1-1-2012

Singular Value Decomposition for High Dimensional Data

Dan Yang
University of Pennsylvania, sylria@gmail.com

Singular Value Decomposition for High Dimensional Data

Abstract
Singular value decomposition is a widely used tool for dimension reduction in multivariate analysis. However, when used for statistical estimation in high-dimensional low rank matrix models, singular vectors of the noise-corrupted matrix are inconsistent for their counterparts of the true mean matrix. We suppose the true singular vectors have sparse representations in a certain basis. We propose an iterative thresholding algorithm that can estimate the subspaces spanned by leading left and right singular vectors and also the true mean matrix optimally under Gaussian assumption. We further turn the algorithm into a practical methodology that is fast, data-driven and robust to heavy-tailed noises. Simulations and a real data example further show its competitive performance. The dissertation contains two chapters. For the ease of the delivery, Chapter 1 is dedicated to the description and the study of the practical methodology and Chapter 2 states and proves the theoretical property of the algorithm under Gaussian noise.

Degree Type
Dissertation

Degree Name
Doctor of Philosophy (PhD)

Graduate Group
Statistics

First Advisor
Andreas Buja

Second Advisor
Zongming Ma

Keywords
Cross validation, Denoise, Low rank matrix approximation, PCA, Penalization, Thresholding

Subject Categories
Statistics and Probability

SINGULAR VALUE DECOMPOSITION FOR HIGH DIMENSIONAL DATA

Dan Yang

A DISSERTATION

in

Statistics

For the Graduate Group in Managerial Science and Applied Economics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2012

Supervisor of Dissertation

_____

Andreas Buja, The Liem Sioe Liong/First Pacific Company Professor; Professor of Statistics

Graduate Group Chairperson

_____

Eric Bradlow, K.P. Chao Professor, Marketing, Statistics and Education

Dissertation Committee
Zongming Ma, Assistant Professor of Statistics
Dylan Small, Associate Professor of Statistics
Mark Low, Walter C. Bladstrom Professor; Professor of Statistics

SINGULAR VALUE DECOMPOSITION FOR HIGH DIMENSIONAL DATA

DEDICATED TO

My parents Youzhi Yang and Chunjing Zhu

My husband Dong Wang

ACKNOWLEDGMENT

First and foremost, I would like to offer my deepest gratitude to my academic advisors Professors Andreas Buja and Zongming Ma. It is my great fortune to have them as my advisors for that they nurture me with not only statistics but also life's philosophy. They are my friends as well as mentors. I can always rely on them whenever I need and wherever I am. I feel myself the luckiest person on the planet. Seriously, without them, would I be a totally different human being. The thought that they are no longer by my side makes my eyes full of tears.

I would like to thank the other members of my dissertation committee: Professors Dylan Small and Mark Low. It is so hard to summarize what they have done for me in a few sentences. Please allow me to take a note here to thank them later.

I also want to thank Professor Larry Brown and Linda Zhao for giving me the help that is most wanted, and for giving me advices that are extremely valuable.

I am also grateful to all the faculty members, staffs, and my fellow students in the Department of Statistics at University of Pennsylvania. Because of you, my life has been colorful for the past five years.

Lastly, Mom, Dad, and Dong, thank you for making my life special and sharing every moment of it with me. I love you.

ABSTRACT

SINGULAR VALUE DECOMPOSITION FOR HIGH DIMENSIONAL DATA

Dan Yang

Andreas Buja and Zongming Ma

Singular value decomposition is a widely used tool for dimension reduction in multivariate analysis. However, when used for statistical estimation in high-dimensional low rank matrix models, singular vectors of the noise-corrupted matrix are inconsistent for their counterparts of the true mean matrix. We suppose the true singular vectors have sparse representations in a certain basis. We propose an iterative thresholding algorithm that can estimate the subspaces spanned by leading left and right singular vectors and also the true mean matrix optimally under Gaussian assumption. We further turn the algorithm into a practical methodology that is fast, data-driven and robust to heavy-tailed noises. Simulations and a real data example further show its competitive performance. The dissertation contains two chapters. For the ease of the delivery, Chapter 1 is dedicated to the description and the study of the practical methodology and Chapter 2 states and proves the theoretical property of the algorithm under Gaussian noise.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

METHODOLOGY

1.1 Introduction

Singular value decompositions (SVD) and principle component analyses (PCA) are the foundations for many applications of multivariate analysis. They can be used for dimension reduction, data visualization, data compression and information extraction by extracting the first few singular vectors or eigenvectors; see, for example, Alter et al. (2001), Prasantha et al. (2007), Huang et al. (2009), Thomasian et al. (1998). In recent years, the demands on multivariate methods have escalated as the dimensionality of data sets has grown rapidly in such fields as genomics, imaging, financial markets. A critical issue that has arisen in large datasets is that in very high dimensional settings classical SVD and PCA can have poor statistical properties (Shabalin and Nobel 2010, Nadler 2009, Paul 2007, and Johnstone and Lu 2009). The reason is that in such situations the noise can overwhelm the signal to such an extent that traditional estimates of SVD and PCA loadings are not even near the ballpark of the underlying truth and can there-

fore be entirely misleading. Compounding the problems in large datasets are the difficulties of computing numerically precise SVD or PCA solutions at affordable cost. Obtaining statistically viable estimates of eigenvectors and eigenspaces for PCA on high-dimensional data has been the focus of a considerable literature; a representative but incomplete list of references is Lu (2002), Zou et al. (2006), Paul (2007), Paul and Johnstone (2007), Shen and Huang (2008), Johnstone and Lu (2009), Shen et al. (2011), Ma (2011). On the other hand, overcoming similar problems for the classical SVD has been the subject of far less work, pertinent articles being Witten et al. (2009), Lee et al. (2010a), Huang et al. (2009) and Allen et al. (2011).

In the high dimensional setting, statistical estimation is not possible without the assumption of strong structure in the data. This is the case for vector data under Gaussian sequence models (Johnstone, 2011), but even more so for matrix data which require assumptions such as low rank in addition to 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). 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. 2010a and Sill et al. 2011). Going one step further, Witten and Tibshirani (2010) used sparsity to develop a novel form of hierarchical clustering.

So far we implied rather than explained that SVD and PCA approaches are not identical. Their commonality is that both apply to data that have the form of a data matrix $X = (x_{ij})$ of size $n \times p$. The main distinction is that the PCA model assumes the rows of X to be i.i.d. samples from a p-dimensional

multivariate distribution, whereas the SVD model assumes the rows $i = 1, 2, ..., n$ to correspond to a "fixed effects" domain such as space, time, genes, age groups, cohorts, political entities, industry sectors, ... This domain is expected to have near-neighbor or grouping structure that will be reflected in the observations x_{ij} in terms of smoothness or clustering as a function of the row domain. In practice, the applicability of either approach is often a point of debate (e.g., should a set of firms be treated as a random sample of a larger domain or do they constitute an enumeration of the domain of interest?), but in terms of practical results the analyses are often interchangeable because the points of difference between the SVD and PCA models are immaterial in the exploratory use of these techniques. The main difference between the models is that the SVD approach analyzes the matrix entries as structured *low-rank means plus error*, whereas the PCA approach analyzes the covariation between the column variables.

In modern developments of PCA, interest is focused on "functional" data analysis situations or on the analog of the "sequence model" (Johnstone, 2011) where the columns also correspond to a structured domain such as space, time, genes, ... It is only with this focus that notions of smoothness and sparseness in the column or row domain are meaningful. A consequence of this focus is the assumption that all entries in the data matrix have the same measurement scale and unit, unlike classical PCA where the columns can correspond to arbitrary quantitative variables with any mix of units. With identical measurement scales throughout the data matrix, it is meaningful to entertain decompositions of the data into signal and fully exchangeable noise:

$$X = \Xi + Z\,, \tag{1.1}$$

where $\Xi = (\xi_{ij})$ is an $n \times p$ matrix representing the signal and $Z = (z_{ij})$ is

an $n \times p$ random matrix representing the noise and consisting of i.i.d. errors as its components. In both PCA and SVD approaches, the signal is assumed to have a multiplicative low-rank structure: $\Xi = UDV' = \sum_{l=1}^{r} d_l \mathbf{u}_l \mathbf{v}_l'$, where for identifiability it is assumed that rank $r < \min(n, p)$, usually even "$\ll$" such as $r = 1,\ 2$ or 3. The difference between SVD and PCA is, using ANOVA language, that in the SVD approach both U and V represent fixed effects that can both be regularized with smoothness or sparsity assumptions, whereas in functional PCA U is a random effect. As indicated above, such regularization is necessary for large n and p because for realistic signal-to-noise ratios recovery of the true U and V may not be possible. — Operationally, estimation under sparsity is achieved through thresholding. In general, if both matrix dimensions are thresholded, one obtains sparse singular vectors of X; if only the second matrix dimension is thresholded, one obtains sparse eigenvectors of $X'X$, which amounts to sparse PCA.

A few recent papers propose sparsity approaches to the high dimensional SVD problem: Witten et al. (2009) introduced a matrix decomposition which constrains the l_1 norm of the singular vectors to impose sparsity on the solutions. Lee et al. (2010a) 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 1.3.] Allen et al. (2011) is a Lagrangian version of Witten et al. (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 $\hat{d}_1, \hat{\mathbf{u}}_1, \hat{\mathbf{v}}_1$ by either constraining the l_1 norm of $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{v}}_1$ or adding it as a penalty. To estimate $\hat{d}_2, \hat{\mathbf{u}}_2, \hat{\mathbf{v}}_2$ for a second rank-one term, they subtract the first term $\hat{d}_1 \hat{\mathbf{u}}_1 \hat{\mathbf{v}}_1'$ 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 et al. (2008), but these do not have the form of a SVD. In our simulations and data examples we use the proposals by Witten et al. (2009) and Lee et al. (2010a) for comparison.

Our approach is to estimate the subspaces spanned by the leading singular vectors simultaneously. As a result, our method yields sparse singular vectors that are orthogonal, unlike the proposals by Witten et al. (2009) and Lee et al. (2010a). 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. (2010a). In terms of computational speed, our method is faster by at least a factor of two compared to the more efficient of the two proposals, which is generally Witten et al. (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 also beats the conventional SVD in terms of speed.

Lastly, our method is grounded in asymptotic theory that comprises minimax results which we describe in Chapter 2. A signature of this 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. As do most asymptotic theories in this area, ours relies heavily 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 present chapter is organized as follows. Section 1.2 describes our method

for computing sparse SVDs. Section 1.3 shows simulation results to compare the performance of our method with that of Witten et al. (2009) and Lee et al. (2010a). Section 1.4 applies our and the competing methods to real data examples. Finally, Section 1.5 discusses the results and open problems.

1.2 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, Chapter 8), which is a straightforward generalization of the power method for computing higher-dimensional invariant subspaces of symmetric matrices. For an arbitrary rectangular matrix Ξ 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 Ξ and Ξ' (its transpose), respectively, each followed by orthnormalization, 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:

(1) Right-to-Left Multiplication:	$U^{(k),mul} = \Xi V^{(k-1)}$
(2) Left Orthonormalization with QR Decomposition:	$U^{(k)} R_u^{(k)} = U^{(k),mul}$
(3) Left-to-Right Multiplication:	$V^{(k),mul} = \Xi' U^{(k)}$
(4) Right Orthonormalization with QR Decomposition:	$V^{(k)} R_v^{(k)} = V^{(k),mul}$

(1.2)

The superscript $^{(k)}$ indicates the k'th iteration, and ^{mul} the generally non-orthonormal

intermediate result of multiplication. For $r = 1$, the QR decomposition step reduces to normalization. If Ξ 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.

1.2.1 The FIT-SSVD Algorithm: "Fast Iterative Thresholding for Sparse SVDs"

Unsurprisingly, the algorithm to be proposed here involves some form of thresholding, be it soft or hard or something inbetween. 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.

Input:
1. Observed data matrix X.
2. Target rank r.
3. Thresholding function η.
4. Initial orthonormal matrix $V^{(0)} \in \mathbb{R}^{p\times r}$.
5. Algorithm f to calculate the threshold level $\boldsymbol{\gamma} = f(X, U, V, \hat{\sigma})$ 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 $\hat{V} = V^{(\infty)}$.

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

repeat

  Right-to-Left Multiplication: $U^{(k),mul} = XV^{(k-1)}$.

  Left Thresholding: $U^{(k),thr} = (u_{il}^{(k),thr})$, with $u_{il}^{(k),thr} = \eta\left(u_{il}^{(k),mul}, \gamma_{ul}\right)$, where $\boldsymbol{\gamma}_u = f(X, U^{(k-1)}, V^{(k-1)}, \hat{\sigma})$.

  Left Orthonormalization with QR Decomposition: $U^{(k)}R_u^{(k)} = U^{(k),thr}$.

  Left-to-Right Multiplication: $V^{(k),mul} = X'U^{(k)}$.

  Right Thresholding: $V^{(k),thr} = (v_{jl}^{(k),thr})$, with $v_{jl}^{(k),thr} = \eta\left(v_{jl}^{(k),mul}, \gamma_{vl}\right)$, where $\boldsymbol{\gamma}_v = f(X', V^{(k-1)}, U^{(k)}, \hat{\sigma})$.

  Right Orthonormalization with QR Decomposition: $V^{(k)}R_v^{(k)} = V^{(k),thr}$.

until *Convergence*;

Algorithm 1: FIT-SSVD

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 1.2.2–1.2.4, 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 (1.2) 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 $\eta_{soft}(x, \gamma) = sign(x)(|x| - \gamma)_+$, hard-thresholding

with $\eta_{hard}(x, \gamma) = x1_{|x|>\gamma}$, as well as the thresholding function used in SCAD (Fan and Li, 2001). The parameter γ is called the threshold level. In Algorithm 1, we apply the same threshold level γ_{ul} (or γ_{vl}) to all the elements in the lth column of $U^{(k),mul}$ (or $V^{(k),mul}$, resp.). For more details on threshold levels, see Section 1.2.4.

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 (that is, $H'H = I$), let $P_H = HH'$ be the associated projection matrix. We stop after the kth iteration if $\max\{\|P_{U^{(k)}} - P_{U^{(k-1)}}\|_2^2, \|P_{V^{(k)}} - P_{V^{(k-1)}}\|_2^2\} \leq \epsilon$, where ϵ is a pre-specified tolerance level, chosen to be $\epsilon = 10^{-8}$ for the rest of this chapter. [$\|A\|_2$ denotes the spectral norm of A.]

1.2.2 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 nearly orthogonal to the true V and thus requires many iterations to accumulate sufficient power to converge. We propose therefore Algorithm 2 which 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

Input:
1. Observed data matrix X.
2. Target rank r.
3. Degree of "Huberization" β (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 α.

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

1 Subset selection:
Let δ be the β-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 ρ function: $\rho(x, \delta) = x^2$ if $|x| \leq \delta$ and $2\delta|x| - \delta^2$ otherwise.
Select a subset $I = \{i_1, i_2, ...\}$ of rows according to the next four steps:
– Let $t_i = \sum_{j=1}^{p} y_{ij}$ for $i = 1, \ldots, n$.
– Let $\hat{\mu} = \text{median}(t_1, ..., t_n)$ and $\hat{s} = 1.4826\ \text{MAD}(t_1, ..., t_n)$.
– Calculate p-values: $p_i = 1 - \Phi(\frac{t_i - \hat{\mu}}{\hat{s}})$, where Φ is the CDF of $N(0, 1)$.
– Perform the Holm method on the p-values at family-wise error rate α, 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 $\mathbf{u}_1^I, \ldots, \mathbf{u}_r^I$ ($|I| \times 1$ each) and $\mathbf{v}_1^J, \ldots, \mathbf{v}_r^J$ ($|J| \times 1$ each).

3 Zero-padding: Create $U^{(0)} = [\mathbf{u}_1^{(0)}, \ldots, \mathbf{u}_r^{(0)}]$ $(n \times r)$ and $V^{(0)} = [\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_r^{(0)}]$ $(p \times r)$, such that $\mathbf{u}_{Il}^{(0)} = \mathbf{u}_{Il}^I$, $\mathbf{u}_{I^c l}^{(0)} = 0$, $\mathbf{v}_{Jl}^{(0)} = \mathbf{v}_{Jl}^J$, $\mathbf{v}_{J^c l}^{(0)} = 0$.

Algorithm 2: Initialization algorithm for FIT-SSVD.

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 χ^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 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 χ^2 tests: they are robust to isolated large values, and they avoid the sensitivity of χ^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 α (default: 5%). The end result are 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.

1.2.3 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 to the full matrix. In simulations not reported here but similar to those of Section 1.3, 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.

1.2.4 Threshold levels

The tuning parameters γ 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 $l = 1, ..., r$ 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 $\mathbf{u}_l^{(k),mul} = X\mathbf{v}_l^{(k-1)}$. Recall that X is assumed to admit an additive decomposition into a low-rank signal plus noise according to model (1.1). Then a theoretically sensible (though not actionable) threshold level for $\mathbf{u}_l^{(k),mul}$ would be $\gamma_{ul} = \mathrm{E}[\|Z\mathbf{v}_l^{(k-1)}\|_\infty]$, where Z is the additive noise matrix, and $\|Z\mathbf{v}_l^{(k-1)}\|_\infty$ is the maximum absolute value of the n entries in the vector $Z\mathbf{v}_l^{(k-1)}$. The signal of any coordinate in $\mathbf{u}_l^{(k),mul}$ with value less than γ_{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 γ_{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 which 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:

$$X = \begin{pmatrix} X_{H_uH_v} & X_{H_uL_v} \\ X_{L_uH_v} & X_{L_uL_v} \end{pmatrix}. \tag{1.3}$$

Since the entries in $\mathbf{v}_l^{(k-1)}$ corresponding to L_v are zero, only $X_{:H_v}$ (of size $n \times |H_v|$, containing the two left blocks in (1.3)) 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: $X_{L_uL_v} \approx Z_{L_uL_v}$.

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 $\mathbf{v}_l^{(k-1)}$ we form $Z^* \mathbf{v}_{H_v l}^{(k-1)}$, which we interpret as an approximate draw from $Z\mathbf{v}_l^{(k-1)}$. We thus estimate $\|Z\mathbf{v}_l^{(k-1)}\|_\infty$ with $\|Z^*\mathbf{v}_{H_v l}^{(k-1)}\|_\infty$, and $\mathrm{E}[\|Z\mathbf{v}_l^{(k-1)}\|_\infty]$ with a median of $\|Z^*\mathbf{v}_{H_v l}^{(k-1)}\|_\infty$ 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 et al. (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 $\mathrm{E}[\|Z\mathbf{v}_1^{(k-1)}\|_\infty]$ 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 $Z\mathbf{v}_1^{(k-1)}$ 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 γ_{ul} to be used on $\mathbf{u}_l^{(k),mul}$. The thresholds for $l = 1, ..., r$ are then collected in the threshold vector $\boldsymbol{\gamma}_u = (\gamma_{u1}, ..., \gamma_{ur})'$.

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.

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. Pre-specified number M of bootstraps;
4. Estimate of the standard deviation of noise $\hat{\sigma}$.

Output: Threshold level $\boldsymbol{\gamma} \in \mathbb{R}^r$.

Subset selection: $L_u = \{i : u_{i1}^{(k)} = ... = u_{ir}^{(k)} = 0\}$,
$L_v = \{j : v_{j1}^{(k)} = ... = v_{jr}^{(k)} = 0\}$,
$H_u = L_u^c,\ H_v = L_v^c$;

if $|L_v||L_u| < n|H_v| \log(n|H_v|)$ **then**
  return $\boldsymbol{\gamma} = \hat{\sigma}\sqrt{2\log(n)}\mathbf{1} \in \mathbb{R}^r$;

else
  for $i \leftarrow 1$ **to** M **do**
    Sample $n|H_v|$ entries from $X_{L_u L_v}$ and reshape them into a matrix $\tilde{Z} \in \mathbb{R}^{n \times |H_v|}$;
    $B = \tilde{Z} V_{H_v:}^{(k)} \in \mathbb{R}^{n \times r}$;
    $C_{i:} = (\|B_{:1}\|_\infty, \|B_{:2}\|_\infty, \ldots, \|B_{:r}\|_\infty)'$;
  $\gamma_l = \text{median}(C_{:l})$;
  return $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_r)'$.

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.

1.2.5 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. To navigate this bias-variance trade-off, we adopted in Section 1.2.4 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 1.2.3 (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 et al. (2009) leave out random subsets of the entries in the data matrix, measure the differences between the fitted values and the original values for those entries, and choose the threshold levels that minimize the differences. Alternatively one could use bi-cross-validation (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. In order to avoid such issues, Lee et al. (2010a) implement their algorithm by embedding the optimization of the choice of the threshold level inside the iterations that calculate $\mathbf{u}_l$ for fixed $\mathbf{v}_l$ and $\mathbf{v}_l$ for fixed $\mathbf{u}_l$ (unlike our methods, theirs fits one rank at a time). They minimize a BIC criterion 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 1.3 below show that the method of Lee et al. (2010a) is computationally very slow.

1.3 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. (2010a) [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:

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

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

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

- The second sparse SVD algorithm for comparison with our proposal is the adaptation of the penalized matrix decomposition scheme by Witten et al. (2009) 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:

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

To make fair comparisons, we use the implementations by their original authors for both LSHM (Lee et al., 2010b) and PMD-SVD (Witten et al., 2010). 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.

1.3.1 Rank-one results

In this part, we generate data matrices according to model (1.1) 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

$\mathrm{E}[\|Z\|_F^2] = np \approx 2$ 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 (Yang et al., 2011); 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 1.1, 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 (Mallat, 2009).] Multi-resolution plots of the wavelet coefficients are shown in Plots (c) ("`wc-peak`") and (d) ("`wc-poly`") of Figure 1.1. We choose $\mathbf{u}_1$ and $\mathbf{v}_1$ to be the wavelet coefficient vectors `wc-peak` and `wc-poly`, respectively.

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 1.1, where we report the results for iid

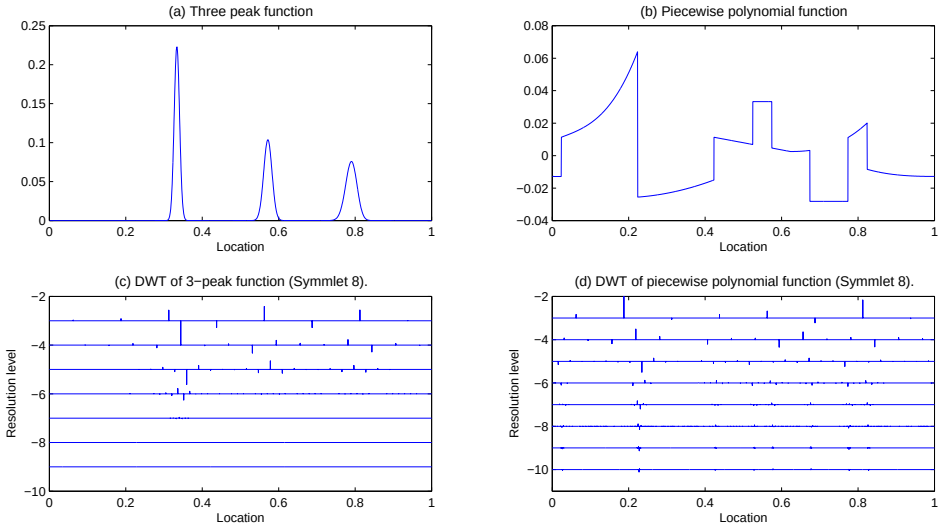

Figure 1.1: (a) `peak`: three-peak function evaluated at 1024 equispaced locations; (b) `poly`: piecewise polynomial function evaluated at 2048 equispaced locations; (c) `wc-peak`: discrete wavelet transform (DWT) of the three-peak function; (d) `wc-poly`: DWT of the piecewise polynomial function. In Plot (c) and (d), each vertical bar is proportional in length to the magnitude of the `Symmlet` 8 wavelet coefficient at the given location and resolution level.

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 1.1: 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).

$N(0,1)$ 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 (2011), 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, $U'U = I_r$). 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 1.1 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, $\hat{\Xi} = \hat{U}\hat{D}\hat{V}'$ and $\hat{D} = diag(\hat{d}_1, \ldots, \hat{d}_r)$ with diagonal entries being $\hat{d}_l = \hat{\mathbf{u}}_l' X \hat{\mathbf{v}}_l$.

- The fourth and fifth panels of Table 1.1 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.

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 1.2 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 1.1. [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 1.1. 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 1.2: 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 $\sqrt{3/5}\, t_5$.

1.3.2 Rank-two results

We show next simulation results for data according to model (1.1) 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 $\mathbf{u}_1 =$ `wc-peak`, $\mathbf{v}_1 =$`wc-poly`, $\mathbf{u}_2 =$ `wc-step`, and $\mathbf{v}_2 =$ `wc-sing`, the properties of the latter two vectors being shown in Figure 1.2.

Table 1.3 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 $\hat{\mathbf{u}}$

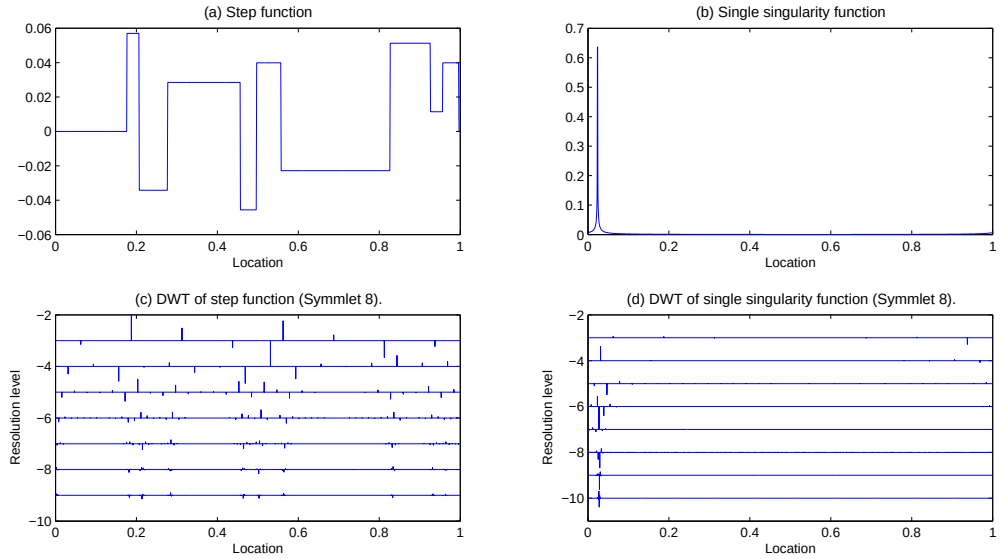

Figure 1.2: (a)`step`: step function evaluated at 1024 equispaced locations, (b)`sing`: single singularity function evaluated at 2048 equispaced locations, (c)`wc-step`: DWT of step function, (d)`wc-sing`: DWT of single singularity function.

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 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

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
$\lvert \cup_1^2 supp(\hat{\mathbf{u}}_l) \rvert$	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
$\lvert \cup_1^2 supp(\hat{\mathbf{v}}_l) \rvert$	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 1.3: Comparison of four methods for the rank-two case, and the noise is iid $N(0, 1)$.

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.

1.4 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).

1.4.1 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 to 110 from 1933 to 2007. The data for people older than 95 was discarded because of their noisy nature. The matrix X is of size 96×75, each row corresponding to an age group and each column to a one-year period. We first pre- and post- multiply 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 bi-crossvalidation (Section 1.2.3). We then applied FIT-SSVD, LSHM, PMD-SVD 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 1.4: none gives very sparse solutions. This is reasonable, because the mortality rate data is of low noise and for data with no noise we should just use the ordinary SVD. Because this data is of small size, it only takes a few seconds for all the algorithms. The plot of singular vectors for all the methods are shown in Figures 1.3 to 1.7. 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 1.3 shows the first left singular vector plotted against age. The curve $\hat{\mathbf{u}}_1$ 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

	FIT-SSVD	LSHM	PMD-SVD	SVD
$\|\hat{\mathbf{u}}_1\|_0$	82	48	96	96
$\|\hat{\mathbf{u}}_2\|_0$	86	56	7	96
$\|\hat{\mathbf{v}}_1\|_0$	66	45	75	75
$\|\hat{\mathbf{v}}_2\|_0$	70	45	43	75

Table 1.4: Mortality data: number of nonzero coordinates in the transformed domain for four methods.

begins an exponential increase. Figure 1.4 zooms into the lower left corner of Figure 1.3 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 1.5 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.

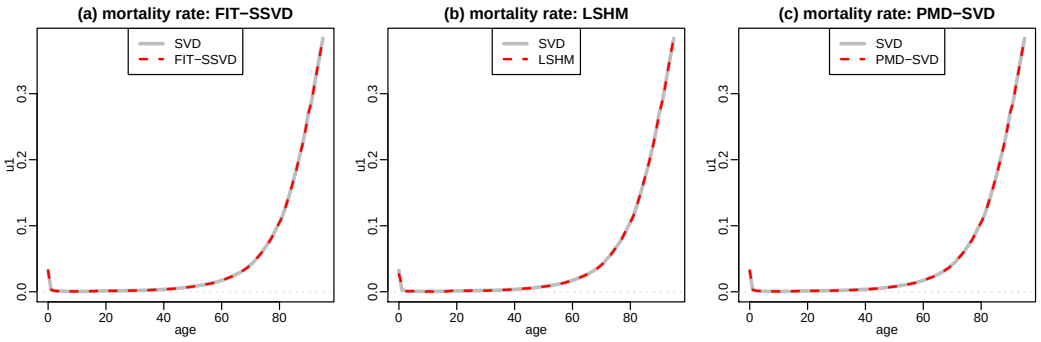

Figure 1.3: Mortality data: plot of $\hat{\mathbf{u}}_1$. Panel (a): FIT-SSVD vs. SVD; Panel (b): LSHM vs. SVD; Panel (c): PMD-SVD vs. SVD.

The second pair of singular vectors is shown in Figures 1.6 and 1.7: They

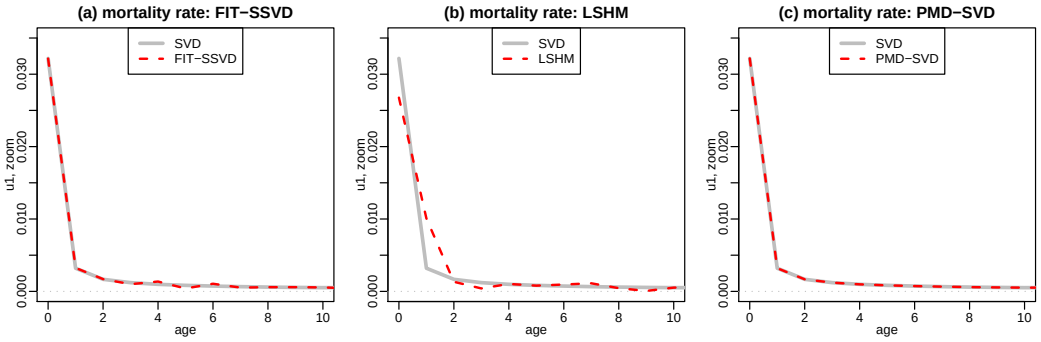

Figure 1.4: Mortality data: Plot of $\hat{\mathbf{u}}_1$. Zoom of the lower left corner of Figure 1.3. Everything else is the same as in Figure 1.3.

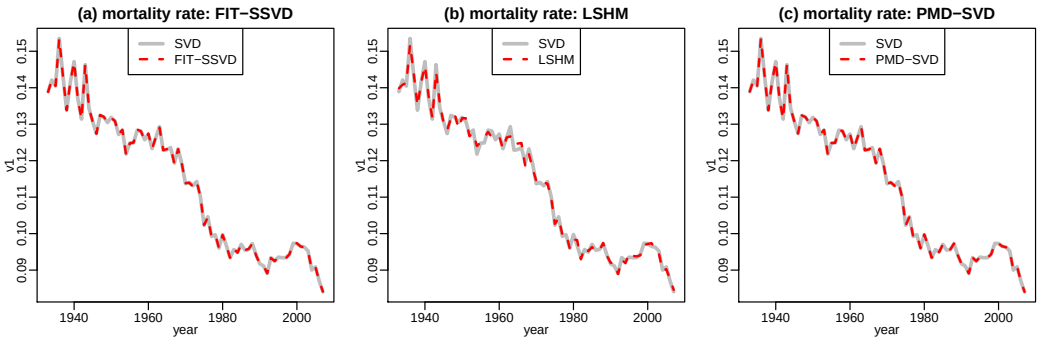

Figure 1.5: Mortality data: Plot of $\hat{\mathbf{v}}_1$. Everything else is the same as in Figure 1.3.

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 $\hat{\mathbf{u}}_2$ is positive. Also, $\hat{\mathbf{v}}_2$ has extreme negative or positive values towards 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 $\hat{\mathbf{u}}_{2,PMD-SVD}$, recalling that $\hat{\mathbf{u}}_{1,PMD-SVD}, \hat{\mathbf{v}}_{1,PMD-SVD}, \hat{\mathbf{v}}_{2,PMD-SVD}$ all follow the ordinary SVD very closely. We think this is again due to the cross-validation technique they use.

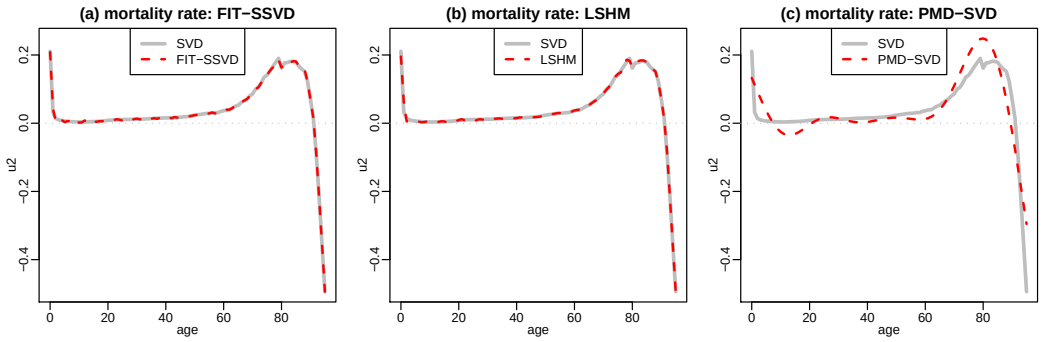

Figure 1.6: Mortality data: plot of $\hat{\mathbf{u}}_2$. Everything else is the same as Figure 1.3.

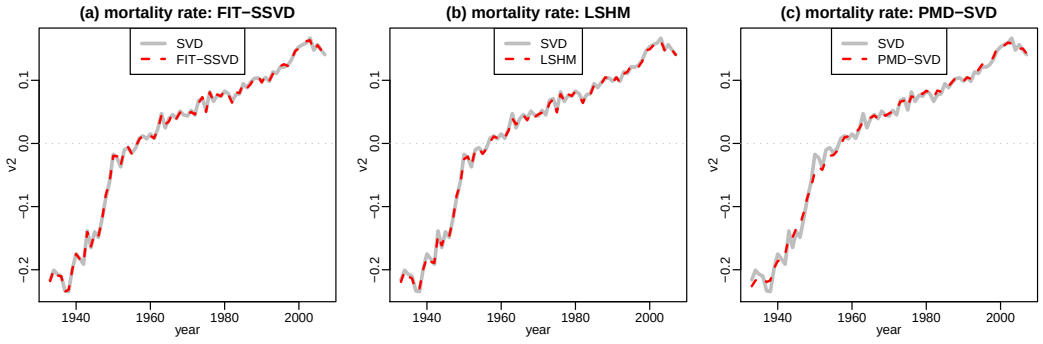

Figure 1.7: Mortality data: plot of $\hat{\mathbf{v}}_2$. Everything else is the same as Figure 1.3.

Huang et al. (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.

1.4.2 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. (2010a) (who in turn have them from Liu et al. (2008)) consists 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 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 PCA or SVD model is more appropriate. It might be argued that the cases really should be considered as being sampled from a population, hence PCA 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 SVD model seems less problematic than either of the dual PCA models.

	FIT-SSVD	LSHM	PMD-SVD	SVD
$\lvert \cup_{l=1}^{3} \text{supp}(\hat{\mathbf{u}}_l) \rvert$	4688	4545	12625	12625
$\lvert \cup_{l=1}^{3} \text{supp}(\hat{\mathbf{v}}_l) \rvert$	56	56	54	56

Table 1.5: Cancer data: summary of cardinality of joint support of three singular vectors for four methods.

We first attempted to estimate the rank of the signal using bi-crossvalidation (Section 1.2.3), but it turns out that the rank is sensitive to the choice of α (Holm family-wise error) and β (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. (2010a), 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 1.5 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 1.8 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 non-zero, 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 $n = 12625$. Yet, the three sparse methods give clearer evidence that the carcinoid group (black cirlces) falls into two subgroups than the ordinary SVD. According to FIT-SSVD and LSHM the separation is along $\hat{\mathbf{v}}_3$ (center and right hand plots), whereas according to PMD-SVD it is by lineup with $\hat{\mathbf{v}}1$ and $\hat{\mathbf{v}}2$, respectively (left hand plot).

Figure 1.9 shows checkerboard plots of the reconstructed rank-three approximations, layed 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 $\hat{\mathbf{u}}_1$, $\hat{\mathbf{u}}_2$ and $\hat{\mathbf{u}}_3$. 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 1.5). 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 1.8, there is a clear difference between these methods in the reconstructions in Figure 1.9: The FIT-SSVD solution exhibits the strongest block structure in its $\hat{\mathbf{u}}_2$-based sort (center plot, top row), implying the strongest evidence of clustering among its non-thresholded genes. Since these blocks consist of many hundreds of genes, it would surprisingly suggest that the differences

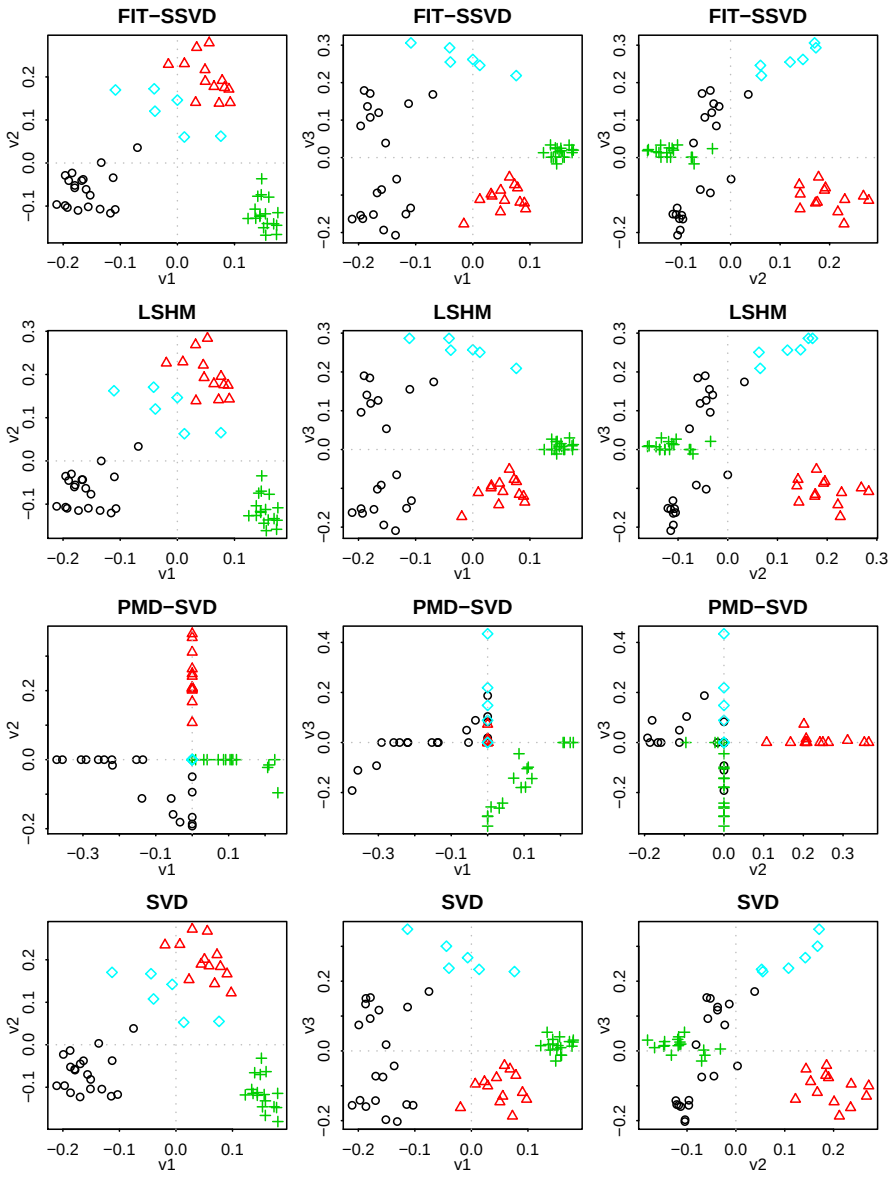

Figure 1.8: Cancer data: Scatterplots of the entries of the first three right singular vectors $\hat{\mathbf{v}}_l, l = 1, 2, 3$ for four methods. Points represent patients. Black circle: Carcinoid; Red triangle: Colon; Green cross: Normal; Blue diamond: SmallCell.

between the four patient groups run into the hundreds, not dozens, of genes.

In spite of the differences in checkerboard patterns in Figure 1.9, the three left singular vectors are highly correlated between FIT-SSVD and LSHM: 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.

1.5 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: (1) Faster algorithms enable the processing of larger datasets. (2) 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. (3) 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. (4) 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.

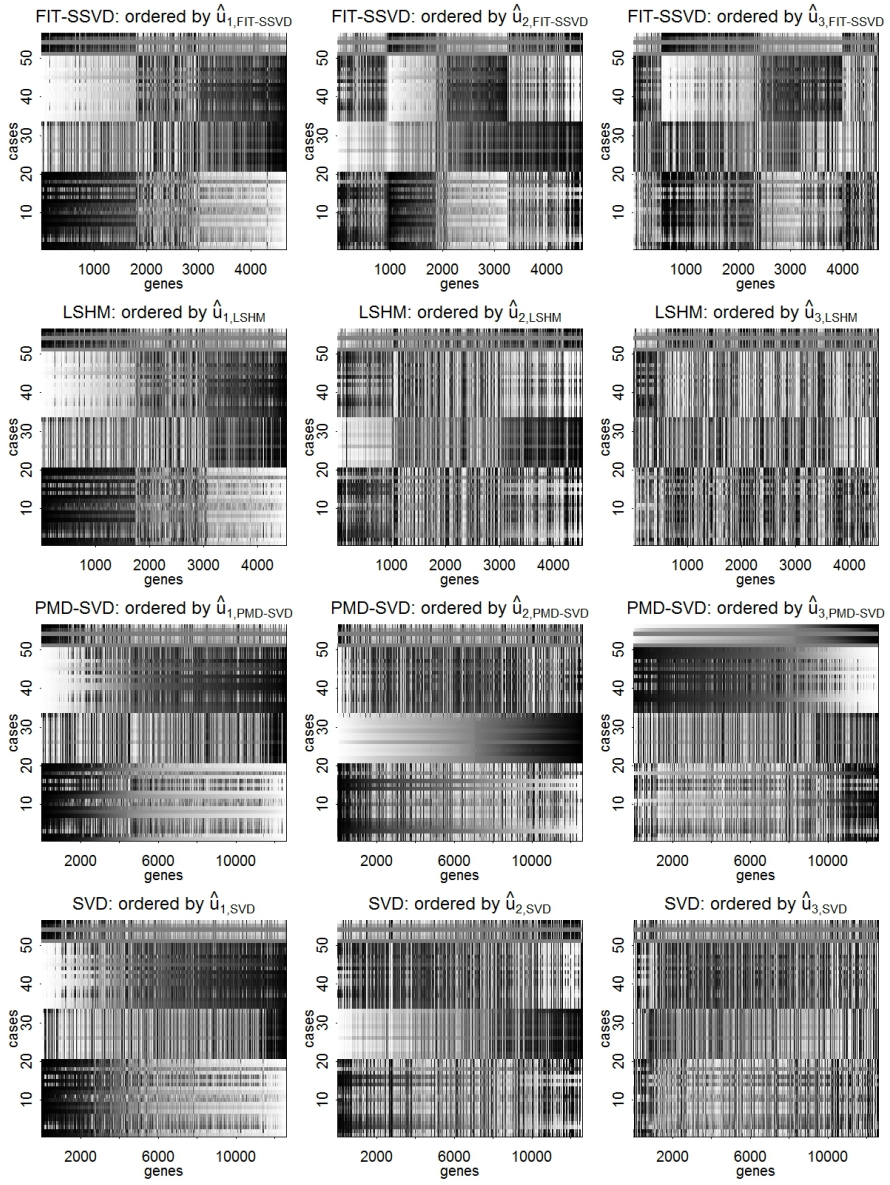

Figure 1.9: Cancer data: Image plots of the rank-three approximations $\sum_{l=1,2,3} \hat{d}_l \hat{\mathbf{u}}_l \hat{\mathbf{v}}_l'$ whose values are gray-coded. Each image is laid out as cases (= rows) by genes (= columns). The same rank-three approximation is shown three times for each method (left to right), each time sorted according to a different $\hat{\mathbf{u}}_l$ ($l = 1, 2, 3$). (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.)

A unique opportunity for sparse approaches is to achieve faster speed than standard non-sparse 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 (in particular: non-random) 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 (non-crossvalidated) 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: (1) 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 non-identifiability 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. (2) FIT-SSVD is derived from asymptotic theory that preceded its realization as a methodology: For Gaussian noise in the model (1.1), we (Yang et al., 2011) showed that our algorithm with appropriately chosen parameters achieves the rate of the mini-

max 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.

CHAPTER 2

THEORY

2.1 Introduction

Singular value decomposition (SVD) is widely used in multivariate analysis for dimension reduction, data visualization, data compression and information extraction in such fields as genomics, imaging, financial markets, etc. However, when used for statistical estimation in high-dimensional low rank matrix models, singular vectors of the noise-corrupted matrix are inconsistent for their counterparts of the true mean matrix Shabalin and Nobel (2010). To achieve consistency in estimation and better interpretability, in addition to low-rankness, we further assume that the true singular vectors have sparse representations in a certain basis.

Sparse SVD in high dimensions has been studied by several recent papers. Witten et al. (2009) introduced penalized matrix decomposition which constrains the l_1 norm of the left and right singular vectors to impose sparsity on the solutions. Lee et al. (2010a) used penalized least squares for rank-one matrix approximations

with l_1 norms of the singular vectors as additive penalties. Both papers focus on obtaining the first pair of singular vectors. The subsequent pairs are then obtained by repeating the same procedure on the residual matrices. This may cause non-identifiability and non-orthogonality issues, and theoretical properties of resulting estimators are not well understood.

The goal of this chapter is to provide a theoretically optimal and computationally efficient solution to the high dimensional SVD problem. In particular, we propose an iterative thresholding estimation procedure, which has the following distinctive features. First, it does not involve any optimization criterion and is based on a simple matrix computation method. Second, it estimates the subspaces spanned by the leading singular vectors simultaneously as well as the true mean matrix, as opposed to the previous one-pair-at-a-time methods. Hence, it yields orthogonal sparse singular vectors. Last but not least, under normality assumption, the resulting estimators achieve near optimal minimax rates of convergence and adaptivity.

2.2 Model

In this section, we formally give the basic model assumptions as well as the loss functions. We further describe the property of the classical SVD given the model assumptions, which motivates the introduction of the sparsity assumption on the singular vectors. Meanwhile, we clarify the relationship between the SVD and Principal Component Analysis (PCA).

2.2.1 Basic Model

We now lay out the model assumptions. To start with, we assume the data matrix is the sum of signal and fully exchangeable noise:

$$X = M + Z\,. \tag{2.1}$$

In 2.1, the signal matrix $M = (m_{ij})$ is of dimension p_u by p_v and has a multiplicative low-rank structure, i.e., the SVD: $M = UDV' = \sum_{l=1}^{r} d_l \mathbf{u}_l \mathbf{v}_l'$, where the singular values satisfy that $d_1 \geq \cdots \geq d_r$ and $1 \geq \frac{d_r}{d_1} \geq c_0 > 0$. The two orthogonal singular vector matrices $U = [\mathbf{u}_1, \ldots, \mathbf{u}_r]$, $V = [\mathbf{v}_1, \ldots, \mathbf{v}_r]$ are both deterministic. The rank $r \ll \min(p_u, p_v)$ is assumed to be fixed in later asymptotic analysis and known throughout. Moreover, the noise matrix $Z = (z_{ij})$ consists of i.i.d. $N(0, \sigma^2)$ errors as its components.

Convention: throughout this chapter, we will use $\|X\|$ to denote matrix spectral norm, $\|X\|_F$ Frobenius norm, and $\|\mathbf{u}\|$ vector l_2 norm.

2.2.2 Loss Functions

Depending on one's interest, there are two possible loss functions.

The first goal is, given the noise-corrupted matrix X, to denoise the observed data table and recover the true mean matrix M. To assess the performance of an estimator $\hat{M}$, an appropriate loss function is

$$L_M(\hat{M}, M) = \frac{\|\hat{M} - M\|_F^2}{\|M\|_F^2}\,, \tag{2.2}$$

which is scale invariant because of the existence of the denominator.

One may be also interested in estimating the singular vectors U, V since SVD is often used as the first step analysis and the followup research mainly relies on the singular vectors. A question immediately raises: should we estimate each singular vector or them as a whole? We decide to aim at the subspaces the singular vectors span for two reasons. First, if the singular values are identical, then the singular vectors are not identifiable, but the subspace they span is. Second, most of the applications focus on projection which depends only upon the subspace not each singular vector.

To measure the accuracy of such a subspace estimator, we consider the projection matrix which is uniquely defined for any given subspace. The distance between two subspaces is specified as the spectral norm of the difference between the true and the estimated projection matrices

$$L_U(\hat{U}, U) = \|P_{\hat{U}} - P_U\|^2 = \sin^2(\alpha)\,, \tag{2.3}$$

where α is the largest canonical angle between the subspaces U and $\hat{U}$. We define $L_V(\hat{V}, V)$ similarly. See Golub and Van Loan (1996) for the details of such distance measures. In the special case when $r = 1$, the loss function (2.3) is simply squares sine of the angle between two unit vectors. Note that the squared sine loss and the squared error loss for estimating a single vector is closely related: $\sin^2 \angle(\hat{\mathbf{u}}, \mathbf{u}) = \|\hat{\mathbf{u}} - sign(\langle\hat{\mathbf{u}}, \mathbf{u}\rangle)\mathbf{u}\|^2(2 - \|\hat{\mathbf{u}} - sign(\langle\hat{\mathbf{u}}, \mathbf{u}\rangle)\mathbf{u}\|^2/2)/2$. If one of the two losses is close to zero, then these two have approximately the same value by a factor of 2. Because of the equivalence of the loss functions, it is natural to see in the subsequent sections that the minimax convergence rate boils down to squared bias and variance. Note that by definition, (2.3) measures the largest possible

discrepancy between the projections of any unit vector onto the two subspaces, hence clearly we have the following relationship $L_U(\hat{U}, U) \geq \sin^2 \angle(\hat{\mathbf{u}}, \mathbf{u})$.

2.2.3 Connection with PCA

SVD and PCA have a lot in common and do differ from each other.

To begin with, their commonality is that both apply to data that have the form of a data matrix $X = (x_{ij})$. Their main distinction is that PCA assumes that the rows of X are iid samples from a p-dimensional multivariate distribution, the columns and the rows correspond to variables and cases respectively, whereas the SVD assumes each row has its own "fixed effect", such as space, time, genes, age groups, and so on, so is each column, and the roles of the rows and columns are switchable. See Chapter 1 (Yang et al., 2011) for more comparisons.

Hence, they have distinct models: SVD assumes low-rank means plus noise, whereas PCA analyzes the covariance between the column variables. We use the notations $p_u \times p_v$ to denote the dimensionality for SVD on purpose to suggest the symmetry of the SVD problem as opposed to the asymmetry of the PCA problem where the convention is to use $n \times p$. The reason is because the matrices M, U, D, V are all deterministic and the increase of the number of the rows will incur the increase of the number of the parameters, which we usually use p to represent. For the same reason, it is not hard to arrive at the conclusion that as p_u, p_v grow, the estimation problem becomes more difficult.

For clarity, it is necessary to specifically write down the model for PCA

$$X = UDV' + Z\,, \tag{2.4}$$

where everything is exactly the same as the SVD model except that here all the entries of U are random rather than fixed and they are iid $N(0,1)$, which makes the rows of X iid multivariate normal $N_p(0, \Sigma = VD^2V' + \sigma^2 I)$.

2.2.4 Rate of Convergence for Classical SVD

Under the setting of Section 2.2.1, the following theorem by Shabalin and Nobel (2010) states the phase transition phenomenon for classical SVD.

Theorem 1 (Shabalin and Nobel, 2010)**.** *Under model (2.1), fix the rank r of the mean matrix M, and let $p_u \to \infty$, $p_v \to \infty$, $p_u/p_v \to c \in (0, \infty)$. For the l'th singular value, define a (limiting, size-adjusted) signal-to-noise ratio by $\rho_l^2 = \lim_{p_u, p_v \to \infty} \frac{d_l^4/\sigma^4}{p_u p_v}$. Let $\hat{\mathbf{v}}_l^c$ be the l'th singular vector of X, where the superscript ^{c} stands for "classical", then we have*

$$\sin^2 \angle(\mathbf{v}_l, \hat{\mathbf{v}}_l^c) \overset{a.s.}{\to} \begin{cases} 1, & \text{if } \rho_l^2 \le 1, \quad (2.5a) \\ \dfrac{1 + \rho_l\sqrt{c}}{\rho_l^2 + \rho_l\sqrt{c}}, & \text{otherwise,} \quad (2.5b) \end{cases}$$

Replacing c by c^{-1} gives the result for $\hat{\mathbf{u}}_l$.

Remarks:

- The classical SVD will be consistent if and only if $\frac{\max(p_u, p_v)}{d_1^2/\sigma^2} \to 0$, which implies that the signal to noise ratio has to be extremely strong for the classic SVD estimates to be consistent. In particular, d_1^2/σ^2 at least has to go to infinity as the dimensions grow.

- PCA has similar phase transition phenomenon. See Johnstone and Lu (2009), Paul and Johnstone (2007), and Nadler (2009). However, the con-

dition for classical PCA to be consistent is quite different: $\frac{p}{n\frac{d_l^4/\sigma^4}{1+d_l^2/\sigma^2}} \to 0$. We once again can see the difference between SVD and PCA. The number of rows of the matrix in the SVD setting p_u and in the PCA setting n play opposite roles. Moreover, in the PCA setting the signal strength d_1^2/σ^2 can shrink as long as the number of observations n increases fast enough to counterbalance the effect of the increase of the number of covariates p for the classical PCA to be consistent.

2.2.5 Sparsity Assumptions for the Singular Vectors

In view of the inconsistency result of the classical SVD in the presence of noise when the signal is weak, we hope the data matrix has extra structure that we can take advantage of to improve our estimation. The extra structure that we are interested in is the sparsity of the singular vectors, which requires the singular vectors to be concentrated in a smaller subset of the coordinates and thus essentially reduces the effective number of parameters from the order of p_u, p_v to a smaller one.

Formally speaking, we adopt the concept of weak l_q ball from Gaussian sequence literature (Johnstone, 2011) to quantify the sparsity level. For any p-vector $\mathbf{u}$, we say that $\mathbf{u}$ belongs to the weak l_q ball of radius s, denoted by $\mathbf{u} \in wl_q(s)$, if

$$|\mathbf{u}|_{(i)} \leq si^{-1/q}, \tag{2.6}$$

where $|\mathbf{u}|_{(i)}$ is the i-th largest element, in the absolute sense. For $0 < q < 2$, condition (2.6) implies rapid decay of the ordered coefficients of $\mathbf{u}$, and hence describes its sparsity.

In the context of functional data analysis, we can think of the singular vectors as vectors of the discrete wavelet coefficients of some smooth functions spanned onto some sufficiently regular basis functions. For example, if a function belongs to the Besov space $B^{\alpha}_{p,p'}$, then its wavelet coefficients will belong to a weak l_q ball (Donoho, 1993; Johnstone, 2011). Therefore, sparsity has more generality than smoothness to a certain extent.

One may want to impose l_q constraint rather than wl_q constraint to define the sparsity. In fact, we have the following relation of inclusion between these two constraints $wl_q(s') \subset l_q(s) \subset wl_q(s)$ for $s' < s$, because of which we will focus on wl_q balls for minimax upper bound and restrict attention to l_q balls to derive the minimax lower bound later.

Finally, combining the low rank mean assumption and the weak l_q ball concept, the parameter spaces characterized by the quadruple (s_u, q_u, s_v, q_v) is defined to be

$$\begin{aligned}\Theta(s_u, q_u; s_v, q_v) = \{M = UDV' : \quad & U'U = I, V'V = I, \\ & D = \mathrm{diag}(d_1, \ldots, d_r) > 0, \\ & \mathbf{u}_l \in wl_{q_u}(s_u), \mathbf{v}_l \in wl_{q_v}(s_v)\}\,.\end{aligned} \tag{2.7}$$

Of course, the parameters (s_u, q_u, s_v, q_v) in (2.7) can be potentially different for each layer of the SVD $d_l \mathbf{u}_l \mathbf{v}_l$. In other words, they may have their own subscripts $_l$ to suggest different levels of sparsity. All the theorems and proofs in this chapter will carry through with the extra subscripts. For simplicity, we suppose they are the same.

From now on, we will further assume that the standard deviation of the normal

noise σ equals 1 because only the signal to noise ratio matters. In the case σ is known, we can scale the data by dividing the matrix by σ. If it is unknown, it can be estimated rather easily, say, by a robust estimate of the standard deviation of all the entries of the data matrix, treating the signal entries as outliers because of the sparsity assumption. This estimate can be further refined by a rough understanding of the locations of the signals in the matrix; see Section 2.4.3 for details.

2.3 Minimax Lower Bound

In this section, a lower bound on the minimax risk of estimating $span(U), span(V), M$ over the parameter space (2.7) is derived under the model (2.1) with the loss functions defined as in (2.2, 2.3).

Theorem 2. *There exists a constant c, such that for any possible estimator $\tilde{U}, \tilde{M}$ of U, M,*

$$\inf_{\tilde{U}} \sup_{M \in \Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_U(\tilde{U}, U) \geq c m_u \epsilon^2 \,, \tag{2.8a}$$

$$\inf_{\tilde{M}} \sup_{M \in \Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_M(\tilde{M}, M) \geq c(m_u \vee m_v) \epsilon^2 \,, \tag{2.8b}$$

where m_u and ϵ^2 are defined by

$$m_u = \min\{d_1^2, p_u, \bar{s}_u^{q_u} d_1^{q_u}\},\ \epsilon^2 = d_1^{-2} \,, \tag{2.9}$$

where $\bar{s}_u^{q_u} = s_u^{q_u} - 1$.

Moreover, if there are numeric constants $K_1, K_2 > 0, \alpha \in (0,1)$, such that

$$\frac{\bar{s}_u^{q_u} d_1^{q_u}}{(\log(p_u \vee p_v))^{q_u/2}} \leq \min\{K_1 \frac{d_1^2}{\log(p_u \vee p_v)}, K_2 p_u^{\alpha}\} \,. \tag{2.10}$$

Then

$$m_u = \frac{\bar{s}_u^{q_u} d_1^{q_u}}{(\log(p_u \vee p_v))^{q_u/2}},\ \epsilon^2 = \frac{\log(p_u \vee p_v)}{d_1^2}\,. \tag{2.11}$$

The minimax lower bound for estimating V can be obtained by replacing the subscript $_u$ by $_v$ accordingly.

Remarks:

- Although the two terms m_u, ϵ^2 vary on a case by case basis (2.9, 2.11), the lower bound (2.8) is always the product of these two quantities: intuitively, ϵ^2 can be thought of as the average error per coordinate estimated and m_u as the "effective" number of "significant" coordinates to be estimated.

  To make the terminology "effective" and "significant" more clear, one has to understand the worst case configuration first, which is the $M \in \Theta(s_u, q_u; s_v, q_v)$ that attains the supreme risk $\sup_{M\in\Theta(s_u,q_u;s_v,q_v)}$ and should be the most difficulty one to for any procedure to estimate. In the proof of Theorem 2 in Section 2.6.1, we will see that the worst case configuration is that the singular vectors have many coordinates of the size a constant times either $1/d_1$ or $\sqrt{\log p_u}/d_1$ depending on whether condition (2.10) is satisfied or not and the rest coordinates are exactly zero except for one coordinate that is close to 1. In this configuration, m_u is simply the number of non-zero coordinates, that is, although there are p_u coordinates in total, we only need to estimate m_u of them. That is why we call it the "effective" dimension.

  This understanding of the lower bound further suggests that for any procedure, if it is optimal, it should be able to extract coordinates of magnitude larger than the one stated above. In the proof of the mimimax upper bound

(Section 2.6.3) which bounds the risk of our procedure (will be given in the next section), we will show that such coordinates will be estimated and there are m_u of them, and the rest coordinates with smaller magnitude will be estimated by zero even if they are not exactly zero.

- The complicate part of the theorem is that m_u varies depending on the relative size of following quantities $d_1^2, p_u, \bar{s}_u^{q_u} d_1^{q_u}, \frac{\bar{s}_u^{q_u} d_1^{q_u}}{(\log(p_u \vee p_v))^{q_u/2}}$. We next explain why these quantities come into play one by one. To start, d_1^2 is the "signal-to-noise" ratio since we set $\sigma^2 = 1$ in Section 2.2.5, which makes $1/d_1^2$ a sensible quantity in ϵ^2 because it can be seen as the "noise-to-signal" ratio and fulfills the duty as σ^2 in a normal mean problem. We then move to p_u, which is the dimension of the data matrix and hence the largest possible number of non-zero coordinates. The third one $\bar{s}_u^{q_u} d_1^{q_u}$ comes from the $wl_q(s)$ sparsity constraint since it involves s_u, q_u and it is the maximum possible number of nonzero coordinates of size $1/d_1$ that satisfies the wl_q ball condition. The last one is similar as the third one except that it captures the number of nonzero coordinates of size $\sqrt{\log p_u}/d_1$ instead. In all, m_u is always upper bounded by the quantities discussed above.

- Understanding these quantities can facilitate our understanding of the whole theorem. Situation (2.9) actually embodies three lower bounds, together with the lower bound in (2.11), we have four cases. We will explain them in detail now.

    1. Low signal case: $m_u = d_1^2 < \min\{p_u, \bar{s}_u^{q_u} d_1^{q_u}\}$. If d_1^2 is so small that it is less than $\min\{p_u, \bar{s}_u^{q_u} d_1^{q_u}\}$, then the lower bound $m_u \epsilon^2$ is a constant, which implies that no algorithm can ever achieve consistency because signal is not strong and the sparsity is not prominent enough at the

same time. Note that it has certain overlap with Theorem 1 for that when $d_1^2 < p_u$, Theorem 1 shows almost sure inconsistency (2.5a), but Theorem 2 shows the inconsistency over the whole parameter space. On the other hand, Theorem 1 is limited to the situation $p_u/p_v \to c$, whereas Theorem 2 does not confine itself to this situation.

2. Dense case: $m_u = p_u < \min\{d_1^2, \bar{s}_u^{q_u} d_1^{q_u}\}$. In this case, the signal is strong enough for consistency since $d_1^2 > p_u$, but the sparsity constraint does not take effect because $\bar{s}_u^{q_u} d_1^{q_u} > p_u$, which makes every coordinate of the singular vector non-zero and to be estimated. The lower bound is p_u/d_1^2, which is the same as the convergence rate of the classical SVD in (2.5b), which suggests the minimaxity of the classical SVD in the dense case when the signal is strong enough.

3. Sparse case: $m_u = \bar{s}_u^{q_u} d_1^{q_u} < \min\{d_1^2, p_u\}$. Here, $\bar{s}_u^{q_u} d_1^{q_u} < p_u$, and the sparsity constraint is active. As long as the signal is sufficiently strong $d_1^2 > \bar{s}_u^{q_u} d_1^{q_u}$, no matter whether $d_1^2 > p_u$ or not, it is possible for recovery. If $d_1^2 > p_u$, classical SVD can get consistent estimate, but not optimal. If $d_1^2 < p_u$, classical SVD is no longer consistent, but procedures taking advantage of the sparsity structure can achieve consistency.

4. Super sparse case (2.11): $m_u = O(p_u^\alpha), 0 < \alpha < 1$, which suggests that the fraction of non-zero coordinates goes to zero as the dimension increases, whereas for the other three cases, the fraction can be non-vanishing. This super sparsity makes estimation harder by a factor of $\sqrt{\log p_u}$ because of the uncertainty of the location of the nonzero coordinates. Under this circumstance, the optimal procedure can only detect signal larger than $\sqrt{\log(p_u \vee p_v)}/d_1$ rather than $1/d_1$ for the other three cases, which makes the error per coordinate ϵ^2 equal $\log(p_u \vee p_v)/d_1^2$.

Other aspects of the super sparse case remain the same as of the sparse case. Looking ahead, Theorem 3 about the minimax upper bound of our procedure in Section 2.5.1 precisely achieves this lower bound and proves the minimax optimality of our procedure.

- A parallel minimax lower bound for the estimation problem for the PCA model (2.4) is given by Paul and Johnstone (2007) Theorem 2(b), which also contains four identical cases as ours, but the proofs for the first three cases and the super sparse case are intertwined and more involved. Our proofs separates the worst case configuration for (2.9) and (2.11), are more straightforward, have more analogy with the traditional non-parametric function estimation settings and Gaussian sequence model and can be easily adjusted to prove their results.

2.4 Estimation Scheme

In Section 2.3, we develop the minimax lower bound, which gives the benchmark for the sparse SVD problem. In this section, we will systematically describes the estimation strategy and in the next section, we shall derive the upper bound for our estimation method and establish its optimality by comparing it to the benchmark.

Recalling that our goal is to estimate the subspaces spanned by the leading singular vectors rather than each singular vector 2.2.2, our estimation scheme originates from an iterative algorithm that can be used to calculate the classical singular vectors simultaneously; see Section 2.4.1. To impose sparsity, in Section 2.4.2, we modify the iterative algorithm in each iteration by thresholding small entries to zero, which we call iterative thresholding algorithm (IT) for sparse

SVD and is our final proposal to the sparse SVD problem. To start the iterative algorithm, an initialization algorithm is provided in Section 2.4.3.

Note that Chapter 1, which is purely methodological, proposes an algorithm called FIT-SSVD that modifies the IT algorithm in the current chapter to cope with non-normal noise. Although simulation and real data studies demonstrate that FIT-SSVD performs well empirically, it does not have theoretical guarantee for non-normal noise. For completeness and clarity of the current chapter, in spite of a certain degree of repetitiveness, we will fully describe the IT algorithm, which although does not adapt to non-normal noise excellently, but is theoretically minimax optimal under normal noise assumption.

It is also worth noting that the IT algorithm for sparse SVD is a two way generalization of the IT for sparse PCA problem by Ma (2011). Simply put, the asymmetry is handled by performing the IT algorithm for sparse PCA problem twice with the observed data matrix and its transpose, which can be seen more clearly in Section 2.4.1 and 2.4.2.

2.4.1 Two-way Orthogonal Iteration Algorithm

This subsection is devoted to explaining an iterative algorithm to perform the task of classical SVD. To that end, we begin with the so-called orthogonal iteration algorithm (See Golub and Van Loan (1996), Chapter 8), that is a generalization of the power method for calculating multiple dimensional invariant subspaces of symmetric matrices by replacing normalization step in the power method by orthonormalization. Going one step further, for an arbitrary asymmetric or rectangular matrix $M_{p_u \times p_v}$ with SVD UDV', in order to compute the subspaces spanned by

the leading r left and right singular vectors, one can generalize the orthogonal iteration algorithm even more by alternating the orthogonal iteration algorithm with M and its transpose M' until convergence. To be more explicit, starting with an orthonormal matrix $V^{(0)}_{p_v \times r}$, repeating the the following four steps until convergence will produce sequences of orthonormal matrices $U^{(k)}, V^{(k)}$ that become closer and closer to U and V respectively:

(1) Right-to-Left Multiplication:	$U^{(k),mul} = MV^{(k-1)}$
(2) Left Orthonormalization with QR Decomposition:	$U^{(k)}R^{(k)}_u = U^{(k),mul}$
(3) Left-to-Right Multiplication:	$V^{(k),mul} = M'U^{(k)}$
(4) Right Orthonormalization with QR Decomposition:	$V^{(k)}R^{(k)}_v = V^{(k),mul}$

The superscript $^{(k)}$ indicates the k'th iteration, and ^{mul} the generally non-orthonormal intermediate output of multiplication step. For $r = 1$, the QR decomposition step for orthonormalization reduces to normalization step in the power method. If the matrix M is symmetric, the first two and last two steps are the same, making the original orthogonal iteration algorithm a special case of the two way orthogonal iteration algorithm.

In the noiseless case, it is easy to verify that the above procedure will converge to the classical SVD of the matrix M by mimicking the proof in Chapter 8 of Golub and Van Loan (1996) given that the starting point $V^{(0)}$ is regular enough, which will be clarified in Section 2.4.3. However, both Theorem 1 and 2 show that in the low signal case, the above procedure applied directly to the observed data matrix $X = M + Z$, equivalent to the classical SVD, fails to estimate the underlying truth consistently because of the overwhelming noise. The problem with the procedure is that under the sparsity assumption, only a small subset of the large noisy matrix contains most of the structure, but the classical SVD

estimates all the coordinates including those from the structureless cells, resulting in unnecessary and huge accumulation of noise. Along with the drawback of the poor statistical property, the above procedure is computationally inefficient because it involves all the seeming noise entries. One solution is to trim those cells that might come from noise, which has the potential of reducing the variance as well as increasing the computational speed. This heuristic motivates our proposal for the sparse SVD problem in the next subsection.

2.4.2 IT Algorithm for Sparse SVDs

The key innovation in the IT algorithm is the addition of the thresholding step, which is unsurprisingly incorporated to kill off the coordinates that are likely noise. The thresholding steps are inserted between the multiplication and the orthonormalization steps, which makes a majority of the entries zero and dramatically reduces the computational time for subsequent orthonormalization and multiplication steps. Although the thresholding steps will reduce the variance at the price of some bias, so long as the sparsity assumption is sensible, the reduced variance will hopefully dominate the inflated bias. The algorithm is schematically laid out in Algorithm 4

In Algorithm 4, the left and right thresholding steps threshold the intermediate output from the previous multiplication steps elementwisely and generate the intermediate result $U^{(k),thr}, V^{(k),thr}$ whose entries may contain many zeros. 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 $\eta_{soft}(x,\gamma) = \text{sign}(x)(|x|-\gamma)_+$, hard-thresholding with $\eta_{hard}(x,\gamma) = x1_{|x|>\gamma}$, as well as the thresholding function used in SCAD (Fan and Li, 2001). The parameter γ in the thresholding function $\eta(x,\gamma)$

Input:
1. Observed data matrix X.
2. Target rank r.
3. Thresholding function η and constants γ_u, γ_v.
4. Initial orthonormal matrix $V^{(0)} \in \mathbb{R}^{p\times r}$.

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

repeat

Right-to-Left Multiplication: $U^{(k),mul} = XV^{(k-1)}$.

Left Thresholding: $U^{(k),thr} = (u_{il}^{(k),thr})$, with $u_{il}^{(k),thr} = \eta\left(u_{il}^{(k),mul}, \gamma_u\sqrt{\log(p_u \vee p_v)}\right)$.

Left Orthonormalization with QR Decomposition: $U^{(k)}R_u^{(k)} = U^{(k),thr}$.

Left-to-Right Multiplication: $V^{(k),mul} = X'U^{(k)}$.

Right Thresholding: $V^{(k),thr} = (v_{jl}^{(k),thr})$, with $v_{jl}^{(k),thr} = \eta\left(v_{jl}^{(k),mul}, \gamma_v\sqrt{\log(p_u \vee p_v)}\right)$.

Right Orthonormalization with QR Decomposition: $V^{(k)}R_v^{(k)} = V^{(k),thr}$.

until *Convergence*;

Algorithm 4: IT Algorithm for Sparse SVDs

is called the threshold level and is set to be $\sqrt{\log(p_u \vee p_v)}$ times large enough constants γ_u, γ_v whose values will be specified later. The threshold level also remains the same across all the iterations and the columns. The order of the threshold level $\sqrt{\log(p_u \vee p_v)}$ is crucial in the minimax analysis for the upper bound of the risk of IT. In practice, we stop the iterations once the subsequent updates of the orthonormal matrices are close to each other, say, in the sense that their distance defined similarly as in (2.3) is below some tolerance level.

Another proposition for the place to insert the thresholding step is after the QR decomposition step, which is turned down for several reasons. We want to maintain orthonormality for the upcoming multiplication step. Further, orthonormal estimation for U, V will make the estimation of D, M easier. Lastly, the computational cost for the QR decomposition step is reduced with the thresholding step coming first.

If the input matrix X is symmetric, the last three steps in Algorithm 4 will be the same as the first three, the output of the algorithm will be an estimate for the leading eigenspace. Furthermore, the three steps resemble the IT algorithm for sparse PCA problem (Ma, 2011).

2.4.3 Initialization Algorithm for Sparse SVDs

Algorithm 4 requires a proper starting frame $V^{(0)}$ whose span has no dimension that is orthogonal to the subspace spanned by the true V, otherwise the algorithm will not get close to the truth no matter after how many iterations. Mathematically, we demand $V'V^{(0)}$ has no zero singular values. People often use the classical SVD as a candidate for $V^{(0)}$, which is however inferior because it is not only computationally expensive for large data, but also asymptotically orthogonal to the true V (Theorem 1) and needs many iterations to accumulate sufficient power to converge. We therefore propose an initialization algorithm to address these issues.

Algorithm 5 is motivated by Johnstone and Lu (2009) who obtained a consistent estimate for the sparse PCA problem by initially reducing the dimensionality which is achieved by focusing on a submatrix of the sample covariance matrix. We adapt their idea to the two-way case: we first select a subset of rows and columns (Step 1), perform the classical SVD on the reduced submatrix afterwards (Step 2), and expand the left and right singular vectors of size of the reduced matrix to their original size by padding zeros to the coordinates that are not selected in the first step (Step 3). The second and third steps are trivial. We will mainly give some intuition on how we select the rows, the selection of the columns being analogous.

Input:
1. Observed data matrix X.
2. Target rank r.
3. User-specified large enough constants α_u, α_v.

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

1 Subset selection:
Let I and J be the subsets of indices,

$$\begin{aligned} I &= \{i : \tfrac{\sum_{j=1}^{p_v} x_{ij}^2}{p_v} \geq 1 + \alpha_u \sqrt{\tfrac{\log(p_u \vee p_v)}{p_v}}\}\,, \\ J &= \{j : \tfrac{\sum_{i=1}^{p_u} x_{ij}^2}{p_u} \geq 1 + \alpha_v \sqrt{\tfrac{\log(p_u \vee p_v)}{p_u}}\}\,. \end{aligned} \tag{2.12}$$

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 $\mathbf{u}_1^I, \ldots, \mathbf{u}_r^I$ ($|I| \times 1$ each) and $\mathbf{v}_1^J, \ldots, \mathbf{v}_r^J$ ($|J| \times 1$ each).

3 Zero-padding: Create $U^{(0)} = [\mathbf{u}_1^{(0)}, \ldots, \mathbf{u}_r^{(0)}]$ ($p_u \times r$) and
$V^{(0)} = [\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_r^{(0)}]$ ($p_v \times r$),
such that $\mathbf{u}_{Il}^{(0)} = \mathbf{u}_{Il}^I\,,\ \mathbf{u}_{I^c l}^{(0)} = 0\,,\ \mathbf{v}_{Jl}^{(0)} = \mathbf{v}_{Jl}^J\,,\ \mathbf{v}_{J^c l}^{(0)} = 0.$

Algorithm 5: Initialization algorithm for sparse SVDs

To understand Step 1, consider the simplest setting when $r = 1$, in which case $x_{ij} = d_1 u_{i1} v_{j1} + z_{ij}$. The goal is to distinguish the significant rows from the rest. We intend to keep the i-th row if $|u_{i1}|$ is large enough. All of the information about u_{i1} is contained in the i-th row, which is a multivariate normal random variable with mean $d_1 \mathbf{v}_1 u_{i1}$ and identity covariance matrix. We want to eliminate the impact of $\mathbf{v}_1$ on the selection of u_{i1}. Note that $\mathbf{v}_1$ has unit norm, which makes the squared l_2 norm of the i-th row follow a non-central chi-square distribution with degree of freedom p_v and non-centrality parameter $\sum_{j=1}^{p_v} d_1^2 u_{i1}^2 v_{j1}^2 = d_1^2 u_{i1}^2$ that does not depend on $\mathbf{v}_1$ any more, which is denoted by $\chi_{p_v}^2(d_1^2 u_{i1}^2)$ thereafter. By law of large number, $\chi_{p_v}^2(0)/p_v \to 1$. If u_{i1} is not near 0, then $\chi_{p_v}^2(d_1^2 u_{i1}^2)/p_v$ will be away from 1, biased upwards (2.12). Hence, we can tell the magnitude of u_{i1} by differentiating between the central and non-central chi-square distribution. As for the appearance of the term $\sqrt{\frac{\log(p_u \vee p_v)}{p_v}}$ in (2.12), it is inherently determined

by the tail behavior of the chi-square distribution and the best one can hope for.

2.5 Minimax Upper Bound

We have so far established the minimax lower bound as the benchmark in Section 2.3 and described our estimation procedure in Section 2.4, we now turn to the asymptotic property of the IT algorithm in this section. Moreover, the initialization algorithm 5 itself can be a crude estimator, in what follows, we will also give an upper bound for its risk and conclude that although it is consistent but not optimal.

2.5.1 Upper Bound for the IT Algorithm

We first state the theorem for the upper bound result for the IT algorithm under model(2.1) and parameter space (2.7) with respect to the loss functions (2.2, 2.3) defined earlier.

Theorem 3. *Let $\hat{U}, \hat{V}$ be the output of Algorithm 4 with initialization Algorithm 5. Define $\hat{M} = \hat{U}\hat{D}\hat{V}'$, where $\hat{D} = diag(\hat{d}_1, \ldots, \hat{d}_r)$ with $\hat{d}_l = \hat{\mathbf{u}}_l' X \hat{\mathbf{v}}_l$. If $s_u^{q_u}\left(\frac{\sqrt{p_u \log(p_u \vee p_v)}}{d_1^2}\right)^{1-q_u/2} = o(1)$ and $s_v^{q_v}\left(\frac{\sqrt{p_v \log(p_u \vee p_v)}}{d_1^2}\right)^{1-q_v/2} = o(1)$, there exists constant C, s.t.,*

$$\sup_{M \in \Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_U(\hat{U}, U) \leq C(m_u \vee m_v)\epsilon^2\,, \tag{2.13a}$$

$$\sup_{M \in \Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_M(\hat{M}, M) \leq C(m_u \vee m_v)\epsilon^2\,, \tag{2.13b}$$

where $\epsilon^2 \sim \frac{\log(p_u \vee p_v)}{d_1^2}$, $m_u \sim s_u^{q_u} d_1^{q_u}\left(\frac{1}{\log(p_u \vee p_v)}\right)^{q_u/2}$ and m_v accordingly.

Remarks:

- Recall that the two critical quantities ϵ^2, m_u that determine the minimax lower bound in the super sparse case are of exactly the same form as those in Theorem 3. Consequently, the minimax risk of the sparse SVD problem satisfies that if $m_u = O(p_u^\alpha), 0 < \alpha < 1$, then

$$\begin{aligned} \inf_{\tilde{U}} \sup_{M \in \Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_U(\tilde{U}, U) &\asymp m_u \epsilon^2 \,, \\ \inf_{\tilde{M}} \sup_{M \in \Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_M(\tilde{M}, M) &\asymp (m_u \vee m_v)\epsilon^2 \,. \end{aligned}$$

  and the IT estimators are minimax rate optimal in the super sparse case.

- Let us compare the upper bound with the lower bound results for the other three cases mentioned in Section 2.3. For the low signal case, no method can achieve consistency, neither can IT, because the right hand side of (2.13) is not $o(1)$. For the sparse case, the rate of convergence of IT is slower than the lower bound by a factor of logarithm $(\log(p_u \vee p_v))^{1-q_u/2}$, which makes our method near optimal. For the dense case, when the classical SVD is minimax, it is illuminating to compare the asymptotic supremum risk of IT with that of the classical SVD. When the dimensions are sufficiently large, such that $p_u \geq s_u^{q_u} \left(\frac{\log(p_u \vee p_v)}{d_1^2}\right)^{-q_u/2}$, one can replace the upper bound in (2.13) by $\frac{(p_u \vee p_v)\log(p_u \vee p_v)}{d^2}$ which is slower than the rate of classical SVD by a factor of $\log(p_u \vee p_v)$.

- Further note that our estimators do not require knowledge of the parameters $(s_u, q_u; s_v, q_v)$ and hence are adaptive.

- As we briefly mentioned in the remarks after Theorem 2, comparing the lower and upper bound results side by side reveals and the proof of Theorem

3 confirms that all of the coordinates of the singular vectors of size of order $\sqrt{\log(p_u \vee p_v)}/d_1$ will be included in the final estimators and the smaller coordinates will be zeroed out. This is the core of the proof.

2.5.2 Upper Bound for the Initialization Algorithm

We will then give the upper bound for the supremum risk of the initialization Algorithm 5 under the same model assumptions, parameter spaces, and loss functions.

Theorem 4. *Let $\hat{U}^{(0)}, \hat{V}^{(0)}$ be the output of Algorithm 5. If $s_u^{q_u}\left(\frac{\sqrt{p_u \log(p_u \vee p_v)}}{d_1^2}\right)^{1-q_u/2} = o(1)$ and $s_v^{q_v}\left(\frac{\sqrt{p_v \log(p_u \vee p_v)}}{d_1^2}\right)^{1-q_v/2} = o(1)$, then there exists constant C, s.t.,*

$$\begin{aligned} &\sup_{M \in \Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_U(\hat{U}^{(0)}, U) \\ &\leq C\left(s_u^{q_u}\left(\frac{\sqrt{p_u \log(p_u \vee p_v)}}{d_1^2}\right)^{1-q_u/2} + s_v^{q_v}\left(\frac{\sqrt{p_v \log(p_u \vee p_v)}}{d_1^2}\right)^{1-q_v/2}\right). \end{aligned} \tag{2.14}$$

Similar result holds for V.

Remarks:

- Theorem 4 shows that we need a looser condition for the initialization algorithm to be consistent, namely, $s_u^{q_u}\left(\frac{\sqrt{p_u \log(p_u \vee p_v)}}{d_1^2}\right)^{q_u/2} \to 0$ as opposed to $\frac{p_u \vee p_v}{d_1^2} \to 0$, which is required for the classical SVD to be consistent.
- It is obvious that the upper bound in (2.14) is much slower than the one for IT in (2.13) and does not achieve the minimax lower bound in (2.8), that makes the initialization algorithm not optimal.

- The upper bounds for the classical SVD, the initialization algorithm and IT involve $\frac{p_u \vee p_v}{d_1^2}$, $\frac{\sqrt{p_u \log(p_u \vee p_v)}}{d_1^2}$, $\frac{\log(p_u \vee p_v)}{d_1^2}$ respectively. It is interesting to note that "ladder" relationship between them.

- The convergence rate $\frac{\sqrt{p_u \log(p_u \vee p_v)}}{d_1^2}$ solely depends on the rate of threshold level $\sqrt{\frac{\log(p_u \vee p_v)}{p_v}}$ in Step 1 of Algorithm 5. The proof of the theorem in Section 2.6.2 will further confirms that only the coordinates whose order are larger than $\frac{(p_u \log(p_u \vee p_v))^{1/4}}{d_1}$ can be identified by the subset selection procedure. Those signals that are not extremely high but still above $\frac{\sqrt{\log(p_u \vee p_v)}}{d_1}$ will be immersed in the noise, resulting in huge bias in the estimator. One may wonder if this is the case, why cannot we lower the threshold level in Step 1 to achieve a better result? The quick answer is the original threshold level is the lowest such that we can tell the central and non-central chi-square distribution apart by concentration inequality, which means there is no space for improvement for an algorithm this simple.

2.6 Proofs

In what follows, we prove the main results Theorems 2, 4, 3 in Sections 2.6.1, 2.6.2, 2.6.3 respectively. A few technical proofs of the lemmas used in this section are deferred to the Appendix. Throughout this section, we denote by C, c generic constants that may vary from place to place.

2.6.1 Proof of Theorem 2

Observing the inclusion relationship between l_q and weak l_q balls $l_q(s) \subset w l_q(s)$, let us define a new parameter space that is a subset of the original one $\Theta'(s_u, q_u; s_v, q_v) \subset \Theta(s_u, q_u; s_v, q_v)$ by

$$\begin{aligned} \Theta'(s_u, q_u; s_v, q_v) = \{M = UDV' : \quad & U'U = I, V'V = I, \\ & D = \mathrm{diag}(d_1, \ldots, d_r) > 0, \\ & \mathbf{u}_l \in l_{q_u}(s_u), \mathbf{v}_l \in l_{q_v}(s_v)\} \,. \end{aligned} \tag{2.15}$$

Together with the fact that $L_U(\hat{U}, U)$ is trivially lower bounded by $L_U(\hat{\mathbf{u}}_1, \mathbf{u}_1)$, we will prove (2.8) by showing that

$$\inf_{\tilde{\mathbf{u}}_1} \sup_{M \in \Theta'(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_U(\tilde{\mathbf{u}}_1, \mathbf{u}_1) \geq c m_u \epsilon^2 \,, \tag{2.16a}$$

$$\inf_{\tilde{M}} \sup_{M \in \Theta'(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_M(\tilde{M}, M) \geq c(m_u \vee m_v)\epsilon^2 \,. \tag{2.16b}$$

The proofs of the two parts (2.9) and (2.11) in Theorem 2 use two different well-known techniques: Assouad's Lemma and Fano's Lemma respectively. In order to use both of the general machinery, it is necessary to obtain a finite number of parameters that belong to the parameter space $\Theta'(s_u, q_u; s_v, q_v)$ (2.15) and meanwhile serve as the worst case configuration. The minimax lower bound for estimating the subspace spanned by the singular vectors U (2.16a) and the whole matrix (2.16b) are proved by the same worst case configuration and will be proved together later.

Proof of (2.9) in Theorem 2. We shall use Assouad's lemma (see Lemma 3 in the Appendix) to prove (2.9). To this end, we shall create the worst case parameters in the following way. Let $m_u = p_u - r$ and $\rho_u \in (0,1)$, the exact values of which are to be specified later in Table 2.1 for different cases discussed in the remarks after Theorem 2. In addition, let $\mathbf{e}_l$ denote the p_u- or p_v-vector depending on the context where the l-th coordinate is 1 and the rest are all zeros.

First, we construct a finite collection of models as the following: for any $\gamma = (\gamma_1, \ldots, \gamma_{m_u}) \in \Gamma = \{0,1\}^{m_u}$, let

$$M(\gamma) = d_1 \mathbf{u}_1(\gamma)\mathbf{v}_1' + \sum_{l=2}^{r} d_l \mathbf{u}_l \mathbf{v}_l' \in \mathbb{R}^{p_u \times p_v} ,$$

where $\mathbf{u}_l = \mathbf{e}_l$ for $l = 2, \ldots, r$, $\mathbf{v}_l = \mathbf{e}_l$ for $l = 1, \ldots, r$, and

$$\mathbf{u}_1(\gamma) = \sqrt{1-\rho_u^2}\mathbf{e}_1 + \rho_u \frac{1}{\sqrt{m_u}} \sum_{h=1}^{m_u} (2\gamma_h - 1)\mathbf{e}_{r+h} .$$

Clearly, $\{\mathbf{u}_l, l = 1, \ldots, r\}$ and $\{\mathbf{v}_l, l = 1, \ldots, r\}$ are two sets of orthonormal vectors for any γ. For each fixed γ,

$$\mathbf{u}_1(\gamma) = (\sqrt{1-\rho_u^2}, \underbrace{0, \ldots, 0}_{r-1}, \underbrace{\pm\frac{\rho_u}{\sqrt{m_u}}, \ldots, \pm\frac{\rho_u}{\sqrt{m_u}}}_{m_u}, \underbrace{0, \ldots, 0}_{p_u - r - m_u})'$$

could be viewed as a perturbation of $\mathbf{e}_1 = (1, 0, \ldots, 0)'$ and for $h = 1, \ldots, m_u$, $\mathbf{u}_{1,r+h}$ is always of size $\frac{\rho_u}{\sqrt{m_u}}$ and is positive if $\gamma_h = 1$ and is negative if $\gamma_h = 0$.

In order to apply Assouad's Lemma, we will consider two metrics: for $\mathbb{R}^{p_u \times p_v}$, the metric d_M between two matrices is given by the Frobenius norm

$$d_M(\tilde{M}, M) = \|\tilde{M}, M\|_F ; \tag{2.17}$$

for $\mathbb{R}^{p_u}$, the metric d_U between two vectors is defined by

$$d_U(\tilde{\mathbf{u}}_1, \mathbf{u}_1) = \sin \angle(\tilde{\mathbf{u}}_1, \mathbf{u}_1)\,. \tag{2.18}$$

Note that there are two free parameters in the above construction: (i) ρ_u, the magnitude of the perturbation, and (ii) m_u, the number of coordinates that are perturbed. To embed this finite collection as a subset of our uniformity class $\Theta'(s_u, q_u; s_v, q_v)$, we impose on ρ_u and m_u the condition

$$(1-\rho_u^2)^{q_u/2} + \rho_u^{q_u} m_u^{1-q_u/2} \le s_u^{q_u}\,, \tag{2.19}$$

and so for any γ, $\mathbf{u}_1(\gamma) \in l_{q_u}(s_u)$, and $\{M(\gamma), \gamma \in \Gamma\} \subset \Theta'(s_u, q_u; s_v, q_v)$.

Next, we compute the quantities that appear in the lower bound in Assouad's Lemma. Define $H(\gamma, \gamma')$ to be the Hamming distance, which counts the number of positions at which γ and γ' differ. Because for any $\gamma \neq \gamma'$ with $H(\gamma, \gamma') = k$, there will be k different entries between $\mathbf{u}_1(\gamma)$ and $\mathbf{u}_1(\gamma')$, we have

$$\begin{aligned}
d_M^2(M(\gamma), M(\gamma')) &= \|M(\gamma) - M(\gamma')\|_F^2 \\
&= d_1^2 \|\mathbf{u}_1(\gamma) - \mathbf{u}_1(\gamma)\|^2 \\
&= d_1^2 k \left(\frac{2\rho_u}{\sqrt{m_u}}\right)^2 \\
&= k\frac{4d_1^2\rho_u^2}{m_u}\,, \\
d_U^2(\mathbf{u}_1(\gamma), \mathbf{u}_1(\gamma')) &= \sin^2 \angle(\mathbf{u}_1(\gamma), \mathbf{u}_1(\gamma')) \\
&= 1 - \cos^2 \angle(\mathbf{u}_1(\gamma), \mathbf{u}_1(\gamma')) \\
&= k\frac{2\rho_u^2}{m_u}\,.
\end{aligned}$$

Therefore, we obtain that

$$\begin{aligned}\min_{H(\gamma,\gamma')\geq 1} \frac{d_M^2(M(\gamma),M(\gamma'))}{H(\gamma,\gamma')} &= \frac{4d_1^2\rho_u^2}{m_u}\,,\\ \min_{H(\gamma,\gamma')\geq 1} \frac{d_U^2(\mathbf{u}_1(\gamma),\mathbf{u}_1(\gamma'))}{H(\gamma,\gamma')} &= \frac{2\rho_u^2}{m_u}\,.\end{aligned}$$

In addition, for each γ, we have a probability measure P_γ in terms of the matrix normal distribution $N_{p_u\times p_v}(M(\gamma), I_{p_u}\otimes I_{p_v})$. It is straightforward to verify that for any $\gamma\neq\gamma'$ with $H(\gamma,\gamma')=1$, the Kullback-Leibler divergence between P_γ and $P_{\gamma'}$ is $KL(P_\gamma,P_{\gamma'})=2d_1^2\rho_u^2/m_u$. By the inequality $\|P\wedge Q\|\geq\frac{1}{2}\exp^{-KL(P,Q)}$ (Tsybakov 2009, Lemma 2.6), this leads to

$$\min_{H(\gamma,\gamma')=1}\|P_\gamma\wedge P_{\gamma'}\|\geq\frac{1}{2}\exp\left(-\frac{2d_1^2\rho_u^2}{m_u}\right).$$

With the last two displays, Assouad's Lemma implies that

$$\begin{aligned}\inf_{\tilde{M}}\sup_{\gamma\in\Gamma}\mathbb{E}2^2\|\tilde{M}-M(\gamma)\|_F^2 &\geq \frac{4d_1^2\rho_u^2}{m_u}\frac{m_u}{2}\frac{1}{2}\exp\left(-\frac{2d_1^2\rho_u^2}{m_u}\right),\\ \inf_{\tilde{\mathbf{u}}_1}\sup_{\gamma\in\Gamma}\mathbb{E}2^2\sin^2\angle(\tilde{\mathbf{u}}_1,\mathbf{u}_1) &\geq \frac{2\rho_u^2}{m_u}\frac{m_u}{2}\frac{1}{2}\exp\left(-\frac{2d_1^2\rho_u^2}{m_u}\right).\end{aligned}$$

Observe that $\|M(\gamma)\|_F^2=\sum_{l=1}^r d_l^2$ for any $\gamma\in\Gamma$, together with the assumption that the singular values are of the same order $1\geq\frac{d_r}{d_1}\geq c>0$, the last display then leads to

$$\begin{aligned}\inf_{\tilde{M}}\sup_{\gamma\in\Gamma}\mathbb{E}_M L_M(\tilde{M},M)&\geq c\rho_u^2\exp\left(-\frac{2d_1^2\rho_u^2}{m_u}\right),\\ \inf_{\tilde{\mathbf{u}}_1}\sup_{\gamma\in\Gamma}\mathbb{E}_M L_U(\tilde{\mathbf{u}}_1,\mathbf{u}_1)&\geq c\rho_u^2\exp\left(-\frac{2d_1^2\rho_u^2}{m_u}\right),\end{aligned}$$

To further investigate the right side, define $\bar{s}_u$ such that $\bar{s}_u^{q_u}=s_u^{q_u}-1$. Note that $\bar{s}_u\asymp s_u$. We specify the values of m_u and ρ_u^2 in three different cases as follows.

Case	Condition	m_u	ρ_u^2
Low signal	$d_1^2 \leq \min\{p_u - r, \bar{s}_u^{q_u} d_1^{q_u}\}$	d_1^2	$\frac{m_u}{d_1^2} = 1$
Dense	$p_u - r \leq \min\{d_1^2, \bar{s}_u^{q_u} d_1^{q_u}\}$	$p_u - r$	$\frac{m_u}{d_1^2}$
Sparse	$\bar{s}_u^{q_u} d_1^{q_u} \leq \min\{d_1^2, p_u - r\}$	$\bar{s}_u^{q_u} d_1^{q_u}$	$\frac{m_u}{d_1^2}$

Table 2.1: Three different cases for minimax lower bound.

Observe that in all of the three cases, condition (2.19) is always satisfied. Combining these three cases, we obtain that

$$\inf_{\tilde{M}} \sup_{M \in \Theta'(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_M(\tilde{M}, M) \geq \inf_{\tilde{M}} \sup_{\gamma \in \Gamma} \mathbb{E}_M L_M(\tilde{M}, M(\gamma)) \geq c\rho_u^2 = cm_u\epsilon^2\,,$$
$$\inf_{\tilde{\mathbf{u}}_1} \sup_{M \in \Theta'(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_U(\tilde{\mathbf{u}}_1, \mathbf{u}_1) \geq \inf_{\tilde{\mathbf{u}}_1} \sup_{\gamma \in \Gamma} \mathbb{E}_M L_U(\tilde{\mathbf{u}}_1, \mathbf{u}_1(\gamma)) \geq c\rho_u^2 = cm_u\epsilon^2\,,$$

where $\epsilon^2 = 1/d_1^2$. By symmetry, the same inequality holds if the right side is replace by $m_v\epsilon^2$ for the second inequality. This completes the proof.

Proof of (2.11) in Theorem 2. The technical tool for proving the second part of Theorem 2 is Lemma 4 which is equivalent to Fano's Lemma.

Throughout the proof, let m_u be the largest even number which is no greater than $\bar{s}_u^{q_u}(d_1^2/\log p_u)^{q_u/2}$. Without loss of generality, let us assume that $m_u \geq 2$. Clearly, $m_u \asymp s_u^{q_u}(d_1^2/\log p_u)^{q_u/2}$.

To construct the metric space in Lemma 4, let $\mathcal{P} = \{\mathbf{p}^{(j)} : j = 1, \ldots, n\}$ be a maximal subset of binary sequences of length p_u which satisfies the following three constraints:

1. For all j, $\mathbf{p}_i^{(j)} = 0, i = 1, \ldots, r$;
2. For all j, $|\{i : \mathbf{p}_i^{(j)} = 1\}| = m_u$;
3. For all $j \neq k$, $|\{i : \mathbf{p}_i^{(j)} = \mathbf{p}_i^{(k)} = 1\}| \leq m_u/2$.

In other words, each $\mathbf{p}^{(j)} \in \mathcal{P}$ has exactly m_u entries equal to 1, all of which are scattered at the $(r+1)$-th to the p_u-th coordinates, and all the rest are zeros. Any two different elements in $\mathcal{P}$ have less than half of their 1's overlapped.

Note that the condition 2.10 implies that $m_u \asymp p_u^\alpha = o(p_u - r)$. Applying Lemma 5 with $p = p_u - r$ and $m = m_u$ leads to

$$\log n \gtrsim m_u \log p_u \,. \tag{2.20}$$

We are now in the position to construct the metric space in Lemma 4. Indeed, let

$$\Theta_{sub} = \{M_j, j = 0, 1, \ldots, n\}$$

be a collection of $p_u \times p_v$ matrices with rank r. Here $M_0 = \sum_{l=1}^r d_l \mathbf{u}_l \mathbf{v}_l'$ with $\mathbf{u}_l = \mathbf{e}_l$ and $\mathbf{v}_l = \mathbf{e}_l$, for $l = 1, \ldots, r$. Moreover, for $j = 1, \ldots, n$, let

$$M_j = d_1 \mathbf{u}_1^{(j)} \mathbf{v}_1 + \sum_{l=2}^r d_l \mathbf{u}_l \mathbf{v}_l',$$

with $\mathbf{u}_l = \mathbf{e}_l$ for $l = 2, \ldots, r$, and $\mathbf{v}_l = \mathbf{e}_l$ for $l = 1, \ldots, r$, and $\mathbf{u}_1^{(j)} = \sqrt{1 - \rho_u^2}\mathbf{e}_1 + \frac{\rho_u}{\sqrt{m_u}}\mathbf{p}^{(j)}$, where $\mathbf{p}^{(j)} \in (P)$ and $\rho_u \in (0, 1)$ is to be specified later. As before, each $\mathbf{u}_1^{(j)}$, similarly as $\mathbf{u}_1(\gamma)$ in the proof for (2.9), could be viewed as a perturbation of $\mathbf{e}_1$. However, the differences between these two perturbations are: first of all, in the previous proof, the locations of the perturbation are known to be from $r+1$-th to $r+m_u$-th coordinate, the locations of the perturbation of the current constructions could be anywhere from $r+1$-th to p_u-th coordinate; secondly, the signs of the perturbation of the previous proof are unknown and depending on

the values of γ, the signs of the current perturbation are always positive; lastly, although the sizes of the perturbations are both $\frac{\rho_u}{\sqrt{m_u}}$, the definitions of the ρ_u and m_u in the two perturbations are different, by a factor of logarithm.

The metrics d_M on $\mathbb{R}^{p_u \times p_v}$ or d_U on $\mathbb{R}^{p_u}$ are defined the same as before (2.17) or (2.18).

So, for any $0 \leq j \neq k \leq n$,

$$\begin{aligned} d_M(M_j, M_k) &= \|M_j - M_k\|_F = d_1 \|\mathbf{u}_1^{(j)} - \mathbf{u}_1^{(k)}\| \geq \tfrac{d_1 \rho_u}{\sqrt{2}} \\ d_U(\mathbf{u}_1^{(j)}, \mathbf{u}_1^{(k)}) &= \sin \angle(\mathbf{u}_1^{(j)}, \mathbf{u}_1^{(k)}) \geq \tfrac{\rho_u}{\sqrt{2}} \end{aligned} \tag{2.21}$$

The inequality holds because the construction of $\mathcal{P}$ ensures that $\mathbf{p}^{(j)}$ and $\mathbf{p}^{(k)}$, and hence $\mathbf{u}_1^{(j)}$ and $\mathbf{u}_1^{(k)}$, differ at at least $m_u/2$ coordinates. Finally, let P_{M_j} be the matrix normal distribution $N_{p_u \times p_v}(M_j, I_{p_u} \otimes I_{p_v})$, then for any $j \neq 0$, $P_{M_j} \ll P_{M_0}$, and

$$\begin{aligned} KL(P_{M_j}, P_{M_0}) &= \frac{d_1^2}{2} \|\mathbf{u}_1^{(j)} - \mathbf{u}_1^{(0)}\|^2 \\ &= \frac{d_1^2}{2} \left((1 - \sqrt{1 - \rho_u^2})^2 + m_u \left(\frac{\rho_u}{\sqrt{m_u}} \right)^2 \right) \\ &\leq d_1^2 \rho_u^2 \,. \end{aligned} \tag{2.22}$$

The last inequality holds because $\sqrt{1-x} \geq 1 - x$ for $x \in [0, 1]$ and $\rho_u^2 \in [0, 1]$.

Set

$$\rho_u^2 = c \frac{m_u \log p_u}{d_1^2} \tag{2.23}$$

for a small enough numeric constant c. Then the condition (2.10) of the theorem implies that $\rho_u^2 < 1$ and that the l_q ball constraint (2.19) is satisfied. So, for any

j, $\mathbf{u}_1^{(j)} \in l_{q_u}(s_u)$ and $\Theta_{sub} \subset \Theta'(s_u, q_u; s_v, q_v) \subset \Theta(s_u, q_u; s_v, q_v)$. In addition, the cardinality of the sub parameter space (2.20) implies that

$$\frac{1}{n}\sum_{j=1}^{n} KL(P_{M_j}, P_{M_0}) \leq d_1^2 \rho_u^2 = c m_u \log p_u \leq \beta \log n,$$

for some $\beta \in (0, 1/8)$. Here the first inequality comes from the bound of the KL divergence (2.22), the equality comes from the definition of ρ_u^2 in (2.23) and the second inequality comes from (2.20).

Therefore, for any estimator $\tilde{\mathbf{u}}_1$ and $\tilde{M}$, we apply Lemma 4 together with (2.21) to obtain

$$\max_{M \in \Theta_{sub}} P_M \left(\|\tilde{M} - M\|_F^2 \geq \frac{d_1^2 \rho_u^2}{8} \right) \geq \left(1 - 2\beta - \sqrt{\frac{2\beta}{n}} \right) > 0\,,$$
$$\max_{M \in \Theta_{sub}} P_M \left(\sin^2 \angle(\tilde{\mathbf{u}}_1, \mathbf{u}_1) \geq \frac{\rho_u^2}{8} \right) \geq \left(1 - 2\beta - \sqrt{\frac{2\beta}{n}} \right) > 0\,.$$

Since $\Theta_{sub} \subset \Theta(s_u, q_u; s_v, q_v)$, and $\|M_j\|_F^2 = \sum d_l^2$ for any $M_j \in \Theta_{sub}$, we conclude that

$$\begin{aligned}
& \inf_{\tilde{M}} \sup_{M \in \Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_M(\tilde{M}, M) \\
\geq\ & \inf_{\tilde{M}} \max_{M \in \Theta_{sub}} \mathbb{E}_M L_M(\tilde{M}, M) \\
\geq\ & \frac{d_1^2 \rho_u^2}{8 \sum d_l^2} \inf_{\tilde{M}} \max_{M \in \Theta_{sub}} P_M \left(\|\tilde{M} - M\|_F^2 \geq \frac{d_1^2 \rho_u^2}{8} \right) \\
\gtrsim\ & \rho_u^2 \\
\asymp\ & \frac{m_u \log p_u}{d_1^2} \\
\asymp\ & m_u \epsilon^2\,,
\end{aligned}$$

where $\epsilon^2 = \frac{\log p_u}{d_1^2}$. Similarly, we have

$$\inf_{\tilde{\mathbf{u}}_1} \sup_{M \in \Theta(s_u, q_u; s_v, q_v)} \mathbb{E}_M L_U(\tilde{\mathbf{u}}_1, \mathbf{u}_1) \gtrsim m_u \epsilon^2 \,.$$

By symmetry, repeating the proof with perturbations of $\mathbf{v}_1$ will complete the proof.

2.6.2 Proof of Theorem 4

In this subsection, we will prove Theorem 4, which is the upper bound for the risk of the initialization algorithm 5.

The proof contains three major steps. In the first step, we analyze the subset selection step 1, and relate the subset I defined in (2.12) to other subsets that are easier to handle with. The second step requires a detailed study of the other subsets defined in the first step, which will be the basis of the further analysis of the classical SVD on the reduced matrix in the third step.

Step 1. We will solely focus on the properties related to U and everything will carry over for V.

Let us first define the following random variables: the squared l_2 norm of each row of the observed data matrix X

$$t_{ui} = \sum_{j=1}^{p_v} X_{ij}^2 \,,$$

and the following quantity that can be thought of the scaled sum of the squares

of the i-th row of U

$$\theta_{ui} = \sum_{l=1}^{r} d_l^2 u_{il}^2 \,.$$

Then we know that these random variables follow non-central chi-sqaure distributions with degree of freedom p_v and non-centrality parameters θ_{ui}

$$t_{ui} \sim \chi^2_{p_v}(\theta_{ui}) \,.$$

Recall that the subset of coordinates that are selected in (2.12) depend on the values of t_{ui}'s, which are random. Let us define two closely related deterministic subsets that depend on the values of θ_{ui}'s as following

$$I^{\pm} = \left\{ i : \theta_{ui} \geq a_{\mp} p_v \sqrt{\frac{\log(p_u \vee p_v)}{p_v}} \right\} , \tag{2.24}$$

where $a_{\mp}$ are large enough constants.

Note that

$$\begin{aligned} & I^{\pm} \\ \subset \;& \cup_{l=1}^{r} \left\{ i : u_{il}^2 \geq a_{\mp} \frac{p_v}{d_l^2} \sqrt{\frac{\log(p_u \vee p_v)}{p_v}} \right\} \\ = \;& \cup_{l=1}^{r} \left\{ i : u_{il} \gtrsim \frac{(p_u \log(p_u \vee p_v))^{1/4}}{d_1} \right\} , \end{aligned} \tag{2.25}$$

where the rate of the last term is the same as we mentioned in the remarks after Theorem 4.

Our goal is to establish a "bracketing" relationship (will be accurate in Lemma 1 shortly) so that performing SVD on the random sub-matrix X_{IJ} can be closely

related to performing SVD on the deterministic sub-matrices $X_{I^+J^+}$ and $X_{I^-J^-}$ in Step 3.

Lemma 1. *$I, I^{\pm}$ defined in (2.12, 2.24) satisfy*

$$I^- \subset I \subset I^+ \tag{2.26}$$

with probability 1.

The proof of Lemma 1 is given in the Appendix A.1.1.

Step 2. Before analyzing the classical SVD step, we first derive a bound on the cardinality of the subsets $I^{\pm}$ using the $wl_q(s)$ constraint in the parameter space (2.7). The key message we pursue is that under that sparsity assumption, very few coordinates will be kept in the initialization algorithm, and the resulting variance of the retained coordinates will be shown of a much smaller order than the squared bias term.

From (2.25), we obtain

$$\begin{aligned} |I^{\pm}| &\leq r|\{i : u_{il}^2 \geq a_{\mp}\frac{p_v}{d_l^2}\sqrt{\frac{\log(p_u \vee p_v)}{p_v}}\}| \\ &\leq r|\{i : s_u^2 i^{-2/q_u} \geq a_{\mp}\frac{p_v}{d_l^2}\sqrt{\frac{\log(p_u \vee p_v)}{p_v}}\}| \\ &\lesssim s_u^{q_u}\left(\frac{d_1^2}{\sqrt{p_v \log(p_u \vee p_v)}}\right)^{q_u/2}, \end{aligned} \tag{2.27}$$

where the second inequality comes from the definition of weak l_q ball (2.6) and the last step relies on the assumption that the rank r is constant.

Another key quantity that will be used in Step 3 is the squared bias, that is

induced by focusing only on the large coordinates and estimating the rest by zero. Let g be the solution to $s_u^2 g^{-2/q_u} = a_+ \frac{p_v}{d_l^2}\sqrt{\frac{\log(p_u \vee p_v)}{p_v}}$, then $g = O\left(\left(\frac{s_u^2 d_1^2}{\sqrt{p_v \log(p_u \vee p_v)}}\right)^{q_u/2}\right)$. We then bound the sum of the squares of the coordinates that are not selected for the l-th singular vector

$$\begin{aligned}
\|U_{I^{-c}l}\|^2 &= \sum_{i \notin I^-} u_{il}^2 \\
&\leq \sum_{i: u_{il}^2 \leq a_+ \frac{p_v}{d_l^2}\sqrt{\frac{\log(p_u \vee p_v)}{p_v}}} u_{il}^2 \\
&\leq \sum_i u_{il}^2 \wedge a_+ \frac{p_v}{d_l^2}\sqrt{\frac{\log(p_u \vee p_v)}{p_v}} \\
&\leq \sum_i s_u^2 i^{-2/q_u} \wedge a_+ \frac{p_v}{d_l^2}\sqrt{\frac{\log(p_u \vee p_v)}{p_v}} \\
&\leq \int_y s_u^2 y^{-2/q_u} \wedge a_+ \frac{p_v}{d_l^2}\sqrt{\frac{\log(p_u \vee p_v)}{p_v}} dy \\
&\lesssim s_u^2 g^{1-2/q_u} + a_+ \frac{p_v}{d_l^2}\sqrt{\frac{\log(p_u \vee p_v)}{p_v}} g \\
&\lesssim s_u^{q_u} \left(\frac{\sqrt{p_v \log(p_u \vee p_v)}}{d_1^2}\right)^{1-q_u/2} \\
&= o(1)\,, \qquad (2.28)
\end{aligned}$$

where the first inequality comes from the definition of I^-, the third one comes from the weak l_q constraint, the last equality is from the condition in Theorem 4.

Note that the rate of $\|U_{I^{-c}l}\|^2$ is exactly what we want to prove on the right hand side of the inequality (2.14) in Theorem 4.

Step 3. Given what we have obtained so far, we shall finally apply Lemma 7 in the appendix to prove Theorem 4. To this end, set A and B in Lemma 7 as

follows

$$\begin{aligned} A &= UDV', \\ B &= \begin{bmatrix} X_{IJ} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} U_{I:}DV'_{J:} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} Z_{IJ} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \end{aligned}$$

Then the perturbation matrix is

$$E = B - A = -\begin{bmatrix} \mathbf{0} & U_{I:}DV'_{J^c:} \\ U_{I^c:}DV'_{J:} & U_{I^c:}DV'_{J^c:} \end{bmatrix} + \begin{bmatrix} Z_{IJ} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{2.29}$$

And we have the following bound on the spectral norm of the perturbation matrix E.

Lemma 2. *Define E as in (2.29), we have with high probability converging to 1,*

$$\|E\|^2 \lesssim d_1^2 s_u^{q_u} \left(\frac{\sqrt{p_v \log(p_u \vee p_v)}}{d_1^2} \right)^{1-q_u/2} + d_1^2 s_v^{q_v} \left(\frac{\sqrt{p_u \log(p_u \vee p_v)}}{d_1^2} \right)^{1-q_v/2}.$$

Based on the condition in Theorem 4, we further know that $\|E\|^2 = o(d_1^2)$.

Based on the specification of A, B, E and Lemma 2, let us apply Lemma 7. First of all, $\Sigma_0(A) = \mathbf{0}$ and we therefore have $\alpha = 0$ in (A.2). Secondly, $\sigma_{\min}(\Sigma_1(B)) \geq \sigma_{\min}(\Sigma_1(A)) - \|E\| = d_r - \|E\|$ from the definition of A and Lemma 2. Hence, we can set $\delta = d_r - \|E\|$ and we get $O(\delta) = O(d_1) - o(d_1)$. The definition of ϵ in (A.3) can be trivially upper bounded by $\|E\|$, which, together

with (A.4), leads to

$$\begin{aligned} L_U(\hat{U}^{(0)}, U) &= \|\hat{U}^{(0)}\hat{U}^{(0)\prime} - UU'\|^2 \\ &\leq \frac{\epsilon^2}{\delta^2} \\ &\precsim \frac{\|E\|^2}{d_1^2} \\ &\precsim s_u^{q_u}\left(\frac{\sqrt{p_u \log(p_u \vee p_v)}}{d_1^2}\right)^{1-q_u/2} + s_v^{q_v}\left(\frac{\sqrt{p_v \log(p_u \vee p_v)}}{d_1^2}\right)^{1-q_v/2}, \end{aligned}$$

with probability converging to 1. Since the loss function (2.3) if always bounded above by 1, the high probability can converge to 1 fast enough to make the statement of the theorem true.

2.6.3 Proof of Theorem 3

This section is dedicated to the proof of Theorem 3, which provides the upper bound of the risk of the IT algorithm. We will first prove the risk for estimating the subspace (2.13a), whose establishment will further proves the risk for estimating the mean matrix (2.13b).

In the proof of Theorem 4 in Section 2.6.2 (especially the proof of Lemma 2), it can be seen that the rate of convergence is determined by the tradeoff between two parts: squared bias $\|U_{I^{-c}l}\|^2$ and variance $\frac{|I^+|}{d_1^2}$. For the initialization algorithm, the former dominates the latter by a factor of $\sqrt{p_u \vee p_v \log(p_u \vee p_v)}$ because the cutoff point for the signal that we can detect is $\frac{(p_u \vee p_v \log(p_u \vee p_v))^{1/4}}{d_1}$. In this section, we will prove that the IT algorithm introduced in Section 2.4.2 will recover signal larger than $\frac{\sqrt{\log(p_u \vee p_v)}}{d_1}$, which makes the squared bias and variance off by only a logarithmic factor.

In parallel with the definition of $I^{\pm}$, we define the subsets of coordinates that are of high signals as

$$H_u = \cup_{l=1}^{r} \left\{ i : u_{il}^2 \geq b_u \frac{\log(p_u \vee p_v)}{d_1^2} \right\},$$
$$H_v = \cup_{l=1}^{r} \left\{ j : v_{jl}^2 \geq b_v \frac{\log(p_u \vee p_v)}{d_1^2} \right\}.$$

And let $L_u = H_u^c, L_v = H_v^c$ where H stands for high signal and L stands for low signal. We call a procedure an oracle one if it has the knowledge of H_u, H_v and use superscript ^{o} to indicate oracle quantities.

The proof will be decomposed intro three steps. First, we will show that the classical SVD of the oracle matrix (which will be defined later) achieves the desired rate of convergence. Second, we verify that the output of the IT algorithm with the oracle knowledge is close to the classical SVD of the the oracle matrix. Third, it is proved that the actual IT algorithm behaves like the IT algorithm with the oracle knowledge.

Step 1. We begin by analyzing the properties of H_u, L_u (Replacing $_u$ by $_v$ will produce the corresponding result for V and will be skipped thereafter). By the same calculations as in (2.27) and (2.28), we obtain the upper bounds on the size of the high signal subset and the squared l_2 norm of the low signal coordinates

$$m_u \stackrel{def}{=} |H_u| \leq s_u^{q_u} \left(\frac{\log(p_u \vee p_v)}{d_1^2} \right)^{-q_u/2}, \tag{2.30}$$
$$\|U_{L_u l}\|^2 \leq s_u^{q_u} \left(\frac{\log(p_u \vee p_v)}{d_1^2} \right)^{1-q_u/2}, \tag{2.31}$$

Define A, B, E as in the proof of Theorem 4 with I, J replaced by H_u, H_v

respectively. Utilizing the same tricks in the proof of Lemma 2 in A.1.3, it can be shown that with large probability,

$$\|E\|^2 \lesssim d_1^2\|U_{L_u l}\|^2 + d_1^2\|V_{L_v l}\|^2 + |H_u| + |H_v| = o(d_1^2)\,.$$

Suppose the SVD of the oracle matrix

$$X^o = \begin{bmatrix} X_{H_u H_v} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

is $U^o D^o V^{o\prime}$.

Plugging in (2.30) and (2.31) and applying Lemma 7 again, we have

$$\begin{aligned} & L(U^o, U) + L(V^o, V) \\ \lesssim\ & \frac{\|E\|^2}{d_1^2} \\ \lesssim\ & s_u^{q_u}\left(\frac{\log(p_u \vee p_v)}{d_1^2}\right)^{1-q_u/2} + s_v^{q_v}\left(\frac{\log(p_u \vee p_v)}{d_1^2}\right)^{1-q_v/2} \end{aligned} \tag{2.32}$$

Step 2. Let us first clarify what the IT algorithm with oracle knowledge is exactly. We initialize the IT algorithm with $U^{(0),o}$ and $V^{(0),o}$ which are the outputs of Algorithm 5 with input X^o. We then construct sequences of orthonormal matrices $U^{(k),o}$ and $V^{(k),o}$ for $k = 1, \ldots$ by implementing the IT algorithm, also with input X^o. The goal is to justify that $L(U^o, U^{(\infty),o})$ is upper bounded by the same rate as in the theorem.

For convenience, denote the largest canonical angle between the column spaces of two matrices U_1, U_2 by $\theta(U_1, U_2)$. Then we have the following results

1. The angles satisfy the inequalities

$$\sin\theta(U^o, U^{(k),o}) \leq \varrho \tan\theta(V^o, V^{(k-1),o}) + \omega_u \sec\theta(V^o, V^{(k-1),o}),$$
$$\sin\theta(V^o, V^{(k),o}) \leq \varrho \tan\theta(U^o, U^{(k),o}) + \omega_v \sec\theta(U^o, U^{(k),o}),$$

   where $\varrho = d^o_{r+1}/d^o_r$ and $O(\omega_u) = O(\frac{\sqrt{m_u \log(p_u \vee p_v)}}{d^o_1})$. These two inequalities recursively characterize the evolvement of the angles $\theta(V^o, V^{(k-1),o})$, $\theta(U^o, U^{(k),o})$, $\theta(V^o, V^{(k),o}), \ldots$.

2. If $\sin^2\theta(V^o, V^{(k-1),o}) \lesssim \omega_u^2(1-\varrho)^{-2}$, so is $\sin^2\theta(U^o, U^{(k),o})$; if $\sin^2\theta(U^o, U^{(k),o}) \lesssim \omega_v^2(1-\varrho)^{-2}$, so is $\sin^2\theta(V^o, V^{(k),o})$.

3. Otherwise, the sine of the sequence of the angles will keep decaying

$$\sin^2\theta(U^o, U^{(k),o}) / \sin^2\theta(V^o, V^{(k-1),o}) \leq c,$$
$$\sin^2\theta(V^o, V^{(k),o}) / \sin^2\theta(U^o, U^{(k),o}) \leq c, 0 < c < 1,$$

The proof of the last three claims can be obtained by mimicking the proof of Proposition 6.1 in Ma (2011).

Following the proof of Proposition 6.2 further, one can show that

$$\sin^2\theta(V^o, V^{(\infty),o}), \sin^2\theta(U^o, U^{(\infty),o}) \lesssim \omega_v^2(1-\varrho)^{-2} \vee \omega_u^2(1-\varrho)^{-2}.$$

Since $\varrho \to \frac{d_{r+1}}{d_r} = 0$ and

$$\begin{aligned} \omega_u^2 &\to O(\frac{m_u \log(p_u \vee p_v)}{d_1^2}) \\ &\lesssim s_u^{q_u} \left(\frac{\log(p_u \vee p_v)}{d_1^2} \right)^{-q_u/2} \frac{\log(p_u \vee p_v)}{d_1^2} \\ &= s_u^{q_u} \left(\frac{\log(p_u \vee p_v)}{d_1^2} \right)^{1-q_u/2} , \end{aligned}$$

The last two displays together prove that

$$L(U^o, U^{(\infty),o}) + L(V^o, V^{(\infty),o}) \lesssim (s_u^{q_u} \vee s_v^{q_v}) \left(\frac{\log(p_u \vee p_v)}{d_1^2} \right)^{1-q_u/2} . \tag{2.33}$$

See Yang et al. (2012) for the complete proof of Step 2 and 3.

Step 3. Now that we have acquired the two bounds (2.32) and (2.33), what is left is to check that

$$U^{(k),o} = U^{(k)}, V^{(k),o} = V^{(k)} , \tag{2.34}$$

which is equivalent to verify that

$$U^{(k)}_{L_u:} = \mathbf{0}, V^{(k)}_{L_v:} = \mathbf{0} . \tag{2.35}$$

For $k = 0$, since the deterministic subsets clearly satisfy that

$$I^+ \subset H_u, J^+ \subset H_v ,$$

and we have the bracketing relation, Lemma 1, so (2.34) is correct with high probability for $k = 0$.

We next prove (2.35) by induction. For $i \in L_u, \forall j$,

$$\begin{aligned}
|X_{iH_v} V^{(k)}_{H_v j}| &= |(U_{i:} D V'_{H_v:} + Z_{iH_v}) V^{(k)}_{H_v j}| \\
&= |(U_{i:} D V'_{H_v:} + Z_{iH_v})(V^{o}_{H_v:} V^{o'}_{H_v:} + (V^{o}_{H_v:} V^{o'}_{H_v:})^{\perp}) V^{(k)}_{H_v j}| \\
&\leq |(U_{i:} D V'_{H_v:} + Z_{iH_v}) V^{o}_{H_v j}|(1 + o(1)) \\
&\leq (|d_j u_{ij}| + |Z_{iH_v} V^{o}_{H_v j}|)(1 + o(1)) \\
&\leq (|d_j u_{ij}| + |N(0,1)|)(1 + o(1)) \\
&\lesssim \sqrt{\log(p_u \vee p_v)}\,,
\end{aligned}$$

with high probability. The first inequality is due to ${V^{o'}_{H_v:}}^{\perp} V^{(k)}_{H_v j} = o(1), V^{o'}_{H_v:} V^{(k)}_{H_v j} \leq 1$. The second last uses the induction and the independence of $Z_{iH_v}, i \in L_u$ and $V^{o}_{H_v j}$. The last one comes from the definition of L_u and the tail behavior of normal distribution.

All the statements in this section with high probability can be made to with probability $1 - (p_u \vee p_v)^{-2}$ as long as we choose the constants carefully to make the probabilities summable. Together with the fact that the loss function is bounded above by 2. The high probability statement can be turned into expectation, which finishes the proof.

Combining (2.32, 2.33, 2.34), with triangle inequality and Jensen's inequality completes the proof of (2.13a).

We next turn to the proof of (2.13b). To start, we first derive the convergence

rate of the estimated singular values

$$\begin{aligned}
\hat{d}_l &= \hat{\mathbf{u}}_l' X \hat{\mathbf{v}}_l \\
&= \sum_{\nu=1}^{r} \hat{\mathbf{u}}_l' \mathbf{u}_\nu d_\nu \hat{\mathbf{v}}_l' \mathbf{v}_\nu + \hat{\mathbf{u}}_l' Z \hat{\mathbf{v}}_l \\
&= d_l(1 + O(L_U(\hat{\mathbf{u}}_l, \mathbf{u}_l)))(1 + o(1)) + (\sqrt{m_u \vee m_v} + \sqrt{m_{vl}})(1 + o(1)) \\
&= d_l(1 + O(\frac{\sqrt{m_u \vee m_v}}{d_l}))\,.
\end{aligned}$$

We next bound the Frobenius loss function (2.2) by the the spectral norm loss function (2.3), whose upper bound is already approved.

$$\begin{aligned}
& \|\hat{U}\hat{D}\hat{V}' - UDV'\|_F^2 \\
= \;& \mathrm{tr}(\hat{D}^2) + \mathrm{tr}(D^2) - 2\mathrm{tr}(DV'\hat{V}\hat{D}\hat{U}'U) \\
= \;& \mathrm{tr}(\hat{D}^2) + \mathrm{tr}(D^2) - 2\mathrm{tr}(D\hat{D}) \\
& + 2\mathrm{tr}(D\hat{D}) - 2\mathrm{tr}(D\hat{D}\hat{U}'U) \\
& + 2\mathrm{tr}(D\hat{D}\hat{U}'U) - 2\mathrm{tr}(DV'\hat{V}\hat{D}\hat{U}'U) \\
= \;& \mathrm{tr}((\hat{D} - D)^2) + 2\mathrm{tr}(D\hat{D}(I - \hat{U}'U)) + 2\mathrm{tr}(\hat{D}\hat{U}'UD(I - \hat{V}'V)) \\
\leq \;& \mathrm{tr}((\hat{D} - D)^2 + 2\mathrm{tr}(D\hat{D})\|I - \hat{U}'U\|_2 + 2\mathrm{tr}(D\hat{D})\|I - \hat{V}'V\|_2 \\
= \;& \sum_{l=1}^{r}(d_l - \hat{d}_l)^2 + 2\sum_{l=1}^{r} d_l\hat{d}_l(\|I - \hat{U}'U\|_2 + \|I - \hat{V}'V\|_2) \\
= \;& \sum_{l=1}^{r} d_l^2 O(\frac{m_u \vee m_v}{d_l^2}) + 2\sum_{l=1}^{r} d_l^2(1 + O(\frac{\sqrt{m_u \vee m_v}}{d_l}))(\|I - \hat{U}'U\|_2 + \|I - \hat{V}'V\|_2) \\
= \;& \sum_{l=1}^{r} d_l^2[O(\frac{m_u \vee m_v}{d_l^2}) + O(\|I - \hat{U}'U\|_2 + \|I - \hat{V}'V\|_2)] \\
= \;& \sum_{l=1}^{r} d_l^2 O(\|I - \hat{U}'U\|_2 + \|I - \hat{V}'V\|_2)\,,
\end{aligned}$$

which together with (2.13a) and $\|UDV'\|_F^2 = \sum_{l=1}^{r} d_l^2$ completes the proof.

Appendices

Appendix A

AUXILIARY RESULTS

A.1 Auxiliary Results

The following technical tool for establishing (2.9) in Theorem 2 is from Assouad (1983).

Lemma 3 (Assouad's Lemma)**.** *Let $\Gamma = \{0,1\}^m$ be the set of all binary sequences of length m, applicable to the problem of estimating an arbitrary quantity $\theta(\gamma)$ belonging to a metric space with metric d. Let $\{P_\gamma, \gamma \in \Gamma\}$ be a set of 2^m probability measures and $H(\gamma,\gamma') = \sum_{i=1}^m |\gamma_i - \gamma_i'|$ be the Hamming distance, which counts the number of positions at which γ and γ' differ. For any estimator $\hat{\theta}$ based on an observation from a distribution in the collection $\{P_\gamma, \gamma \in \Gamma\}$,*

$$\sup_{\gamma\in\Gamma} 2^s \mathbb{E} d^s(\hat{\theta}, \theta(\gamma)) \geq \min_{H(\gamma,\gamma')\geq 1} \frac{d^s(\theta(\gamma),\theta(\gamma'))}{H(\gamma,\gamma')} \cdot \frac{m}{2} \cdot \min_{H(\gamma,\gamma')=1} \|P_\gamma \wedge P_{\gamma'}\| .$$

Here, $\|P_\gamma \wedge P_{\gamma'}\|$ is the total variation affinity, defined as $\|P \wedge Q\| = \int (p \wedge q) d\mu$.

The next lemma is used in proving (2.11) in Theorem 2, which is Theorem 2.5 in Tsybakov (2009).

Lemma 4 (Tsybakov (2009)). *For $n \geq 2$, let $\Theta = \{\theta_0, \theta_1, \ldots, \theta_n\}$ be a metric space with metric d, such that $d(\theta_j, \theta_k) \geq 2\delta > 0$ for all $j \neq k$. Consider a collection of distributions $\{P_\theta, \theta \in \Theta\}$ which satisfies that $P_{\theta_j} \ll P_{\theta_0}$ for all $j \neq 0$, and that*

$$\frac{1}{n}\sum_{j=1}^{n} KL(P_{\theta_j}, P_{\theta_0}) \leq \beta \log n, \quad \text{for some } \beta \in (0, 1/8).$$

Suppose $\hat{\theta}$ is an estimator based on an observation from a distribution in the above collection, then

$$\max_{\theta \in \Theta} P_\theta\big(d(\hat{\theta}, \theta) \geq \delta\big) \geq \frac{\sqrt{n}}{1+\sqrt{n}}\left(1 - 2\beta - \sqrt{\frac{2\beta}{\log n}}\right) > 0.$$

The following counting lemma is also used in the proof of (2.11) in Theorem 2.

Lemma 5 (Paul and Johnstone (2007), Lemma 7). *Let p be a positive number and $0 < m \leq p$ be an even number. Let $\mathcal{B}$ be a maximal set of $\{0,1\}^p$ such that*

1. *for any $\mathbf{b} \in \mathcal{B}$, $|\{i : \mathbf{b}_i = 1\}| = m$, and*
2. *for any pair $\mathbf{b}, \widetilde{\mathbf{b}} \in \mathcal{B}$, $|\{i : \mathbf{b}_i = \widetilde{\mathbf{b}}_i = 1\}| \leq m/2$.*

If $m = o(p)$ as $p \to \infty$, then $\log|\mathcal{B}| \gtrsim m \log p$.

The following lemma is used in the proof of the "bracketing" Lemma 1, which in turn is used in the proof of Theorem 4. This lemma is the large deviation result for non-central chi-square distribution.

Lemma 6 (Noncentral Chi-square tail)**.** *There exists constant C, such that for noncentral chi-square distribution,*

$$\begin{aligned} &P(\chi_n^2(n\mu^2) - n - n\mu^2 \geq n\epsilon) \leq \exp(-Cn\epsilon^2)\,, \\ &P(\chi_n^2(n\mu^2) - n - n\mu^2 \leq -n\epsilon) \leq \exp(-Cn\epsilon^2)\,. \end{aligned} \tag{A.1}$$

The following lemma is used in the proof of Theorem 4, which bounds the difference between the subspaces spanned by the singular vectors of one matrix and its perturbation.

Lemma 7 (Wedin (1972))**.** *Suppose two matrices A and $B = A + E$ have the following SVDs*

$$\begin{aligned} A &= U_1(A)\Sigma_1(A)V_1'(A) + U_0(A)\Sigma_0(A)V_0'(A)\,, \\ B &= U_1(B)\Sigma_1(B)V_1'(B) + U_0(B)\Sigma_0(B)V_0'(B)\,. \end{aligned}$$

Assume that

$$\sigma_{\min}(\Sigma_1(B)) \geq \alpha + \delta\,,\ \sigma_{\max}(\Sigma_0(A)) \leq \alpha\,, \tag{A.2}$$

for some $\alpha \geq 0, \delta > 0$. Take

$$\epsilon = \max\{\|U_0(A)'EV_1(B)\|, \|U_1(B)'EV_0(A)\|\}\,, \tag{A.3}$$

then we have

$$\begin{aligned} &\|U_1(A)U_1'(A) - U_1(B)U_1'(B)\| \leq \tfrac{\epsilon}{\delta}\,, \\ &\|V_1(A)V_1'(A) - V_1(B)V_1'(B)\| \leq \tfrac{\epsilon}{\delta}\,. \end{aligned} \tag{A.4}$$

Note that since $\|X\| = \|-X\|$ for any matrix X, we can switch the role of A and B in the definition of ϵ by setting $A = B + (-E)$.

We have the following lemma that gives the bound of the spectral of a random matrix.

Lemma 8 (Davidson and Szarek (2001))**.** *Let Z be a $p_u \times p_v$ random matrix with iid $N(0,1)$ entries, when $p_u \leq p_v$,*

$$P\left(\frac{\|Z\|}{\sqrt{p_v}} > 1 + \sqrt{p_u/p_v} + t\right) \leq \exp(-p_v t^2/2)\,.$$

A.1.1 Proof of Lemma 1

We will prove the two inclusion properties separately.

$$\begin{aligned}
& P(I^- \not\subseteq I) \\
\leq\ & p_u P(\frac{t_{ui}}{p_v} \leq 1 + \alpha_u \sqrt{\frac{\log(p_u \vee p_v)}{p_v}}, \theta_{ui} \geq a_+ p_v \sqrt{\frac{\log(p_u \vee p_v)}{p_v}}) \\
\leq\ & p_u P(\chi^2_{p_v}(\theta_{ui})/p_v \leq 1 + \alpha_u \sqrt{\frac{\log(p_u \vee p_v)}{p_v}}, \theta_{ui} \geq a_+ p_v \sqrt{\frac{\log(p_u \vee p_v)}{p_v}}) \\
\leq\ & p_u P(\chi^2_{p_v}(\theta_{ui})/p_v - 1 - \theta_{ui}/p_v \leq (\alpha_u - a_+)\sqrt{\frac{\log(p_u \vee p_v)}{p_v}}, \\
& \theta_{ui} \geq a_+ p_v \sqrt{\frac{\log(p_u \vee p_v)}{p_v}}) \\
\leq\ & p_u P(\chi^2_{p_v}(\theta_{ui})/p_v - 1 - \theta_{ui}/p_v \leq (\alpha_u - a_+)\sqrt{\frac{\log(p_u \vee p_v)}{p_v}}) \\
\lesssim\ & p_u \exp(-p_v C(\alpha_u - a_+)^2 \frac{\log(p_u \vee p_v)}{p_v}) \\
\lesssim\ & p_u (p_u \vee p_v)^{-a'}\,,
\end{aligned}$$

where in the last step a_+ can be large enough to make $a' > 2$ and $a_+ > \alpha_u$. Here, all the inequalities are straightforward, the second last step is by applying the concentration inequality of the non-central chi-square distribution in Lemma 6.

Similarly,

$$\begin{aligned} P(I \not\subseteq I^+) &\leq p_u P(\chi^2_{p_v}(\theta_{ui})/p_v \geq 1 + \alpha_u \sqrt{\frac{\log(p_u \vee p_v)}{p_v}}, \theta_{ui} \leq a_- p_v \sqrt{\frac{\log(p_u \vee p_v)}{p_v}}) \\ &\leq p_u P(\chi^2_{p_v}(\theta_{ui})/p_v - 1 - \theta_{ui}/p_v \geq (\alpha_u - a_-)\sqrt{\frac{\log(p_u \vee p_v)}{p_v}}) \\ &\lesssim p_u \exp(-p_v C(\alpha_u - a_-)^2 \frac{\log(p_u \vee p_v)}{p_v}) \\ &\lesssim p_u (p_u \vee p_v)^{-a''}, \end{aligned}$$

where a_- is less than α_u, which in turn can be large enough to make $a'' > 2$.

Therefore,

$$\sum P(I^- \not\subseteq I) + P(I \not\subseteq I^+) < \infty .$$

By Borel-Cantelli Lemma, (2.26) holds.

A.1.2 Proof of Lemma 6

In order to prove (A.1), we will use Bernstein Inequality with sub-exponential random variables. To this end, let us first give the related definitions of sub-Gaussian and sub-exponential distributions.

Definition 1 (Sub-gaussian)**.** *We say a random variable Y is sub-gaussian if there*

is $K > 0$, such that

$$(\mathbb{E}|Y|^p)^{1/p} \leq K\sqrt{p}\,,\ \forall p \geq 1\,,$$

and the sub-gaussian norm of Y, denoted by $\|Y\|_{\psi_2}$, is defined to be the smallest K that satisfies the property, i.e.,

$$\|Y\|_{\psi_2} = \sup_{p\geq 1} p^{-1/2}(\mathbb{E}|Y|^p)^{1/p}\,.$$

Definition 2 (Sub-exponential)**.** *We say a random variable Y is sub-gaussian if there is $K > 0$, such that*

$$(\mathbb{E}|Y|^p)^{1/p} \leq Kp\,,\ \forall p \geq 1\,,$$

and the sub-exponential norm of Y, denoted by $\|Y\|_{\psi_1}$, is defined to be the smallest K that satisfies the property, i.e.,

$$\|Y\|_{\psi_1} = \sup_{p\geq 1} p^{-1}(\mathbb{E}|Y|^p)^{1/p}\,.$$

Let X_i be iid $N(\mu, 1)$, because it is well known that a Gaussian random variable is a sub-Gaussian random variable, we know that X_i is sub-Gaussian. Due to the fact that a random variable is sub-Gaussian iff its square is sub-exponential, we know that $X_i^2 \sim \chi_1^2(\mu^2)$ is sub-exponential. Furthermore, it can be easily verified that centering does not affect the property of sub-exponential. Hence, the mean zero random variable defined as $Y_i \stackrel{def}{=} X_i^2 - 1 - \mu^2$ is also sub-exponential. Let

$K = \|Y\|_{\psi_1}$, by Bernstein inequality,

$$P\left(|\sum_{i=1}^{n} Y_i| \geq \epsilon n\right) \leq 2\exp\left(-c\min\left(\frac{\epsilon^2}{K^2}, \frac{\epsilon}{K}\right) n\right) . \tag{A.5}$$

Plugging in the definition of Y_i and observing that $\sum X_i^2 \sim \chi_n^2(n\mu^2)$ proves the desired result.

A.1.3 Proof of Lemma 2

Note that from the definition of E in (2.29), we have

$$\begin{aligned}
\|E\| &\leq \|U_{I:}DV'_{J^c:}\| + \|U_{I^c:}DV'_{J:}\| + \|U_{I^c:}DV'_{J^c:}\| + \|Z_{IJ}\| \\
&\lesssim \sum_{l=1}^{r} (d_l\|U_{Il}\|\|V_{J^cl}\| + d_l\|U_{I^cl}\|\|V_{Jl}\| + d_l\|U_{I^cl}\|\|V_{J^cl}\|) + \sqrt{|I|} + \sqrt{|J|} \\
&\quad \text{by Lemma 8} \\
&\lesssim \sum_{l=1}^{r} (d_l\|V_{J^-{}^cl}\| + d_l\|U_{I^-{}^cl}\| + d_l\|U_{I^-{}^cl}\|\|V_{J^-{}^cl}\|) + \sqrt{|I^+|} + \sqrt{|J^+|} \\
&\quad \text{by Lemma 1} \\
&\lesssim d_1\|V_{J^-{}^cl}\| + d_1\|U_{I^-{}^cl}\| + \sqrt{|I^+|} + \sqrt{|J^+|} \text{ by (2.28)} \\
&= o(d_1)\,, \text{ by (2.28) and (2.27)}\,,
\end{aligned}$$

with probability converging to 1.

By the convexity of the map $x \mapsto x^2$, and plugging in (2.28) and (2.27), we get

$$\begin{aligned}
\|E\|^2 &\lesssim d_1^2\|V_{J^{-c}l}\|^2 + d_1^2\|U_{I^{-c}l}\|^2 + |I^+| + |J^+| \\
&\lesssim d_1^2 s_u^{q_u}\left(\frac{\sqrt{p_v\log(p_u\vee p_v)}}{d_1^2}\right)^{1-q_u/2} + d_1^2 s_v^{q_v}\left(\frac{\sqrt{p_u\log(p_u\vee p_v)}}{d_1^2}\right)^{1-q_v/2} \\
&\quad + s_u^{q_u}\left(\frac{d_1^2}{\sqrt{p_v\log(p_u\vee p_v)}}\right)^{q_u/2} + s_u^{q_u}\left(\frac{d_1^2}{\sqrt{p_v\log(p_u\vee p_v)}}\right)^{q_u/2} \\
&\lesssim d_1^2 s_u^{q_u}\left(\frac{\sqrt{p_v\log(p_u\vee p_v)}}{d_1^2}\right)^{1-q_u/2} + d_1^2 s_v^{q_v}\left(\frac{\sqrt{p_u\log(p_u\vee p_v)}}{d_1^2}\right)^{1-q_v/2},
\end{aligned}$$

which completes the proof.

Bibliography

G. I. Allen, L. Grosenick, and J. Taylor. A generalized least squares matrix decomposition. *Rice University Technical Report No. TR2011-03*, 2011.

O. Alter, P. O. Brown, and D. Botstein. Processing and modeling genome-wide expression data using singular value decomposition. *Proc. Natl. Acad. Sci.*, 97 (18):10101–10106, 2001.

P. Assouad. Deux remarques sur l'estimation. *CR Acad. Sci. Paris Ser. I Math.*, 296(1021-1024):23, 1983.

F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. *CoRR*, abs/0812.1869, 2008.

P. J. Bickel, F. Gotze, and W. R. Van Zwet. Resampling fewer than n observations: gains, losses, and remedies for losses. *Statist. Sinica*, 7:1–32, 1997.

K.R. Davidson and S. Szarek. *Handbook on the Geometry of Banach Spaces*, volume 1, chapter Local operator theory, random matrices and Banach spaces, pages 317–366. Elsevier Science, 2001.

D. L. Donoho. Unconditional bases are optimal bases for data compression and for statistical estimation. *Applied and Computational Harmonic Analysis*, pages 100–115, 1993.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.*, 96:1348–1360, 2001.

K. R. Gabriel. *Journal de la Societe Francaise de Statistique*, 143(3):5–56, 2002.

G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, 1996. ISBN 0801854148.

P. D. Hoff. Model averaging and dimension selection for the singular value decomposition. *J. Am. Stat. Assoc.*, 102(478):674–685, 2007.

S. Holm. A simple sequentially rejective multiple test procedure. *Scand. J. Statist.*, 6:65–70, 1979.

J. Z. Huang, H. Shen, and A. Buja. The analysis of two-way functional data using two-way regularized singular value decompositions. *J. Am. Stat. Assoc.*, 104 (488):1609–1620, 2009.

I. M. Johnstone. Gaussian estimation: Sequence and multiresolution models. Available at `http://www-stat.stanford.edu/~imj/`, 2011.

I. M. Johnstone and A. Y. Lu. On consistency and sparsity for principal components analysis in high dimensions. *J. Am. Stat. Assoc.*, 104(486):682–693, 2009.

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

M. Lee, H. Shen, J. Z. Huang, and J. S. Marron. R code for LSHM, 2010b. URL `http://www.unc.edu/~haipeng/publication/ssvd-code.rar`.

Y. Liu, D. N. N. Hayes, A. Nobel, and J. S. Marron. Statistical significance of clustering for High-Dimension, Low-Sample size data. *J. Am. Stat. Assoc.*, 103 (483):1281–1293, 2008.

A. Y. Lu. *Sparse principal component analysis for functional data*. PhD thesis, Stanford University, Stanford, CA, 2002.

Z. Ma. Sparse principal component analysis and iterative thresholding. 2011.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Learning for Matrix Factorization and Sparse Coding. *J. Mach. Learn. Res.*, 11(1):19–60, 2010.

S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 2009.

B. Nadler. Discussion of "On consistency and sparsity for principal components analysis in high dimensions" by Johnstone and Lu. *J. Am. Stat. Assoc.*, 104 (486):694–697, 2009.

A. B. Owen and P. O. Perry. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *Ann. Appl. Stat.*, 3:564–594, 2009.

D. Paul. Asymptotics of sample eigenstruture for a large dimensional spiked covariance model. *Stat. Sinica*, 17(4):1617–1642, 2007.

D. Paul and I. M. Johnstone. Augmented sparse principal component analysis for high dimensional data. Preprint, available at `http://anson.ucdavis.edu/~debashis/techrep/augmented-spca.pdf`, 2007.

H. S. Prasantha, H. L. Shashidhara, and K. N. Balasubramanya Murthy. Image compression using SVD. In *Proceedings of the International Conference on*

Computational Intelligence and Multimedia Applications, pages 143–145. IEEE Computer Society, 2007.

A. Shabalin and A. Nobel. Reconstruction of a low-rank matrix in the presence of gaussian noise. Preprint, availabel at `http://arxiv.org/abs/1007.4148`, 2010.

D. Shen, H. Shen, and J. S. Marron. Consistency of sparse PCA in high dimension, low sample size contexts. 2011.

H. Shen and J. H. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *J. Multivariate Anal.*, 99:1015–1034, 2008.

M. Sill, S. Kaiser, A. Benner, and A. Kopp-Schneider. Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics*, 27(15):2089–2097, 2011.

A. Thomasian, V. Castelli, and C. Li. Clustering and singular value decomposition for approximate indexing in high dimensional spaces. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 201–207, 1998.

A. B. Tsybakov. *Introduction to nonparametric estimation*. Springer Verlag, 2009.

P. A. Wedin. Perturbation bounds in connection with singular value decomposition. *BIT*, 12:99–111, 1972.

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.

D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *J. Am. Stat. Assoc.*, 105(490):713–726, 2010.

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.

S. Wold. Cross-validatory estimation of the number of components in factor and principal component models. *Technometrics*, 20(4):397–405, 1978.

D. Yang, Z. Ma, and A. Buja. A sparse SVD method for high-dimensional data. Preprint, availabel at `http://arxiv.org/abs/1112.2433`, 2011.

D. Yang, Z. Ma, and A. Buja. Near optimal sparse SVD in high dimensions. 2012.

W. Zheng, S. Z. Li, J. H. Lai, and S. Liao. On constrained sparse matrix factorization. *Computer Vision, IEEE International Conference on*, 0:1–8, 2007.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Comput. Graph. Stat.*, 15:265–286, 2006.