computational cost, and inconsistency (Shabalin and Nobel 2010), it makes an inferior starting frame. Another popular choice is initialization with a random frame, which, however, is often
**Input:**
1. Observed data matrix \( X \).
2. Target rank \( r \).
3. Degree of “Huberization” \( \beta \) (typically 0.95 or 0.99), that defines a quantile of the absolute values of entries in \( X \).
4. Significance level of a selection test \( \alpha \).

**Output:** Orthogonal matrices \( \hat{U} = U^{(0)} \) and \( \hat{V} = V^{(0)} \).

1. **Subset selection:**
   - Let \( \delta \) be the \( \beta \)-quantile of the absolute values of all the entries in \( X \).
   - Define \( Y = (y_{ij}) \) by \( y_{ij} = \rho(x_{ij}, \delta) \), where \( \rho(x, \delta) \) is the Huber \( \rho \) function:
     \[
     \rho(x, \delta) = x^2 \text{ if } |x| \leq \delta \text{ and } 2\delta|x| - \delta^2 \text{ otherwise.}
     \]
   - Calculate \( p \)-values:
     \[ p_i = 1 - \Phi \left( \frac{t_i - \hat{\mu}}{\hat{s}} \right), \]
     where \( \Phi \) is the CDF of \( N(0, 1) \).
   - Perform the Holm method on the \( p \)-values at family-wise error rate \( \alpha \), and let \( I \) be the indices of the \( p \)-values that result in rejection.
   - Select a subset of columns \( J \) similarly.
   - Form the submatrix \( X_{IJ} \) of size \(|I| \times |J| \).

2. **Reduced SVD:** Compute \( r \) leading pairs of singular vectors of the submatrix \( X_{IJ} \). Denote them by \( u_{i1}^{(0)}, \ldots, u_{ir}^{(0)} \) (\( |I| \times 1 \) each) and \( v_{j1}^{(0)}, \ldots, v_{jr}^{(0)} \) (\( |J| \times 1 \) each).

3. **Zero-padding:** Create \( U^{(0)} = [u_{i1}^{(0)}, \ldots, u_{ir}^{(0)}] \) \((n \times r)\) and \( V^{(0)} = [v_{j1}^{(0)}, \ldots, v_{jr}^{(0)}] \) \((p \times r)\), such that \( u_{i1}^{(0)} = u_{i1}, u_{ir}^{(0)} = 0, v_{j1}^{(0)} = v_{j1}, v_{jr}^{(0)} = 0 \).

**Algorithm 2:** Initialization algorithm for FIT-SSVD.

nearly orthogonal to the true \( V \) and thus requires many iterations to accumulate sufficient power to converge. We propose therefore Algorithm 2 that overcomes these difficulties.

The algorithm is motivated by Johnstone and Lu (2009) who obtained a consistent estimate for principal components under a sparsity assumption by initially reducing the dimensionality. We adapt their scheme to the two-way case, and we weaken its reliance on the assumption of normal noise, which in real data would result in too great a sensitivity to even slightly heavier tails than normal. To this end, we make use of some devices from robust estimation. The intent is to perform a row- and column-preselection (Step 1) before applying a classical SVD (Step 2) so as to concentrate on a much smaller submatrix that contains much of the signal. We discuss the row selection process, column selection being analogous.

Signal strength in rows would conventionally be measured under Gaussian assumptions with row sums of squares and tested with \( \chi^2 \) tests with \( p \) degrees of freedom. As mentioned this approach turns out to be much too sensitive when applied to real data matrices due to isolated large cells that may stem from heavier than normal tails. We therefore mute the influence of isolated large cells by Huberizing the squares before forming row sums. We then form approximate \( z \)-score test statistics, one per row, drawing on the CLT (central limit theorem) since we assume \( p \) (the number of entries in each row) to be large. Location
and scale for the z-scores are estimated with the median and MAD ("median absolute
deviation," instead of mean and standard deviation) of the row sums, the assumption being
that over half of the rows are approximate “null rows” with little or no signal. If the signal
is not sparse in terms of rows, this procedure will have low power, which is desirable
because it biases the initialization of the iterative Algorithm 1 toward sparsity. Using robust
z-score tests has two benefits over $\chi^2$ tests: they are robust to isolated large values, and they
avoid the sensitivity of $\chi^2$ tests caused by their rigid coupling of expectation and variance.
Finally, since $n$ tests are being performed, we protect against over-detection due to multiple
testing by applying Holm’s (1979) stepwise testing procedure at a specified family-wise
significance level $\alpha$ (default: 5%). The end result is a set of indices $I$ of “significant rows”—
the same procedure is then applied to the columns, resulting in an index set $J$ of “significant
columns.”

The submatrix $X_{IJ}$ is then submitted to an initial reduced SVD. It is this initial reduction
that allows the present algorithm to be faster than a conventional SVD of the full matrix
$X$ when the signal is sparse. The left and right singular vectors are of size $|I|$ and $|J|$, respectively. To serve as initializations for the iterative Algorithm 1, they are expanded
and zero-padded to length $n$ and $p$, respectively (Step 3)—this concludes the initialization
Algorithm 2.

2.3 Rank Estimation

In Algorithm 1, a required input is the presumed rank of the signal underlying $X$. In
practice, we need to determine the rank based on the data. Proposals for rank estimation
are the subject of a literature with a long history, of which we only cite Wold (1978), Gabriel
(2002), and Hoff (2007). The proposal we chose is the bi-cross-validation (BCV) method
by Owen and Perry (2009), but with a necessary twist.

The original BCV method was proposed for low-rank matrices with dense singular
vectors. Thus, we apply it to the submatrix $X_{IJ}$ obtained from the initialization Algorithm
2, instead of $X$ itself. The submatrix should have much denser singular vectors and, even
more importantly, much higher signal-to-noise ratio compared with the full matrix. In
simulations not reported here but similar to those of Section C of the Appendix, BCV on
$X_{IJ}$ yielded consistent rank estimation when the signal was sufficiently strong for detection
in relation to sparsity and signal-to-noise ratio.

2.4 Threshold Levels

The tuning parameters $\gamma$ in the thresholding function $\eta(x, \gamma)$ are called “threshold
levels,” they play a key role in the procedure. At each thresholding step in Algorithm 1, a
(potentially different) threshold level needs to be chosen for each column $l = 1, \ldots, r$ of
$U^{(k)}$ and $V^{(k)}$ to strike an acceptable bias-variance trade-off. In what follows, we focus on
$U^{(k)}$, while the case of $V^{(k)}$ can be obtained by symmetry.

The goal is to process the iterating left and right frames in such a way as to retain the
coordinates with high signal and eliminate those with low signal. To be more specific,
we focus on one column $u^{(k), \text{mul}}_l = X^{(k-1)}_l$. Recall that $X$ is assumed to admit an additive
decomposition into a low-rank signal plus noise according to model (1). Then a theoretically
sensible (though not actionable) threshold level for \( u^{(k),\text{mul}} \) would be \( \gamma_{ul} = E[\| Zv^{(k-1)} \|_\infty] \), where \( Z \) is the additive noise matrix, and \( \| Zv^{(k-1)} \|_\infty \) is the maximum absolute value of the \( n \) entries in the vector \( Zv^{(k-1)} \). The signal of any coordinate in \( u^{(k),\text{mul}} \) with value less than \( \gamma_{ul} \) could be regarded low since it is weaker than the expected maximum noise level in the \( l \)th rank given that there are \( n \) rows.

The threshold \( \gamma_{ul} \) as written above is of course not actionable because it involves knowledge of \( Z \), but we can obtain information by leveraging the (presumably large) part of \( X \) that is estimated to have no or little signal. This can be done as follows: Let \( L_u \) be the index set of rows that have all zero entries in \( U^{(k-1)} \), and let \( H_u \) be its complement; define \( L_v \) and \( H_v \) analogously. We may think of \( L_u \) and \( L_v \) as the current estimates of low-signal rows and columns. Consider next a reordering and partitioning of the rows and columns of \( X \) according to these index sets:

\[
X = \begin{pmatrix} X_{H_u,H_v} & X_{H_u,L_v} \\ X_{L_u,H_v} & X_{L_u,L_v} \end{pmatrix}. \tag{3}
\]

Since the entries in \( v^{(k-1)} \) corresponding to \( L_v \) are zero, only \( X_{H_v} \) (of size \( n \times |H_v| \), containing the two left blocks in (3)) is effectively used in the right-to-left multiplication of the iterative Algorithm 1. We can therefore simulate a “near-null” situation in this block by filling it with random samples from the bottom right block, which we may assume to have no or only low signal: \( X_{L_u,L_v} \approx Z_{L_u,L_v}. \) Denote the result of such “bootstrap transfer” from \( X_{L_u,L_v} \) to \( X_{H_v} \) by \( \tilde{Z}^* (n \times |H_v|) \). Passing \( \tilde{Z}^* \) through the right-to-left multiplication with \( v^{(k-1)} \) we form \( \tilde{Z}^* v^{(k-1)} \), which we interpret as an approximate draw from \( Zv^{(k-1)} \).

We thus estimate \( \| Zv^{(k-1)} \|_\infty \) with \( \| \tilde{Z}^* v^{(k-1)} \|_\infty \), and \( E[\| Zv^{(k-1)} \|_\infty] \) with a median of \( \| \tilde{Z}^* v^{(k-1)} \|_\infty \) over multiple bootstraps of \( Z^* \).

In order for this to be valid, the block \( X_{L_u,L_v} \) needs to be sufficiently large in relation to \( X_{H_v} \). This is the general problem of the “\( m \) out of \( n \)” bootstrap, which was examined by Bickel, Gotze, and Zwet (1997). According to their results, this bootstrap is generally consistent as long as \( m = o(n) \). Hence, when the size \( |L_u||L_v| \) of the matrix \( X_{L_u,L_v} \) is large, say, larger than \( n|H_v| \log(n|H_v|) \), we estimate \( E[\| Zv^{(k-1)} \|_\infty] \) by the median of \( M \) bootstrap replications for sufficiently large \( M \). When the condition is violated, \( |H_v| \) tends to be large, the central limit theorem takes effect, and each element of \( Zv^{(k-1)} \) would be close to a normal random variable. Thus, the expected value of the maximum is near the asymptotic value \( \sqrt{2 \log n} \) times the standard deviation—we have now fully defined the threshold \( \gamma_{ul} \) to be used on \( u^{(k),\text{mul}} \). The thresholds for \( l = 1, \ldots, r \) are then collected in the threshold vector \( \gamma_u = (\gamma_{u1}, \ldots, \gamma_{ur}) \).

A complete description of the scheme is given in Algorithm 3. Based on an extensive simulation study, setting the number of bootstrap replications to \( M = 100 \) yields a good balance between the accuracy of the threshold level estimates and computational cost.

### 2.5 Alternative Methods for Selecting Threshold Levels

In methods for sparse data, one of the most critical issues is selecting threshold levels wisely. Choosing thresholds too small kills off too few entries and retains too much variance, whereas choosing them too large kills off too many entries and introduces too much bias.
Input:
1. Observed data matrix \( X \in \mathbb{R}^{n \times p} \);
2. Previous estimators of singular vectors \( U^{(k)} \in \mathbb{R}^{n \times r} \), \( V^{(k)} \in \mathbb{R}^{p \times r} \);
3. Prespecified number \( M \) of bootstraps;
4. Estimate of the standard deviation of noise \( \hat{\sigma} \).

Output: Threshold level \( \gamma \in \mathbb{R}^r \).

1. Subset selection: \( L_u = \{ i : u_{i1}^{(k)} = \cdots = u_{ir}^{(k)} = 0 \} \), \( L_v = \{ j : v_{j1}^{(k)} = \cdots = v_{jr}^{(k)} = 0 \} \), \( H_u = L_u^c \), \( H_v = L_v^c \);
2. if \( |L_u| \|L_u\| < n |H_v| \log(n|H_v|) \) then
   return \( \gamma = \hat{\sigma} \sqrt{2 \log(n)} 1 \in \mathbb{R}^r \);
else
   for \( i \leftarrow 1 \) to \( M \) do
   Sample \( n |H_u| \) entries from \( X_{L_u,L_u} \) and reshape them into a matrix \( \tilde{Z} \in \mathbb{R}^{n \times |H_u|} \);
   \( B = \tilde{Z} V_{H_v}^{(k)} \in \mathbb{R}^{n \times r} \);
   \( C_i = (\|B_1\|_\infty, \|B_2\|_\infty, \ldots, \|B_r\|_\infty) \);
   \( \gamma_i = \text{median}(C_i) \);
   return \( \gamma = (\gamma_1, \ldots, \gamma_r) \).

Algorithm 3: The threshold level function \( f(X, U, V, \hat{\sigma}) \) for Algorithm 1. As shown, the code produces thresholds for \( U \). A call to \( f(X', V, U, \hat{\sigma}) \) produces thresholds for \( V \).

To navigate this bias-variance trade-off, we adopted in Section 2.4 an approach that can be described as a form of testing: we established max-thresholds that are unlikely to be exceeded by any \( U \)- or \( V \)-coordinates under the null assumption of absent signal in the corresponding row or column of the data matrix.

To navigate bias-variance trade-offs, other commonly used approaches include various forms of cross-validation, a version of which we adopted for the different problem of rank selection in Section 2.3 (bi-cross-validation or BCV according to Owen and Perry 2009). Indeed, a version of cross-validation for threshold selection is used by one of the two competing proposals with which we compare ours: Witten, Tibshirani, and Hastie (2009) left out random subsets of the entries in the data matrix, measured the differences between the fitted values and the original values for those entries, and chose the threshold levels that minimize the differences. Alternatively one could use BCV for this purpose as well, by leaving out sets of rows and columns and choosing the thresholds that minimize the discrepancy between the hold-out and the predicted values. However, this would be computationally slow for simultaneous minimization of two threshold parameters. Moreover, the possible values of the thresholds vary from zero to infinity, which makes it difficult to choose grid points for the parameters. To avoid such issues, Lee et al. (2010) implemented their algorithm by embedding the optimization of the choice of the threshold level inside the iterations that calculate \( u_l \) for fixed \( v_l \) and \( v_l \) for fixed \( u_l \) (unlike our methods, theirs fits one rank at a time). They minimized a Bayesian information criterion (BIC) over a grid of order statistics of current estimates. This idea could be applied to our simultaneous
space-fitting approach, but the simulation results in Section C of the Appendix below show that the method of Lee et al. (2010) is computationally very slow.

3. REAL DATA EXAMPLES

All the methods mentioned above require sparse singular vectors (with most entries close to zero). One source of such data is two-way functional data whose row and column domains are both structured, for example, temporally or spatially, as when the data are time series collected at different locations in space. Two-way functional data are usually smooth as functions of the row and column domains. Thus, if we expand them in suitable basis functions, such as an orthonormal trigonometric basis, the coefficients should be sparse (Johnstone 2011).

3.1 MORTALITY RATE DATA

As our first example we use the US mortality rate data from the Berkeley Human Mortality Database (http://www.mortality.org/). They contain mortality rates in the United States for ages 0–110 from 1933 to 2007. The data for people older than 95 were discarded because of their noisy nature. The matrix $X$ is of size $96 \times 75$, each row corresponding to an age group and each column to a 1 year period. We first pre- and postmultiply the data matrix with orthogonal matrices whose columns are the eigenvectors of second-order difference matrices of proper sizes; the result is a matrix of coefficients of the same size as $X$. The rank of the signal is estimated to be 2 using BCV (Section 2.3). We then applied FIT-SSVD, LSHM, PMD-SVD (penalized matrix decomposition), and ordinary SVD to get the first two pairs of singular vectors. Finally, we transformed the sparse estimators of the singular vectors back to the original basis to get smooth singular vectors.

The estimated number of nonzero elements in each singular vector (before the back transformation) is summarized in Table 1: none gives very sparse solutions. This is reasonable, because the mortality rate data are of low noise and for data with no noise we should just use the ordinary SVD. Because these data are of small size, they only take a few seconds for all the algorithms. The plots of singular vectors for all the methods are shown in Figures 1–5. The red dashed line in the left plot is for FIT-SSVD, in the middle for LSHM, and on the right for PMD-SVD. We use the wider gray curve for the ordinary SVD as a reference.

Figure 1 shows the first left singular vector plotted against age. The curve $\hat{u}_1$ shows a pattern for mortality as a function of age: a sharp drop between age 0 and 2, then a gradual decrease till the teen years, flat till the 30s, after which begins an exponential increase.

| Table 1. Mortality data: number of nonzero coordinates in the transformed domain for four methods |
|-----------------|-----------------|-----------------|-----------------|
|                 | FIT-SSVD | LSHM | PMD-SVD | SVD |
| $\|\hat{u}_1\|_0$ | 82       | 48   | 96      | 96  |
| $\|\hat{u}_2\|_0$ | 86       | 56   | 7       | 96  |
| $\|\hat{v}_1\|_0$ | 66       | 45   | 75      | 75  |
| $\|\hat{v}_2\|_0$ | 70       | 45   | 43      | 75  |
Figure 1. Mortality data: plot of $\hat{u}_1$. Panel (a) FIT-SSVD versus SVD. Panel (b) LSHM versus SVD. Panel (c) PMD-SVD versus SVD.

Figure 2. Mortality data: Plot of $\hat{u}_1$. Zoom of the lower left corner of Figure 1. Everything else is the same as in Figure 1.

Figure 3. Mortality data: Plot of $\hat{v}_1$. Everything else is the same as in Figure 1.

Figure 4. Mortality data: plot of $\hat{u}_2$. Everything else is the same as Figure 1.
Figure 2 zooms into the lower left corner of Figure 1 to show the details between age 0 and 10. LSHM, as always turns out to be the sparsest (or smoothest) among the three iterative procedures in the transformed (or original) domain. We believe that FIT-SSVD and PMD-SVD make more sense based on a parallel coordinates plot of the raw data (not shown here), in which the drop in the early age appears to be sharp and therefore should not be smoothed out. Figure 3 shows the first right singular vectors plotted against year. It implies that mortality decreases with time. All of the panels show a wiggly structure, with LSHM again being the smoothest. Here, too, we believe that the zigzag structure is real and not due to noise in the raw data, based again on a parallel coordinate plot of the raw data. The zigzags may well be systematic artifacts, but they are unlikely to be noise.

The second pair of singular vectors is shown in Figures 4 and 5: they correct the pattern that the first pair of singular vectors does not capture. The contrast mainly focuses on people younger than 2 or between 60 and 90 where \( \hat{u}_2 \) is positive. Also, \( \hat{v}_2 \) has extreme negative or positive values toward the both ends, 1940s and 2000s. Together, they suggest that babies and older people had lower mortality rates in the 1940s and higher mortality rates in the 2000s than what the first component expresses. One final aspect to note is the strange behavior of \( \hat{u}_{2,\text{PMD-SVD}} \), recalling that \( \hat{u}_{1,\text{PMD-SVD}}, \hat{v}_{1,\text{PMD-SVD}}, \hat{v}_{2,\text{PMD-SVD}} \) all follow the ordinary SVD very closely. We think this is again due to the cross-validation technique they use.

Huang, Shen, and Buja (2009) also used the mortality data from 1959 to 1999 to illustrate their version of regularized SVDs to get smooth singular vectors by adding second-order difference penalties. If we compare the results shown in this section with theirs, our solutions lack the smoothness of their solutions, but we think we recover more information from the data by capturing not only the general trend but also local details such as year-to-year fluctuations.

3.2 CANCER DATA

We consider next another data example where some sparse structure may be expected to exist naturally. The cancer data used by Lee et al. (2010) (who in turn have them from Liu et al. 2008) consist of the gene expression levels of 12,625 genes for 56 cases of four different types of cancer. It is believed that only a part of the genes regulate the types and
Table 2. Cancer data: summary of cardinality of joint support of three singular vectors for four methods

|                | FIT-SSVD | LSHM  | PMD-SVD | SVD  |
|----------------|----------|-------|---------|------|
| $| \bigcup_{l=1}^{3} \text{supp}(\hat{u}_l)|$ | 4688    | 4545  | 12,625 | 12,625 |
| $| \bigcup_{l=1}^{3} \text{supp}(\hat{v}_l)|$ | 56      | 56    | 54     | 56   |

hence the singular vectors corresponding to the genes should ideally be sparse. We apply the four SVD methods directly to the raw data without change of basis.

Before we proceed it may be proper to discuss briefly some modeling ambiguities posed by this dataset as it is not a priori clear whether a spiked covariance or spiked mean model is more appropriate. It might be argued that the cases really should be considered as being sampled from a population, hence PCA in a spiked covariance model would be the proper analysis, with the genes representing the variables. The counter argument is, however, that the cases are stratified, and the strata are pure convenience samples of sizes that bear no relation to the sizes of naturally occurring cancer populations. A dual interpretation with genes as samples and cases as variables would be conceivable also, but it seems even more far fetched in the absence of any sampling aspect with regard to genes. In light of the problems raised by any sampling assumption, it would seem more appropriate to condition on the cases and the genes and adopt a fixed effects view of the data. As a result the spiked mean model seems less problematic than either of the dual spiked covariance models.

We first attempted to estimate the rank of the signal using BCV (Section 2.3), but it turns out that the rank is sensitive to the choice of $\alpha$ (Holm family-wise error) and $\beta$ (Huberization quantile) in Algorithm 2, ranging from $r = 3$ to $r = 5$. We decided to use $r = 3$ because this is the number of contrasts required to cluster the cases into four groups. Also, this is the rank used by Lee et al. (2010), which grants comparison of their and our results.

On a different note, running LSHM on these data with rank three took a couple of hours, which may be a disincentive for users to seek even higher ranks. The hours of run time of LSHM compares with a few minutes for PMD-SVD and merely a few seconds for FIT-SSVD. (In addition, LSHM’s third singular vectors do not seem to converge within 300 iterations.)

Table 2 summarizes the cardinalities of the union of supports of three singular vectors for each method. For the estimation of left singular vectors corresponding to different genes, the PMD-SVD solution is undesirably dense, while FIT-SSVD and LSHM give similar levels of sparsity. For the estimation of right singular vectors corresponding to the cases, we would expect that all cases have their own effects rather than zero, so it is not surprising that the estimated singular vectors are dense.

Figure 6 shows the scatterplots of the entries of the first three right singular vectors for the four methods. Points represent patients, each row represents one method, and each column corresponds to two of the three singular vectors. The four known groups of patients are easily discerned in the plots. A curiosity is the cross-wise structure produced by PMD-SVD, where the singular vectors are nearly mutually exclusive: if one coordinate in a singular vector is nonzero, most corresponding coordinates in the other singular vectors are zero. The other three methods, including the ordinary SVD, agree strongly among each other in
the placement of the patients. The agreement with the ordinary SVD is not a surprise as $p = 56$ is a relatively small column dimension on which sparsity may play a less critical role compared to the row dimension with $n = 12,625$. Yet, the three sparse methods give clearer evidence that the carcinoid group (black circles) falls into two subgroups than the
ordinary SVD. According to FIT-SSVD and LSHM, the separation is along \( \hat{v}_3 \) (center and right-hand plots), whereas according to PMD-SVD it is by lineup with \( \hat{v}_1 \) and \( \hat{v}_2 \), respectively (left-hand plot).

Figure 7 shows checkerboard plots of the reconstructed rank-three approximations, laid out with patients on the vertical axis and genes on the horizontal axis. Each row of plots represents one method, and the plots in a given row show the same reconstructed matrix but successively ordered according to the coordinates of the estimated left singular vectors \( \hat{u}_1 \), \( \hat{u}_2 \), and \( \hat{u}_3 \). There are fewer than 5000 genes shown for FIT-SSVD and LSHM, because the rest are estimated to be zero, whereas all 12,625 genes are shown for PMD-SVD and SVD (Table 2). We can see clear checkerboard structure in some of the plots, indicating biclustering. In spite of the strong similarity between the patient projections for FIT-SSVD and LSHM in Figure 6, there is a clear difference between these methods in the reconstructions in Figure 7: the FIT-SSVD solution exhibits the strongest block structure in its \( \hat{u}_2 \)-based sort (center plot, top row), implying the strongest evidence of clustering among its nonthresholded genes. Since these blocks consist of many hundreds of genes, it would surprisingly suggest that the differences between the four patient groups run into the hundreds, not dozens, of genes.

In spite of the differences in checkerboard patterns in Figure 7, the three left singular vectors are highly correlated between FIT-SSVD and LSHM: \( \text{corr} = 0.985, 0.981, \) and 0.968, respectively, and the top 20 genes with largest magnitude in the estimated three left singular vectors of FIT-SSVD overlap with those of LSHM except for one gene in the second singular vector. These shared performance aspects notwithstanding, the two methods differ hugely in computing time, FIT-SSVD taking seconds, LSHM taking a couple of hours.

4. DISCUSSION

We presented a procedure, called FIT-SSVD, for the fast and simultaneous extraction of singular vectors that are sparse in both the row and the column dimension. While the procedure is state of the art in terms of statistical performance, its overriding advantage is sheer speed. The reasons why speed matters are several: (a) faster algorithms enable the processing of larger datasets. (b) The use of SVDs in data analysis is most often for exploratory ends, which call for unlimited iterations of quickly improvised steps—something that is harder to achieve as datasets grow larger. (c) Sparse multivariate technology is still a novelty and hence at an experimental stage; if its implementation is fast, early adopters of the technology have a better chance to rapidly gain experience by experimenting with its parameters. (d) If a statistical method such as sparse SVD has a fast implementation, it can be incorporated as a building block in larger methodologies, for example, in processing data arrays that are more than two dimensional. For these reasons, we believe that fast SVD technology is of the essence for its success.

A unique opportunity for sparse approaches is to achieve faster speed than standard nonsparse approaches when the structure in the data is truly sparse. Our algorithm achieves this to some extent through initialization that is both sparse and smart: sparse initialization consists of a standard SVD of smaller size than the full data matrix, while smart
Figure 7. Cancer data: Image plots of the rank-three approximations $\sum_{l=1,2,3} \hat{u}_l \hat{v}_l'$ whose values are gray-coded. Each image is laid out as cases (= rows) by genes (= columns). The same rank-three approximation is shown three times for each method (left to right), each time sorted according to a different $\hat{u}_l$ ($l = 1, 2, 3$). (The mapping of the rank-three approximation values to gray scales is by way of a rank transformation, using a separate transformation for each image. Rank transformations create essentially uniform distributions that better cover the range of gray scale values.)
(in particular: nonrandom) initialization reduces the number of iterations to convergence. A statistical benefit is that inconsistent estimation by the standard SVD on large data matrices with weak signal is avoided—an imperative for fast implementations is avoiding where possible such slow devices as cross-validation. A considerable speed advantage we achieve is through relatively fast (noncross-validated) selection of thresholding levels based on an analytical understanding of their function.

Our proposal has conceptual and theoretical features that are unique at this stage of the development of the field: (a) FIT-SSVD extracts $r$ orthogonal left and right singular vectors simultaneously, which puts it more in line with standard linear dimension reduction where orthogonal data projections are the norm. In addition, simultaneous extraction can be cast as subspace extraction, which provides a degree of immunity to nonidentifiability and slow convergence of individual singular vectors when some of the first $r$ underlying singular values are nearly tied: since we measure convergence in terms of distance between successive $r$-dimensional subspaces, our algorithm does not need to waste effort in pinning down ill-determined singular vectors as long as the left and right singular subspaces are well determined. Such a holistic view of the rank-$r$ approximation is only available to simultaneous but not to successive extraction. (b) FIT-SSVD is derived from asymptotic theory that preceded its realization as a methodology: for Gaussian noise in the model (1), we (Yang, Ma, and Buja 2011) showed that our algorithm with appropriately chosen parameters achieves the rate of the minimax lower bound. In other words, in a specific parameter space, our algorithm is asymptotically optimal in terms of minimizing the maximum possible risk over the whole parameter space.

As for future work, the current state of the art raises several questions. For one, it would be of interest to better understand the relative merits of the currently proposed sparse SVD approaches since they have essential features in common, such as power iterations and thresholding. Another natural question arises from the fact that sparse SVDS build on the sequence model: many methods for choosing parameters from the data have been shown to be asymptotically equivalent to first order in the sequence model (see, e.g., Haerdle et al. 1988), including cross-validation, generalized cross-validation, Rice’s method based on unbiased estimates of risk, final prediction error, and the Akaike information criterion. Do these asymptotic equivalences hold in the matrix setting for sparse SVD approaches? How does the choice of the BIC in LSHM compare? Also, our algorithm and underlying theory allow a wide range of thresholding functions: Is there an optimal choice in some sense? Further, there exists still a partial disconnect between asymptotic theory and practical methodology: The theory requires a strict rank $r$ model, whereas by all empirical evidence the algorithm works well in a “trailing rank” situation where real but small singular values exist. Finally, there is a robustness aspect that is specific to sparse SVD approaches: heavier than normal tails in the noise distribution generate “random factors” caused by single outlying cells. While we think we have made reasonable and empirically successful choices in drawing from the toolkit of robustness, we have not provided a theoretical framework to justify them—just the same, even if the proposed FIT-SSVD algorithm may be subject to some future tweaking, in the substance it has the promise of lasting merit.
SUPPLEMENTARY MATERIALS

Appendix. Summary of the asymptotic minimax theorem and the simulation results. (supplement.pdf)

File ssvd.r. The R code implementing the algorithms introduced in this article.

Files mortality.txt and lung.txt. The data used in the real data section.

ACKNOWLEDGMENTS

Andreas Buja was partially supported by NSF grants DSM-1007689 and DSM-1007657. Dan Yang was partially supported by NSF grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute.

REFERENCES

Allen, G. I., Grosenick, L., and Taylor, J. (2011), “A Generalized Least Squares Matrix Decomposition,” Rice University Technical Report No. TR2011-03. [925]

Bach, F., Mairal, J., and Ponce, J. (2008), “Convex Sparse Matrix Factorizations,” CoRR, abs/0812.1869. [926]

Baik, J., and Silverstein, J. W. (2006), “Eigenvalues of Large Sample Covariance Matrices of Spiked Population Models,” Journal of Multivariate Analysis, 97, 1382–1408. [924]

Bickel, P. J., Gotze, F., and Van Zwet, W. R. (1997), “Resampling Fewer Than n Observations: Gains, Losses, and Remedies for Losses,” Statistica Sinica, 7, 1–32. [931]

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. [928]

Gabriel, K. R. (2002), “Le Biplot-Outil d’Exploration de Données Multidimensionnelles,” [“The Biplot-DNA Data Exploration Tool Multidimensional”] Journal de la Societe Francaise de Statistique, 143, 5–56. [930]

Golub, G. H., and Van Loan, C. F. (1996), Matrix Computations (3rd ed.), Baltimore, MD: Johns Hopkins University Press. [926]

Haerdle, W., Hall, P., and Marron, J. S. (1988), “How Far are Automatically Chosen Regression Smoothing Parameters From Their Minimum?” Journal of the American Statistical Association, 83, 86–95. [940]

Hoff, P. D. (2007), “Model Averaging and Dimension Selection for the Singular Value Decomposition,” Journal of the American Statistical Association, 102, 674–685. [930]

Holm, S. (1979), “A Simple Sequentially Rejective Multiple Test Procedure,” Scandinavian Journal of Statistics, 6, 65–70. [930]

Huang, J. Z., Shen, H., and Buja, A. (2009), “The Analysis of Two-Way Functional Data Using Two-Way Regularized Singular Value Decompositions,” Journal of the American Statistical Association, 104, 1609–1620. [925,935]

Johnstone, I. M. (2011), “Gaussian Estimation: Sequence and Multiresolution Models,” available at http://www-stat.stanford.edu/imj/. [925,933]

Johnstone, I. M., and Lu, A. Y. (2009), “On Consistency and Sparsity for Principal Components Analysis in High Dimensions,” Journal of the American Statistical Association, 104, 682–693. [924,925,929]

Lee, M., Shen, H., Huang, J. Z., and Marron, J. S. (2010), “Biclustering Via Sparse Singular Value Decomposition,” Biometrics, 66, 1087–1095. [925,926,932,935,936]
Liu, Y., Hayes, D. N. 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. [935]

Lu, A. Y. (2002), “Sparse Principal Component Analysis for Functional Data,” Ph.D. dissertation, Stanford University, Stanford, CA. [925]

Ma, Z. (2013), “Sparse Principal Component Analysis and Iterative Thresholding,” *The Annals of Statistics*, 41, 772–801. [924,925]

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010), “Online Learning for Matrix Factorization and Sparse Coding,” *Journal of Machine Learning Research*, 11, 19–60. [926]

Nadler, B. (2009), Discussion of “On Consistency and Sparsity for Principal Components Analysis in High Dimensions” by Johnstone and Lu, *Journal of the American Statistical Association*, 104, 694–697. [925]

Owen, A. B., and Perry, P. O. (2009), “Bi-Cross-Validation of the SVD and the Nonnegative Matrix Factorization,” *The Annals of Applied Statistics*, 3, 564–594. [930,932]

Paul, D. (2007), “Asymptotics of Sample Eigenstructure for a Large Dimensional Spiked Covariance Model,” *Statistica Sinica*, 17, 1617–1642. [924,925]

Paul, D., and Johnstone, I. M. (2007), “Augmented Sparse Principal Component Analysis for High Dimensional Data,” available at http://anson.ucdavis.edu/debashis/techrep/augmented-spca.pdf. [924,925]

Shabalin, A., and Nobel, A. (2010), “Reconstruction of a Low-Rank Matrix in the Presence of Gaussian Noise,” available at http://arxiv.org/abs/1007.4148. [925,927,928]

Shen, D., Shen, H., and Marron, J. S. (2013), “Consistency of Sparse PCA in High Dimension, Low Sample Size Contexts,” *Journal of Multivariate Analysis*, 115, 317–333. [925]

Shen, H., and Huang, J. H. (2008), “Sparse Principal Component Analysis Via Regularized Low Rank Matrix Approximation,” *Journal of Multivariate Analysis*, 99, 1015–1034. [925]

Sill, M., Kaiser, S., Benner, A., and Kopp-Schneider, A. (2011), “Robust Biclustering by Sparse Singular Value Decomposition Incorporating Stability Selection,” *Bioinformatics*, 27, 2089–2097. [925]

Witten, D. M., and Tibshirani, R. (2010), “A Framework for Feature Selection in Clustering,” *Journals of the American Statistical Association*, 105, 713–726. [925]

Witten, D. M., Tibshirani, R., and Hastie, T. (2009), “A Penalized Matrix Decomposition, With Applications to Sparse Principal Components and Canonical Correlation Analysis,” *Biostatistics*, 10, 515–534. [925,926,932]

Wold, S. (1978), “Cross-Validatory Estimation of the Number of Components in Factor and Principal Component Models,” *Technometrics*, 20, 397–405. [930]

Yang, D., Ma, Z., and Buja, A. (2011), “Near Optimal Sparse SVD in High Dimensions.” [926,940]

Zheng, W., Li, S. Z., Lai, J. H., and Liao, S. (2007), “On Constrained Sparse Matrix Factorization,” in *IEEE International Conference on Computer Vision*, pp. 1–8. [925]

Zou, H., Hastie, T., and Tibshirani, R. (2006), “Sparse Principal Component Analysis,” *Journal of Computational and Graphical Statistics*, 15, 265–286. [925]