Appendix to “A Sparse SVD Method for High-dimensional Data”

Dan Yang, Zongming Ma, Andreas Buja
University of Pennsylvania

Abstract

This supplement states the theoretical results and provides simulation results accompanying the paper “A Sparse SVD Method for High-dimensional Data”.

In this supplement, we provide a summary of the theoretical results related to the FIT-SSVD algorithm for Gaussian data and demonstrate the performance of FIT-SSVD through simulation results. To this end, Section A introduces the decision-theoretic framework. In addition, Section B gives the minimax lower bounds for the three loss functions introduced in Section 3.1 of the paper, together with upper bounds results showing that the FIT-SSVD algorithm attains near optimal rates of convergence for all three losses. Finally, Section C delivers the simulation.

A Problem Setup

For theoretical study, we focus on the following data generating model

$$X = \Xi + Z, \quad (1)$$

where the $p_u \times p_v$ signal matrix Ξ has low rank structure $\Xi = UDV' = \sum_{l=1}^r d_l \mathbf{u}_l \mathbf{v}_l'$ with non-increasing singular values $d_1 \geq \dots \geq d_r > 0$ and deterministic singular vector matrices $U = [\mathbf{u}_1, \dots, \mathbf{u}_r]$ and $V = [\mathbf{v}_1, \dots, \mathbf{v}_r]$, and the noise matrix Z has i.i.d. $N(0, 1)$ entries. Note that in the paper, we use (n, p) to denote the size of the matrix, whereas in the supplement, we switch to (p_u, p_v) for the ease of delivery of the results.

In addition, we assume that the singular vectors are sparse and use the weak l_q ball to quantify the level of sparsity. For any p -vector $\mathbf{u}$, we say that $\mathbf{u}$ belongs to a weak l_q ball of radius s , denoted by $\mathbf{u} \in wl_q(s)$, if

$$|u|_{(i)} \leq s i^{-1/q}, \quad \text{for } i = 1, \dots, p,$$

where $|u|_{(i)}$ the i -th largest coordinate of $\mathbf{u}$ in magnitude. When $0 < q < 2$, the weak l_q condition implies rapid decay of the ordered coefficients, and hence sparsity. Altogether, for $q_u, q_v \in (0, 2)$, we focus on parameter spaces characterized by parameters (s_u, q_u, s_v, q_v) as the following:

$$\begin{aligned} \Theta(s_u, q_u; s_v, q_v) = \{ \Xi = UDV' : U'U = I_r, V'V = I_r, D = \text{diag}(d_1, \dots, d_r) > 0, \\ \mathbf{u}_l \in wl_{q_u}(s_u), \mathbf{v}_l \in wl_{q_v}(s_v), \text{ for } l = 1, \dots, r \}. \end{aligned} \quad (2)$$

Last but not least, we consider the following loss functions as defined in Section 3.1 of the paper:

$$L_{space}(U, \hat{U}) = \|P_U - P_{\hat{U}}\|_2^2, \quad L_{space}(V, \hat{V}) = \|P_V - P_{\hat{V}}\|_2^2, \quad L(\Xi, \hat{\Xi}) = \|\hat{\Xi} - \Xi\|_F^2 / \|\Xi\|_F^2. \quad (3)$$

B Main Theoretical Results

We now state our main results. Their detailed proofs can be found in Yang (2012). For any real numbers a and b , let $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$.

Theorem 1 gives the minimax lower bounds for estimating the subspaces and the signal matrix.

Theorem 1 (Minimax lower bounds). *Under model (1) and parameter space (2) with $q_u, q_v \in (0, 2)$ and $s_u, s_v > 1$, for the loss functions (3), there exists a numeric constant $c > 0$, s.t., for any estimators $\tilde{U}, \tilde{V}, \tilde{\Xi}$,*

$$\inf_{\tilde{U}} \sup_{\Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_{\Xi} L_{space}(U, \tilde{U}) \geq c m_u \epsilon_u^2, \quad (4)$$

$$\inf_{\tilde{V}} \sup_{\Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_{\Xi} L_{space}(V, \tilde{V}) \geq c m_v \epsilon_v^2, \quad (5)$$

$$\inf_{\tilde{\Xi}} \sup_{\Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_{\Xi} L(\Xi, \tilde{\Xi}) \geq c(m_u \epsilon_u^2 \vee m_v \epsilon_v^2). \quad (6)$$

Here m_u and ϵ_u^2 are defined by

$$m_u = \min\{d_1^2, p_u, \bar{s}_u^{q_u} d_1^{q_u}\}, \quad \epsilon_u^2 = d_1^{-2},$$

with $\bar{s}_u^{q_u} = s_u^{q_u} - 1$. Moreover, if there exist numeric constants $K_1, K_2 > 0$ and $\alpha \in (0, 1)$, such that $\frac{\bar{s}_u^{q_u} d_1^{q_u}}{(\log(p_u \vee p_v))^{q_u/2}} \leq K_1 \frac{d_1^2}{\log(p_u \vee p_v)} \wedge K_2 p_u^\alpha$, then they can be replaced by

$$m_u = \frac{\bar{s}_u^{q_u} d_1^{q_u}}{(\log(p_u \vee p_v))^{q_u/2}}, \quad \epsilon_u^2 = \frac{\log(p_u \vee p_v)}{d_1^2}.$$

The quantities m_v and ϵ_v^2 are defined analogously.

The following theorem shows that the estimator of the signal matrix obtained from Algorithm 1 achieves the lower bound in Theorem 1 up to a log factor, and hence is near optimal. We note that the result depends on the normality of the noise matrix Z .

Theorem 2 (Rates of convergence for Algorithm 1). *Let $\hat{U}, \hat{V}$ be the output of Algorithm 1 with $V^{(0)}$ given by a slight modification of Algorithm 2 (see remarks below for details), and the function f in the thresholding step being $f(X, U, V, \hat{\sigma}) = \hat{\sigma} \sqrt{2 \log(p_u \vee p_v)}$. Define $\hat{\Xi} = \hat{U} \hat{D} \hat{V}'$, where $\hat{D} = \text{diag}(\hat{d}_1, \dots, \hat{d}_r)$ with $\hat{d}_l = \hat{\mathbf{u}}_l' X \hat{\mathbf{v}}_l$. If as both $p_u, p_v \rightarrow \infty$, r is fixed and*

$$s_u^{q_u} \left(\frac{\sqrt{p_u \log(p_u \vee p_v)}}{d_1^2} \right)^{1-q_u/2}, s_v^{q_v} \left(\frac{\sqrt{p_v \log(p_u \vee p_v)}}{d_1^2} \right)^{1-q_v/2} = o(1),$$

then there exists a constant $C > 0$, s.t., $\sup_{\Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_{\Xi} L_{space}(U, \hat{U})$, $\sup_{\Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_{\Xi} L_{space}(V, \hat{V})$, and $\sup_{\Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_{\Xi} L(\Xi, \hat{\Xi})$ are all upper bounded by

$$C \left(s_u^{q_u} \left(\frac{\log(p_u \vee p_v)}{d_1^2} \right)^{1-q_u/2} \vee s_v^{q_v} \left(\frac{\log(p_u \vee p_v)}{d_1^2} \right)^{1-q_v/2} \right). \quad (7)$$

Remark 1. Algorithm 2 consists of 3 steps, where Step 1 involves Huber ρ function and Holm method to tackle non-normal noise. In the case when the noise is normally distributed, which is assumed for theoretical analysis in our theorems, Step 1 (Subset Selection Step) can be simply replaced by the following:

$$\begin{aligned} I &= \{i : \frac{\sum_{j=1}^{p_v} x_{ij}^2}{p_v} \geq 1 + \alpha \sqrt{\frac{\log(p_u \vee p_v)}{p_v}}\}, \\ J &= \{j : \frac{\sum_{i=1}^{p_u} x_{ij}^2}{p_u} \geq 1 + \alpha \sqrt{\frac{\log(p_u \vee p_v)}{p_u}}\}, \end{aligned}$$

where α is large enough constant (see [9] for the details of the choice of α).

Remark 2. One may notice that the selections of subset of indices in Remark 1 and Algorithm 2 are similar to each other since both of them try to select those rows (or columns) that have large row (or column) norm. There are two differences. The first one is Remark 1 uses standard squared l_2 norm whereas Algorithm 2 Huberizes each entry of the matrix to accommodate heavy tailed distribution. The second one is the choice of the threshold: under the normality assumption in Remark 1, the threshold is set based on the tail of χ^2 distribution, whereas Algorithm 2 utilizes multiple testing procedure.

Although these two approaches have dissimilarities, we have tried extensive simulations on normal data, the two different initializations do not influence the final output of Algorithm 1.

C Simulation Results

In this section, we show the results of numerical experiments to compare the performance of FIT-SSVD with two state-of-the-art sparse SVD methods from the literature (as well as with the ordinary SVD). In contrast to FIT-SSVD which acquires whole subspaces spanned by sparse vectors simultaneously, both comparison methods are stepwise procedures that acquire sparse rank-one approximations $\hat{d}_l \hat{\mathbf{u}}_l \hat{\mathbf{v}}_l'$ successively; for example, the second rank-one approximation $\hat{d}_2 \hat{\mathbf{u}}_2 \hat{\mathbf{v}}_2'$ is found by applying the same method to the residual matrix $X - \hat{d}_1 \hat{\mathbf{u}}_1 \hat{\mathbf{v}}_1'$, and so on. For both methods it is therefore only necessary to describe how they obtain the first rank-one term.

- The first sparse SVD algorithm for comparison was proposed by Lee et al. [2] [referred to from here on by their initials, “LSHM”]. They obtain a first pair of sparse singular vectors by finding the solution to the following l_1 penalized SVD problem under an l_2 constraint:

$$\min_{\mathbf{u}, \mathbf{v}, s} (\|X - s\mathbf{u}\mathbf{v}'\|_F^2 + s\lambda_u \|\mathbf{u}\|_1 + s\lambda_v \|\mathbf{v}\|_1) \quad \text{subject to} \quad \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1.$$

LSHM solve this problem by alternating between the following steps till convergence:

$$\begin{aligned} (1) \quad & \tilde{\mathbf{u}}_l = \eta_{soft}(X\mathbf{v}_l^{old}, \lambda_u), \quad \mathbf{u}_l^{new} \leftarrow \frac{\tilde{\mathbf{u}}_l}{\|\tilde{\mathbf{u}}_l\|_2}, \\ (2) \quad & \tilde{\mathbf{v}}_l = \eta_{soft}(X'\mathbf{u}_l^{new}, \lambda_v), \quad \mathbf{v}_l^{new} \leftarrow \frac{\tilde{\mathbf{v}}_l}{\|\tilde{\mathbf{v}}_l\|_2}. \end{aligned}$$

- The second sparse SVD algorithm for comparison with our proposal is the adaptation of the penalized matrix decomposition scheme by Witten et al. [7] to the sparse SVD case [referred to as “PMD-SVD” from here on]. They obtain the first pair of sparse singular vectors by imposing simultaneous l_1 and l_2 constraints on both vectors:

$$\min \|X - d\mathbf{u}\mathbf{v}'\|_F^2, \quad \text{subject to } \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1, \|\mathbf{u}\|_1 \leq s_u, \|\mathbf{v}\|_1 \leq s_v.$$

The PMD-SVD algorithm iterates between the following two steps until convergence:

$$\begin{aligned} (1) \quad & \mathbf{u} = \frac{\eta_{soft}(X\mathbf{v}, \delta_u)}{\|\eta_{soft}(X\mathbf{v}, \delta_u)\|_2}, \quad \text{where } \delta_u \text{ is chosen by binary search such that } \|\mathbf{u}\|_1 = s_u, \\ (2) \quad & \mathbf{v} = \frac{\eta_{soft}(X'\mathbf{u}, \delta_v)}{\|\eta_{soft}(X'\mathbf{u}, \delta_v)\|_2}, \quad \text{where } \delta_v \text{ is chosen by binary search such that } \|\mathbf{v}\|_1 = s_v. \end{aligned}$$

To make fair comparisons, we use the implementations by their original authors for both LSHM [3] and PMD-SVD [6]. The tuning parameters are always chosen automatically by the default methods in their implementations. For FIT-SSVD, we always use $\eta = \eta_{hard}$ in Algorithm 1, Huberization $\beta = 0.95$ and Holm family-wise error rate $\alpha = 0.05$ in Algorithm 2, and $M = 100$ bootstraps in Algorithm 3. We did try different values of α , β and M in FIT-SSVD, and the results are not sensitive to these choices. Thus, in our experience there is no need for cross-validated selection of these parameters.

In what follows, we report simulation results for situations in which the true underlying matrix has rank one and two, respectively. Throughout this section, the rank of the true underlying matrix is assumed known.

C.1 Rank-one results

In this part, we generate data matrices according to model (1) with rank $r = 1$, $n = 1024$ and $p = 2048$, the singular value d_1 ranging in $\{50, 100, 200\}$, and iid noise $Z_{ij} \sim (\mu=0, \sigma^2=1)$. At first glance $d_1 = 50$ may appear like an outsized signal strength, but it actually is not: The expected sum of squares of noise is $\mathbb{E}[\|Z\|_F^2] = np \approx 2$ million, whereas the sum of squares of signal is a comparably vanishing $\|d_1\mathbf{u}_1\mathbf{v}_1'\|_F^2 = d_1^2 = 2500$, for a signal-to-noise ratio $S/N = 0.0012$ (which makes the failure of the ordinary SVD in these tasks less surprising). Even $d_1 = 200$ amounts to a $S/N = 0.012$ only.

As mentioned in the introduction, the FIT-SSVD method was motivated by theoretical results that were based on Gaussian assumptions [8]; it is therefore a particular concern to check the robustness of the method under noise with heavier tails than Gaussian. To this end we report simulation results both for $N(0, 1)$ and $\sqrt{3/5}t_5$ noise, the latter also having unit variance (the purpose of the factor $\sqrt{3/5}$).

For the construction of meaningful singular vectors we use a functional data analysis context: We choose functions gleaned from the literature and represent them in wavelet bases where they feature realistic degrees of sparsity. In Figure 1, Plot (a) (“**peak**”) shows the graph of a function with three peaks, evaluated at 1024 equispaced locations, while Plot (b) (“**poly**”) shows a piecewise polynomial function, evaluated at 2048 equispaced locations. Both functions create dense evaluation vectors but sparse wavelet coefficient vectors. [In all simulation results reported below, we use **Symmelet** 8 wavelet coefficients [5].] Multi-resolution plots of the wavelet coefficients are shown in Plots (c) (“**wc-peak**”) and (d) (“**wc-poly**”) of Figure 1. We choose $\mathbf{u}_1$ and $\mathbf{v}_1$ to be the wavelet coefficient vectors **wc-peak** and **wc-poly**, respectively.

Figure 1: (a) **peak**: three-peak function evaluated at 1024 equispaced locations; (b) **poly**: piecewise polynomial function evaluated at 2048 equispaced locations; (c) **wc-peak**: discrete wavelet transform (DWT) of the three-peak function; (d) **wc-poly**: DWT of the piecewise polynomial function. In Plot (c) and (d), each vertical bar is proportional in length to the magnitude of the **Symmetlet** 8 wavelet coefficient at the given location and resolution level.

For each simulated scenarios, we ran 100 simulations, applied each algorithm under comparison, and summarized the results in terms of median and MAD-based standard error. The criteria which we use for comparison of the methods are best explained with reference to Table 1, where we report the results for iid $N(0, 1)$ noise Z :

- The first block examines the estimation accuracy of the left singular vector, with the three rows corresponding to three different values of d_1 . Following Ma [4], we define the loss function for estimating the column space of U for a general rank- r by $L_{space}(U, \hat{U}) = \|P_U - P_{\hat{U}}\|_2^2$, where $P_U = UU'$ is the projection matrix onto the subspace spanned by the columns of U (which is of size $n \times r$ and has orthonormal columns, $U'U = I_r$). See Golub and Van Loan

losses	d_1	FIT-SSVD		LSHM		PMD-SVD		SVD	
		median	MAD	median	MAD	median	MAD	median	MAD
$L_{space}(\mathbf{u}_1, \hat{\mathbf{u}}_1)$	50	0.0513	0.0009	0.0669	0.0014	0.0783	0.0007	0.5225	0.0034
	100	0.0127	0.0003	0.0159	0.0004	0.0254	0.0002	0.1114	0.0005
	200	0.0036	0.0001	0.0044	0.0001	0.0102	0.0000	0.0264	0.0001
$L_{space}(\mathbf{v}_1, \hat{\mathbf{v}}_1)$	50	0.0958	0.0008	0.1095	0.0016	0.1399	0.0008	0.6330	0.0025
	100	0.0325	0.0004	0.0385	0.0005	0.0566	0.0003	0.1878	0.0006
	200	0.0112	0.0001	0.0131	0.0002	0.0241	0.0001	0.0499	0.0001
$L(\Xi, \hat{\Xi})$	50	0.1454	0.0014	0.1726	0.0019	0.3280	0.0016	2.2217	0.0082
	100	0.0457	0.0004	0.0549	0.0007	0.0973	0.0003	0.3709	0.0009
	200	0.0149	0.0001	0.0177	0.0003	0.0364	0.0001	0.0805	0.0002
$\ \hat{\mathbf{u}}_1\ _0$	50	24	0.1483	22	0.2965	242.5	1.4085	1024	0
	100	34	0.1483	32	0.2965	372.5	1.4085	1024	0
	200	43	0.1483	41	0.2965	577.0	1.3343	1024	0
$\ \hat{\mathbf{v}}_1\ _0$	50	18	0.2965	14	0.2965	535.0	2.2239	2048	0
	100	40	0.2965	38	0.4448	854.5	1.7050	2048	0
	200	66	0.4448	64	0.6672	1303.0	2.0756	2048	0
time	50	0.3364	0.0096	36.7316	0.4497	2.0578	0.0129	1	0
	100	0.4401	0.0209	30.8268	0.3305	1.9607	0.0124	1	0
	200	0.5685	0.0360	23.7639	0.2542	1.9274	0.0102	1	0

Table 1: Comparison of four methods in the rank-one case: $\mathbf{u}_1$ is **wc-peak**, $\mathbf{v}_1$ is **wc-poly**, and the noise is iid $N(0,1)$.

[1] for the details of such distance measures. In the rank-one case here, the loss reduces to $\sin^2 \angle(\mathbf{u}_1, \hat{\mathbf{u}}_1)$.

- The second block in Table 1 reports the loss for right singular vectors.
- The third block shows the scaled recovery error for the low-rank signal matrix $\Xi = UDV'$, defined as $L(\Xi, \hat{\Xi}) = \|\hat{\Xi} - \Xi\|_F^2 / \|\Xi\|_F^2$. Here, $\hat{\Xi} = \hat{\mathbf{U}}\hat{\mathbf{D}}\hat{\mathbf{V}}'$ and $\hat{\mathbf{D}} = \text{diag}(\hat{d}_1, \dots, \hat{d}_r)$ with diagonal entries being $\hat{d}_l = \hat{\mathbf{u}}_l' X \hat{\mathbf{v}}_l$.
- The fourth and fifth panels of Table 1 show the sparsity of the solutions measured by $\|\hat{\mathbf{u}}_1\|_0$ and $\|\hat{\mathbf{v}}_1\|_0$, that is, the number of nonzero elements in the estimates.
- The last block shows timing results as a fraction or multiple of the ordinary SVD.

We remark that the three loss functions used in the first three blocks are the same as those we use in the theoretical study. For statements on the minimax rates regarding these losses, we refer the reader to the online appendix.

The results are as follows:

- From the first three blocks we see that FIT-SSVD uniformly outperforms the other methods with respect to the three statistical criteria. While LSHM is not far behind FIT-SSVD, PMD-SVD lags in several cases by a factor of two or more. The ordinary SVD fails entirely for low signal strength as the results for $d_1 = 50$ illustrate, impressing the need to leverage sparsity in such situations. Rather expectedly, all methods achieve better statistical accuracy as the signal strength d_1 increases
- As for the sparsity metrics, FIT-SSVD and LSHM produce similar levels of sparsity, while PMD-SVD estimators are much denser. The results also suggest that as the signal strength d_1 gets stronger, the three sparse SVD methods estimate more coordinates.
- Finally, the timing results indicate that FIT-SSVD is faster than all other methods, the ordinary SVD included. LSHM stands out as slower than FIT-SSVD by factors of over 40 to over 100. PMD-SVD is more competitive but still at least a factor of three slower than FIT-SSVD. The variation in time for PMD-SVD is small because the majority is spent in cross-validation.

To examine the effect of heavy-tailed noise, we report in Table 2 the simulation results when the entries of the noise matrix Z are distributed iid $\sqrt{3/5} t_5$, all else being the same as in Table 1. [Recall that the scaling factor $\sqrt{3/5}$ is used to ensure unit variance.] The statistical performance for all methods is worse than in Table 1. In terms of the statistical metrics, the performances of FIT-SSVD and LSHM are in a statistical dead heat, whereas PMD-SVD trails behind by as much as a factor of two in the case of high signal strength, $d_1 = 200$. Again, FIT-SSVD and LSHM have comparable sparsities, whereas PMD-SVD is much denser. In terms of computation time, again FIT-SSVD is uniformly fastest, followed by PMD-SVD which trails by factors of over two to over five, and LSHM being orders of magnitude slower (by factors of 28 to 110).

losses	d_1	FIT-SSVD		LSHM		PMD-SVD		SVD	
		median	MAD	median	MAD	median	MAD	median	MAD
$L_{space}(\mathbf{u}_1, \hat{\mathbf{u}}_1)$	50	0.0802	0.0015	0.0819	0.0017	0.0907	0.0011	0.5405	0.0037
	100	0.0177	0.0003	0.0180	0.0004	0.0282	0.0003	0.1115	0.0006
	200	0.0048	0.0001	0.0047	0.0001	0.0108	0.0001	0.0262	0.0002
$L_{space}(\mathbf{v}_1, \hat{\mathbf{v}}_1)$	50	0.1193	0.0014	0.1191	0.0018	0.1560	0.0014	0.6432	0.0039
	100	0.0451	0.0005	0.0415	0.0007	0.0601	0.0003	0.1870	0.0006
	200	0.0145	0.0002	0.0137	0.0002	0.0249	0.0001	0.0498	0.0002
$L(\Xi, \hat{\Xi})$	50	0.1944	0.0024	0.1937	0.0028	0.3719	0.0028	2.2624	0.0107
	100	0.0625	0.0007	0.0600	0.0009	0.1041	0.0005	0.3706	0.0011
	200	0.0192	0.0002	0.0187	0.0002	0.0378	0.0001	0.0805	0.0002
$\ \hat{\mathbf{u}}_1\ _0$	50	20	0.2965	21.5	0.3706	235.0	1.5567	1024	0
	100	31	0.1483	33.0	0.2965	364.0	1.8532	1024	0
	200	40	0.1483	41.0	0.2965	569.5	2.0015	1024	0
$\ \hat{\mathbf{v}}_1\ _0$	50	13	0.1483	14.0	0.2965	526.0	2.0015	2048	0
	100	31	0.2965	38.5	0.5189	841.5	2.1498	2048	0
	200	56	0.2965	64.5	0.6672	1307.5	2.2980	2048	0
time	50	0.3714	0.0190	41.0695	0.9339	2.0826	0.0046	1	0
	100	0.8072	0.0652	30.5216	0.3774	2.0187	0.0039	1	0
	200	0.8238	0.0710	23.0527	0.2554	1.9520	0.0048	1	0

Table 2: Comparison of four methods in the rank-one case: $\mathbf{u}_1$ is **wc-peak**, $\mathbf{v}_1$ is **wc-poly**, and the noise is iid $\sqrt{3/5} t_5$.

C.2 Rank-two results

We show next simulation results for data according to model (1) with $r = 2$, and again $n = 1024$ and $p = 2048$. The singular values (d_1, d_2) range among the pairs $(100, 50)$, $(200, 50)$, and $(200, 100)$. The singular vectors are $\mathbf{u}_1 = \mathbf{wc}\text{-peak}$, $\mathbf{v}_1 = \mathbf{wc}\text{-poly}$, $\mathbf{u}_2 = \mathbf{wc}\text{-step}$, and $\mathbf{v}_2 = \mathbf{wc}\text{-sing}$, the properties of the latter two vectors being shown in Figure 2.

Figure 2: (a)**step**: step function evaluated at 1024 equispaced locations, (b)**sing**: single singularity function evaluated at 2048 equispaced locations, (c)**wc-step**: DWT of step function, (d)**wc-sing**: DWT of single singularity function.

Table 3 reports the results from 100 repetitions when the noise is iid $N(0,1)$. In terms of statistical metrics, FIT-SSVD always outperforms LSHM though not hugely. PMD-SVD does slightly better than FIT-SSVD for $L_{space}(U, \hat{U})$, but much worse for $L_{space}(V, \hat{V})$ and $L(\Xi, \hat{\Xi})$. This is due to the special type of cross-validation used in the package **PMA**: the parameters s_u, s_v are set to be proportional to each other after being scaled according to the dimensionality, $\sqrt{n}$ and $\sqrt{p}$, which essentially reduces the simultaneous cross-validation on two parameters to one. Therefore, PMD-SVD actually enforces the same level of sparsity on $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$.

In terms of sparsity of the estimators, the fourth and fifth blocks show the cardinality of the joint support of the estimated singular vectors, which indicate that FIT-SSVD and LSHM are again about comparable, and PMD-SVD is much denser as in the rank-one case. [We do not compare the losses and the l_0 norms for individual singular vectors because LSHM and PMD-SVD do not produce orthogonal singular vectors.]

Finally, in terms of computation time, FIT-SSVD dominates again, and the differences become somewhat more pronounced than in the rank-one case. In the high signal scenario, $d_1 = 200$ and $d_2 = 100$, FIT-SSVD gets a boost because by avoiding the costly bootstrap in Algorithm 3 because

losses	d_1	d_2	FIT-SSVD		LSHM		PMD-SVD		SVD	
			median	MAD	median	MAD	median	MAD	median	MAD
$L_{space}(U, \hat{U})$	100	50	0.1163	0.0010	0.1413	0.0021	0.1022	0.0009	0.5315	0.0037
	200	50	0.1148	0.0013	0.1422	0.0018	0.1007	0.0009	0.5265	0.0027
	200	100	0.0376	0.0003	0.0443	0.0006	0.0321	0.0003	0.1114	0.0005
$L_{space}(V, \hat{V})$	100	50	0.0514	0.0009	0.0596	0.0010	0.1230	0.0008	0.6376	0.0029
	200	50	0.0506	0.0009	0.0601	0.0011	0.1259	0.0006	0.6293	0.0023
	200	100	0.0144	0.0002	0.0172	0.0003	0.0538	0.0002	0.1870	0.0005
$L(\Xi, \hat{\Xi})$	100	50	0.0691	0.0006	0.0825	0.0007	0.1403	0.0004	0.7439	0.0017
	200	50	0.0234	0.0001	0.0285	0.0002	0.0529	0.0001	0.2070	0.0005
	200	100	0.0228	0.0001	0.0261	0.0002	0.0483	0.0001	0.1387	0.0003
$ supp(\hat{\mathbf{u}}_1) \cup supp(\hat{\mathbf{u}}_2) $	100	50	49	0.2965	45.0	0.4448	479	1.1861	1024	0
	200	50	56	0.2965	49.0	0.2965	649	1.5567	1024	0
	200	100	77	0.2965	73.5	0.5189	657	1.3343	1024	0
$ supp(\hat{\mathbf{v}}_1) \cup supp(\hat{\mathbf{v}}_2) $	100	50	54	0.2965	50.5	0.3706	1158.0	2.2239	2048	0
	200	50	78	0.2965	74.5	0.5189	1486.5	2.1498	2048	0
	200	100	81	0.4448	82.5	0.5930	1623.0	2.1498	2048	0
time	100	50	1.1675	0.0829	64.7840	0.6037	2.7991	0.0141	1	0
	200	50	1.4572	0.1011	55.6839	0.5436	2.7018	0.0142	1	0
	200	100	0.8000	0.0668	54.2361	0.2363	2.6429	0.0073	1	0

Table 3: Comparison of four methods for the rank-two case, with noise iid $N(0, 1)$.

Condition 2 is satisfied and the much cheaper normal approximation on Line 3 of Algorithm 3 can be used to compute the threshold level. Since LSHM repeats its scheme on the residual matrix to get the second layer of SVD, computation time doubles. As for PMD-SVD, since the time is mainly spent in cross-validation and the same penalty parameter is used for different ranks, the increase in time is not obvious.

References

- [1] G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, 1996. ISBN 0801854148.
- [2] M. Lee, H. Shen, J. Z. Huang, and J. S. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66:1087–1095, 2010.
- [3] M. Lee, H. Shen, J. Z. Huang, and J. S. Marron. R code for LSHM, 2010. URL <http://www.unc.edu/~haipeng/publication/ssvd-code.rar>.
- [4] Z. Ma. Sparse principal component analysis and iterative thresholding. 2011.
- [5] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 2009.

- [6] D. Witten, R. Tibshirani, and S. Gross. *PMA: Penalized Multivariate Analysis*, 2010. URL <http://CRAN.R-project.org/package=PMA>. R package version 1.0.7.
- [7] D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10: 515–534, 2009.
- [8] D. Yang, Z. Ma, and A. Buja. Near optimal sparse SVD in high dimensions. 2011.
- [9] Dan Yang. Singular value decomposition for high dimensional data. dissertation, university of pennsylvania.