Bi-cross-validation for factor analysis

Art B. Owen
Stanford University

Jingshu Wang
Stanford University

March 2015

Abstract

Factor analysis is over a century old, but it is still problematic to choose the number of factors for a given data set. The scree test is popular but subjective. The best performing objective methods are recommended on the basis of simulations. We introduce a method based on bi-cross-validation, using randomly held-out submatrices of the data to choose the number of factors. We find it performs better than the leading methods of parallel analysis (PA) and Kaiser's rule. Our performance criterion is based on recovery of the underlying factor-loading (signal) matrix rather than identifying the true number of factors. Like previous comparisons, our work is simulation based. Recent advances in random matrix theory provide principled choices for the number of factors when the noise is homoscedastic, but not for the heteroscedastic case. The simulations we choose are designed using guidance from random matrix theory. In particular, we include factors too small to detect, factors large enough to detect but not large enough to improve the estimate, and two classes of factors large enough to be useful. Much of the advantage of bi-cross-validation comes from cases with factors large enough to detect but too small to be well estimated. We also find that a form of early stopping regularization improves the recovery of the signal matrix.

1 Introduction

Factor analysis is a core technology for handling large data matrices, with applications in signal processing [39] bioinformatics [33, 23, 36, 14], finance [11], and other areas [24, 17, 13]. In psychology, the factor model dates back at least to the paper of Spearman in 1904 [35]. The matrix $Y \in \mathbb{R}^{N\times n}$ is represented as a matrix X of some

low rank k (the signal) plus heteroscedastic noise. The signal X in turn can be factored into an $N \times k$ matrix times a $k \times n$ matrix and this (non-unique) factorization may then be interpreted as a product of latent variables times loading coefficients. It is surprisingly difficult to choose the number k of factors. The best performing methods are primarily ad hoc (see for example [31]) and recommendations among them are based largely on simulation studies [9]. Classical hypothesis testing methods based on likelihood ratios do not perform well on big matrices because they are derived in an asymptotic regime with a growing number of observations and fixed number of variables. There are more realistic approaches derived from random matrix theory, but those are not well suited to heteroscedastic noise. As a result, the present state of theory does not provide usable guidelines. This is a significant gap, because the performance of many estimation methods depends critically on the number of factors chosen [14, 20].

In this paper, we develop an approach to choosing the number of factors using bi-cross-validation (BCV) [28]. Our BCV involves holding out some rows and some columns of Y, fitting a factor model to the held-in data and comparing held-out data to corresponding fitted values. We derive our method using recent insights from random matrix theory. We test our method empirically using test cases that are also designed using insights from random matrix theory. Our goal is not to recover the true number k of factors, but instead to choose the number k that lets us best recover the signal matrix X. Using the true number of factors will lead to a noisy estimate of X when some factors are too weak to detect.

Based on previous theoretical results, we employ a taxonomy dividing factors into four types based on their strength in an asymptotic setting. The four factor levels are: undetectable, harmful, helpful, and giant. Giant factors are those that asymptotically explain a fixed percentage of variance in the matrix Y. They become trivial to detect, but surprisingly, their presence causes difficulties for some methods of choosing k. The other factor types are weak and explain a fraction of variance approaching some limit c/N as $n, N \to \infty$ with $N/n \to \gamma$. If c is small compared to a detection threshold, then no method can distinguish that factor from noise, and the factor is undetectable. If c is somewhat larger, then that factor can be detected but the corresponding eigenvectors cannot be estimated accurately enough for that factor to improve estimation of X. Such factors are harmful. We are better off leaving that factor out of estimates $\hat{X}$. Methods that use all detectable factors might fall prey to such factors. If c is still larger, then we can not only detect the factor but including it in $\hat{X}$ yields an improvement. We call those factors helpful. Giant factors are also helpful, but helpful by itself will refer to helpful weak factors. This taxonomy is based on homoscedastic Gaussian noise.

This paper is organized as follows. In section 2 we specify the factor model we study, the asymptotic regime, and our estimation criterion. Section 3 reviews prior work on rank selection and determining the number of factors, both in the field of low-rank matrix recovery and classical factor analysis. It also presents our four level taxonomy of factor sizes. Section 4 describes our early stopping alternation (ESA) algorithm to estimate the low-rank factor matrix with a given target k for the number of factors. Section 5 introduces the BCV technique to determine the number of factors k. Section 6 summarizes extensive simulation results. In those cases BCV is more reliably close to an oracle's performance than either parallel analysis or Kaiser's method. In hard settings with many harmful factors, both of those methods can perform badly. Unlike those methods, BCV becomes more likely to choose the unknown best rank as sample size increases. Section 7 illustrates the BCV choice of k on some data sampled from a meteorite. Section 8 concludes the paper. An Appendix includes a detailed account of the simulations.

2 Problem Formulation

Our data matrix is $Y \in \mathbb{R}^{N\times n}$ with a row for each variable and a column for each observation. In the motivating problems $N \gg n$, but this is not assumed. In a factor model, Y can be decomposed into a low rank signal matrix plus noise:

$$Y = X + \Sigma^{\frac{1}{2}}E = LR + \Sigma^{\frac{1}{2}}E \tag{1}$$

where the low rank signal $X \in \mathbb{R}^{N\times n}$ is a product of factors $L \in \mathbb{R}^{N\times k_0}$ and $R \in \mathbb{R}^{k_0\times n}$, both of rank k_0. The noise matrix $E \in \mathbb{R}^{N\times n}$ has independent and identically distributed (IID) $\mathcal{N}(0,1)$ entries. Each variable has its own noise variance given by $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_N^2)$.

The factor model is usually applied when we anticipate that $k_0 \ll \min(n, N)$. Then identifying those factors suggests possible data interpretations to guide further study. When the factors correspond to real world quantities there is no reason why they must be few in number and then we should not insist on finding them all in our data. We should instead seek the relatively important ones.

In a typical factor analysis, R has n IID columns corresponding to factors and L has nonrandom loadings. We work conditionally on R so that X becomes a fixed unknown matrix. Our goal is to recover X, seeking to minimize

$$\mathrm{Err}_X(\hat{X}) \equiv \mathbb{E}\big(\|\hat{X} - X\|_F^2\big). \tag{2}$$

Definition 1 (Oracle rank and estimate)**.** Let M be a method that for each integer $k \geq 0$ gives a rank k estimate $\hat{X}^M(k)$ of X using Y from model (1). The oracle estimate of X with respect to M is

$$\hat{X}^M_{\text{opt}} = \hat{X}^M\big(k^*_M\big) \tag{3}$$

for an oracle rank

$$k^*_M = \text{argmin}_k\big(\|\hat{X}^M(k) - X\|^2_F\big). \tag{4}$$

Our algorithm has two steps. First we need to devise a method M to effectively estimate X given a rank k. Then with such a method in hand, we need a means to choose k. Section 4 describes our early stopping alternation (ESA) algorithm for estimating X at a given k. Then Section 5 describes our BCV for choosing k for the ESA algorithm. First we describe previous methods that we compare our method to and the relevant random matrix theory that motivates our comparisons.

3 Theory and factor taxonomy

Here we review the most commonly used methods for choosing the number of factors in factor analysis. They are typically based on a limit with $n \to \infty$ while N is fixed. Then we survey some recent work on random matrix theory (RMT) that is related to choosing k but requires constant error variance. Then we use the results of that theory to develop the four level taxonomy of factor sizes that guides our simulations.

3.1 Rank determination for factor analysis

The most popular classical methods for determining the number of factors are the scree test [8, 9], Bartlett's chi-square test [3, 4], Kaiser's rule [21], parallel analysis (PA) [18, 6] and the minimum average partial (MAP) test of [37]. Those methods are aimed at estimating the true number k_0 of factors. They are derived for a setting where $n \to \infty$ with N fixed. In that case, both the maximum-likelihood estimation of the factors and the sample covariance matrix will be consistent. Neither of those hold in the limit we consider.

There is a difference between determining the number of components for PCA and determining the number of factors for FA. Factor analysis has additive heteroscedastic noise that is not present in PCA. The above methods have been modified to be applied to both problems. Many researchers [21, 6, 42, 38] have found out that those methods usually perform much better for PCA than for FA. Some of them

[42, 38] suggest that even for FA, one should perform PCA first in the initial stage to determine the number of factors before estimating the factors. We observed a similar phenomenon in our simulations, and we adopt this suggestion in this paper later when applying the traditional rank determination methods mentioned above.

There is a large amount of evidence [42, 19, 38, 31] that PA is the most accurate of the above classical methods for determining the number of factors. Parallel analysis compares the observed eigenvalues to those obtained in a Monte Carlo simulation. The first factor is retained if and only if its associated eigenvalue is larger than the 95'th percentile of simulated first eigenvalues. For $k \geq 2$, the k'th factor is retained when the first $k-1$ factors were retained and the observed k'th eigenvalue is larger than the 95'th percentile of simulated k'th factors. Horn [18] simulates using IID Gaussian entries in Y, while Buja and Eyuboglu [6] simulate by applying independent uniform random permutations to each of the variables stored in Y. The permutation version is one of the most effective modifications of PA and has been used recently in bioinformatics [23, 36]. Though there exist no theoretical results to guarantee the accuracy of PA, it performs very well in practice. However, it's unclear how it behaves for high-dimensional data.

Besides the classical methods, Bai and Ng [2] developed a series of rules to determine the number of factors for factor models in econometrics. Their theory was developed for large N and n, without any assumptions on the relation between N and n and allowing for correlated noise and even some dependence between the noise and the factors. However, they only consider giant factors, which are quite easy to detect. Bai and Li [1] develop further results for growing N and n, including consistent estimation of Σ but once again restricted to the case of giant factors.

3.2 Some random matrix theory

Recovery of a low-rank matrix in white noise has been well studied in RMT. The model is commonly framed as

$$Y = \sqrt{n} U D V^{\mathsf{T}} + \sigma E \tag{5}$$

where $\sqrt{n} U D V^{\mathsf{T}}$ is the SVD of the signal matrix X, and U and V are $N \times k_0$ and $n \times k_0$ singular vector matrices satisfying $U^{\mathsf{T}} U = V^{\mathsf{T}} V = I_{k_0 \times k_0}$. The diagonal matrix $D = \mathrm{diag}(d_1, d_2, \cdots, d_{k_0})$ defines the strength of each signal. The noise matrix $E \in \mathbb{R}^{N \times n}$ is usually taken to have IID $\mathcal{N}(0,1)$ entries.

Estimation of X is typically through the singular value decomposition (SVD) of Y, retaining the fitted singular vectors, but shrinking or truncating the corresponding singular values. In the limit $N/n \to \gamma$, there is a well known phase transition

threshold of the signal strength for detecting the signals. If $d_i^2 < \sigma^2\sqrt{\gamma}$ then the corresponding factor is not detectable. See [30, 5, 32] for statements of this result.

One way to select the rank is to estimate the number of signals with d_i above the detection threshold. Rao and Edelman [26] use an AIC-based criterion and Kritchman and Nadler [22] developed an algorithm based on a sequence of hypothesis tests to determine the number of detectable signals.

Neither the true rank, nor the number of detectable factors will necessarily optimize our criterion. The problem is that a factor stronger than the detection threshold might still not be strong enough to allow adequate estimation of the corresponding singular vectors. Owen and Perry [28] propose a BCV algorithm to choose k for the truncated SVD, motivated by the loss (2). Perry's dissertation [32] on BCV identifies a higher threshold for d_i^2 beyond which including the corresponding singular vectors reduces the loss (2). He also shows that the rank selected by BCV will track the oracle's rank for truncated SVD; his formal statement is in Theorem 5.3 below. Donoho and Gavish [15] propose a hard-thresholding rule for the empirical singular values of the data that also accounts for this higher threshold.

The above results are only valid in the white noise model (5), which is much simpler than the heteroscedastic model (1) we are considering. For the model which assumes an arbitrary noise covariance, several recent theoretical results have been developed, but none of them solve our problem. For example, Nadler [27] derives some properties of the eigenvalues and eigenvectors under a spiked covariance model with some dependence. The columns of his noise matrix are independent vectors with a covariance whose eigenvalues converge to some limiting distribution. However, he does not show how to explicitly use his results to estimate a spiked covariance model and further, our heteroscedastic model (1) is not directly related to a spiked covariance model. Nadakuditi [25] developed a method to shrink singular values to recover a low-rank signal matrix with noise from a class of distributions more general than IID Gaussian. But he assumed that either the noise matrix or the signal matrix is bi-orthogonally invariant, and to apply his method, one still needs to estimate the rank beforehand.

3.3 Factor categories and test cases

When we simulate the factor model for our tests, we will generate it as

$$Y = \Sigma^{1/2}(\Sigma^{-1/2}X + E) = \Sigma^{1/2}(\sqrt{n}UDV^{\mathsf{T}} + E). \tag{6}$$

The matrix $\Sigma^{-1/2}X = \sqrt{n}UDV^{\mathsf{T}}$ has the same low rank that X does. Here UDV^{T} is an SVD and we generate the matrices U and V from appropriate distributions.

The normalization in (6) allows us to make direct use of RMT in choosing D. The matrix V is uniformly distributed, but U has a non-uniform distribution to avoid making rows with large mean squared U-values coincide with rows having large Σ_i. Such a coincidence could make the problem artificially easy. See the Appendix for a description of the sampler.

Based on the discussion in Section 3.2, we may place each factor into a category depending on the size of d_i^2. The categories are:

1. Undetectable: d_i^2 is below the detection threshold.

2. Harmful: d_i^2 is above the detection threshold but below the threshold at which their inclusion in the model improves accuracy.

3. Useful: d_i^2 is above the detection threshold but is $O(1)$. It contributes an $N \times n$ matrix to Y with sum of squares $O(n)$, while the expected sum of squared errors is $nN\sigma^2$.

4. Giant: d_i^2 grows proportionally to N. The factor sum of squares is then proportional to the noise level. Such factors are easy to detect.

Undetectable factors essentially add to the noise level. Asymptotically, no method can detect them, and so they play a small role in determining which method to choose k is best.

Harmful factors can cause severe difficulties for factor models. They are large enough to be detected but including them makes the loss (2) larger. Changing an algorithm to better detect such factors could lead it to have worse performance.

Useful weak factors are large enough that including them reduces the loss. It is generally not possible to estimate their corresponding eigenvectors consistently. The estimated and true eigenvectors only converge in a limit where d_i^2 is an arbitrarily large constant. Separating useful from harmful weak factors is important for accurate estimation of X.

The giant factors are large enough to be almost unmissable. When one or more of them is present they may very well put a clear knee in the scree plot, though that knee won't necessarily be at the optimal k when there are also some useful weak factors. When some giant factors are much larger than others, it is also possible for the scree plot to have some its most prominent knee in a place that leads one to omit one or more giant factors.

Real data often have giant factors. In a matrix of dimensional measurements on animals, there is likely to be a giant factor for the overall size of those animals. In educational testing data where n students each answer N questions there is very often

	easy				hard	
# Undetectable	1	1	1	1	1	1
# Harmful	1	1	1	1	6	6
# Useful	6	5	3	0	1	0
# Giant	0	1	3	6	0	1

Table 1: Six cases of factor strengths considered in our simulations. Hard cases have mostly harmful factors; easy cases have just one.

a giant factor interpreted as student ability with a corresponding loading for item difficulty. In modeling daily returns of stocks there may be one factor corresponding to overall market movements that affect all stocks. Although giant factors are quite easy to detect, they can cause severe difficulties for some algorithms. Useful weak factors may appear negligible in comparison to the giant ones.

In the following sections we compare methods using six testing scenarios described in Table 1. In all of these cases there are eight nonzero factors. The easy cases have only one harmful factor. The hard cases have six such factors. Every case has one undetectable factor. Among the hard cases one has a giant factor and the other has a useful weak factor instead. Among the easy cases, the number of giant factors is 0 or 1 or 3 or 6 and the remaining ones are useful weak factors. See Table 1.

In the white noise model, the category that a factor falls into depends on the ratio $d_i^2/(\sigma^2\sqrt{\gamma})$. When we simulate factors we use the same critical ratios but replace σ^2 by $(1/N)\sum_{i=1}^N \sigma_i^2$.

For each of these six cases we consider various levels of noise variance. The σ_i^2 are independent inverse gamma random variables with mean 1 and variances 0 or 1 or 10. We also consider 5 aspect ratios, $N/n \in \{0.02, 0.2, 1, 5, 20\}$. For each aspect ratio we consider two sizes n. That is, we consider $6 \times 3 \times 5 \times 2 = 180$ cases spanning a wide range of problems. The complete details are in the Appendix.

4 Estimating X given the rank k

Here we consider how to estimate X using exactly k factors. This will be the inner loop for an algorithm that tries various k. Because the components of E in (1) are IID $\mathcal{N}(0,1)$, we get the log-likelihood function:

$$\log \mathcal{L}(X, \Sigma) = -\frac{Nn}{2}\log(2\pi) - \frac{n}{2}\log\det\Sigma + \operatorname{tr}\Big[-\frac{1}{2}\Sigma^{-1}(Y-X)(Y-X)^{\mathsf{T}}\Big]. \quad (7)$$

If Σ were known it would be straightforward to estimate X, but Σ is unknown. Given an estimate of X it is straightforward to optimize the likelihood over Σ. Next we describe our alternating algorithm and we employ an early stopping rule to regularize it.

The truncated SVD of a matrix Y is

$$Y(k) = U(k)D(k)V(k)^{\mathsf{T}} \tag{8}$$

where $D(k)$ is the diagonal matrix of the k largest singular values of Y, and $U(k)$ and $V(k)$ are the matrices of the corresponding singular vectors. We start with an initial estimate

$$\hat{\Sigma} = \mathrm{diag}\Big(\Big(Y - \frac{1}{n}Y\mathbf{1}_{n\times n}\Big)\Big(Y - \frac{1}{n}Y\mathbf{1}_{n\times n}\Big)^{\mathsf{T}}\Big). \tag{9}$$

Given an estimate $\hat{\Sigma}$, our rank k estimate $\hat{X}$ is the truncated SVD of the reweighted matrix $\tilde{Y} = \hat{\Sigma}^{-\frac{1}{2}}Y$:

$$\hat{X} = \hat{\Sigma}^{\frac{1}{2}}\tilde{Y}(k). \tag{10}$$

Given an estimate $\hat{X}$, our new variance estimate $\hat{\Sigma}$ contains the mean squares of the residuals:

$$\hat{\Sigma} = \frac{1}{n}\mathrm{diag}\big[\big(Y - \hat{X}\big)\big(Y - \hat{X}\big)^{\mathsf{T}}\big]. \tag{11}$$

Both of the above two steps can increase $\log \mathcal{L}(X, \Sigma)$ but not decrease it. Simply alternating those two steps to convergence is not effective. The algorithm often does not converge. Nor should it, because the likelihood is unbounded as even one the σ_i decreases to zero. A similar degenerate problem arises when one tries to fit real valued data to a mixture of two Gaussians. In that case the likelihood is unbounded as one of the mixture components converges to a point mass.

It is not straightforward to prevent σ_i from approaching 0. Imposing a bound $\sigma_i \geq \epsilon > 0$ leads to some σ_i converging to ϵ. There are numerous approaches to regularizing $\hat{X}$ to prevent $\sigma_i \to 0$. One could model the σ_i as IID from some prior distribution. However, such a distribution must also avoid putting too much mass near zero. We believe that this transfers the singularity avoidance problem to the choice of hyperparameters in the σ distribution and does not really solve it. We have also found in trying it that even when σ_i are really drawn from our prior, the algorithm still converged towards some zero estimates.

A second, related approach is to employ a penalized likelihood

$$L_{\mathrm{reg}}\big(Y,\lambda,\hat{X},\hat{\Sigma}\big) = -n\log\det\hat{\Sigma} + \mathrm{tr}\big[\Sigma^{-1}(Y-\hat{X})(Y-\hat{X})^{\mathsf{T}}\big] + \lambda P\big(\hat{\Sigma}\big), \tag{12}$$

where P penalizes small components σ_i. This approach has two challenges. The penalty P must be strong enough to ensure that the likelihood is bounded, but it is not clear how strong would be too strong. Additionally, it requires a choice of λ. Tuning λ by cross-validation within our bi-cross-validation algorithm is unattractive. Also there is a risk that cross-validation might choose $\lambda = 0$ allowing one or more $\sigma_i \to 0$.

We do not claim that these methods cannot in the future be made to work. They are however not easy to use, and we found a simpler approach that works well. Our approach is to employ early stopping. We start at (9) and iterate the pair (10) and (11) some number m of times and then stop.

To choose m, we investigated 180 test cases based on the six factor designs in Table 1, three dispersion levels for the σ_i^2, five aspect ratios γ and 2 data sizes. The details are in the Appendix. The finding is that taking $m = 3$ works almost as well as if we used whichever m gave the smallest error for each given data set.

More specifically, define the oracle estimating error using early stopping at m steps as

$$\mathrm{Err}_X(m) = \min_k \|\hat{X}^m(k) - X\|_F^2 \tag{13}$$

where $\hat{X}^m(k)$ is the estimate of X using m iterations and rank k. We judge each number m of steps, by the best k that might be used with it.

For early stopping alternation (ESA), we define the oracle stopping number of steps on a data set as

$$m_{\mathrm{Opt}} = \mathrm{argmin}_m\, \mathrm{Err}_X(m). \tag{14}$$

We have found that $m = 3$ is very nearly optimal in almost all cases. We find that $\mathrm{Err}_X(3)/\mathrm{Err}_X(m_{\mathrm{Opt}})$ is on average about 1.01. Using $m = 3$ steps with the best k is nearly as good as using the best possible combination of m and k. We then turn our attention to trying to get the best possible k in Section 5. Early stopping has the additional benefit that it is fast compared to trying to iterate to convergence.

Remark 4.1. Early-stopping of iterative algorithms is a well-known regularization strategy for inverse problems and training machine learning models like neural networks and boosting [40, 41, 16, 7]. An equivalence between early-stopping and adding a penalty term has been demonstrated in some settings [12, 34].

Remark 4.2. ESA starting from (9) with $m = 1$ is equivalent to the applying the widely used standardization step before applying SVD. Using $m > 1$ iterations can be interpreted as using an estimated signal matrix to improve the estimation of Σ, so ESA with $m = 3$ can be understood as applying truncated SVD on a more properly reweighted data than one gets with $m = 1$. We find that $\mathrm{Err}_X(3)/\mathrm{Err}_X(1)$ is on average about 0.89 in the Appendix.

5 Bi-cross-validatory choice of k

Here we describe how BCV works in the heteroscedastic noise setting. Then we give our choice for the shape and size of the held-out submatrix using theory from [32].

5.1 Bi-cross-validation to estimate $k^*_{\mathbf{ESA}}$

We want k to minimize the squared estimation error (4) in $\hat{X}$. We adapt the BCV technique of Owen and Perry [28] to this setting of unequal variances. We randomly select n_0 columns and N_0 rows as the held-out block and partition the data matrix Y (by permuting the rows and columns) into four folds,

$$Y = \begin{pmatrix} Y_{00} & Y_{01} \\ Y_{10} & Y_{11} \end{pmatrix}$$

where Y_{00} is the selected $N_0 \times n_0$ held-out block, and the other three blocks Y_{01}, Y_{10} and Y_{11} are held-in. Correspondingly, we partition X and Σ as

$$X = \begin{pmatrix} X_{00} & X_{01} \\ X_{10} & X_{11} \end{pmatrix}, \quad \text{and} \quad \Sigma = \begin{pmatrix} \Sigma_0 & 0 \\ 0 & \Sigma_1 \end{pmatrix}.$$

The idea is to use the three held-in blocks to estimate X_{00} for each candidate rank k and then select the best k based on the BCV estimated prediction error.

We rewrite the model (1) in terms of the four blocks:

$$\begin{aligned} \begin{pmatrix} Y_{00} & Y_{01} \\ Y_{10} & Y_{11} \end{pmatrix} &= \begin{pmatrix} X_{00} & X_{01} \\ X_{10} & X_{11} \end{pmatrix} + \begin{pmatrix} \Sigma_0 & 0 \\ 0 & \Sigma_1 \end{pmatrix}^{\frac{1}{2}} \begin{pmatrix} E_{00} & E_{01} \\ E_{10} & E_{11} \end{pmatrix} \\ &= \begin{pmatrix} L_0 R_0 & L_0 R_1 \\ L_1 R_0 & L_1 R_1 \end{pmatrix} + \begin{pmatrix} \Sigma_0^{\frac{1}{2}} E_{00} & \Sigma_0^{\frac{1}{2}} E_{01} \\ \Sigma_1^{\frac{1}{2}} E_{10} & \Sigma_1^{\frac{1}{2}} E_{11} \end{pmatrix} \end{aligned}$$

where $L = \begin{pmatrix} L_0 \\ L_1 \end{pmatrix}$ and $R = \begin{pmatrix} R_0 & R_1 \end{pmatrix}$ are decompositions of the factors.

The held-in block

$$Y_{11} = X_{11} + \Sigma_1^{\frac{1}{2}} E_{11}$$

has the low-rank matrix plus noise form, so we can use ESA to get estimates $\hat{X}_{11}(k)$ and $\hat{\Sigma}_1$ for a given rank k. Next for $k < \text{rank}(Y_{11})$ we choose rank k matrices $\hat{L}_1$ and $\hat{R}_1$ with

$$\hat{X}_{11}(k) = \hat{L}_1 \hat{R}_1. \tag{15}$$

Then we can estimate L_0 by solving N_0 linear regression models

$$Y_{01}^{\mathsf{T}} = \hat{R}_1^{\mathsf{T}} L_0^{\mathsf{T}} + E_{01}^{\mathsf{T}} \Sigma_0^{\frac{1}{2}},$$

and estimate R_0 by solving n_0 weighted linear regression models

$$Y_{10} = \hat{L}_1 R_0 + \hat{\Sigma}_1^{\frac{1}{2}} E_{10}.$$

These least square solutions are

$$\hat{R}_0 = (\hat{L}_1^{\mathsf{T}} \hat{\Sigma}_1^{-1} \hat{L}_1)^{-1} \hat{L}_1^{\mathsf{T}} \hat{\Sigma}_1^{-1} Y_{10}, \quad \text{and} \quad \hat{L}_0 = Y_{01} \hat{R}_1^{\mathsf{T}} (\hat{R}_1 \hat{R}_1^{\mathsf{T}})^{-1}$$

which do not depend on the unknown Σ_0. We get a rank k estimate of X_{00} as

$$\hat{X}_{00}(k) = \hat{L}_0 \hat{R}_0. \tag{16}$$

Though the decomposition (15) is not unique, the estimate $\hat{X}_{00}(k)$ is unique. To prove it we need a reverse order theorem for Moore-Penrose inverses. For a matrix $Z \in \mathbb{R}^{n \times d}$, the Moore-Penrose pseudo-inverse of Z is denoted Z^+.

Theorem 5.1. *Suppose that $X = LR$, where $L \in \mathbb{R}^{m \times r}$ and $R \in \mathbb{R}^{r \times n}$ both have rank r. Then $X^+ = R^+ L^+ = R^{\mathsf{T}}(RR^{\mathsf{T}})^{-1}(L^{\mathsf{T}}L)^{-1}L^{\mathsf{T}}$.*

Proof. This is MacDuffee's theorem. There is a proof in [28]. □

Proposition 5.2. *The estimate $\hat{X}_{00}(k)$ from* (16) *does not depend on the decomposition of $\hat{X}_{11}(k)$ in* (15) *and has the form*

$$\hat{X}_{00}(k) = Y_{01}\big(\hat{\Sigma}_1^{-\frac{1}{2}} \hat{X}_{11}(k)\big)^{+} \hat{\Sigma}_1^{-\frac{1}{2}} Y_{10}. \tag{17}$$

PROOF. Let $\hat{X}_{11}(k) = \hat{L}_1\hat{R}_1$ be any decomposition satisfying (15). Then

$$\begin{aligned}\hat{X}_{00} &= \hat{L}_0\hat{R}_0 \\ &= Y_{01}\hat{R}_1^{\mathsf{T}}\big(\hat{R}_1\hat{R}_1^{\mathsf{T}}\big)^{-1}\big(\hat{L}_1^{\mathsf{T}}\hat{\Sigma}_1^{-1}\hat{L}_1\big)^{-1}\hat{L}_1^{\mathsf{T}}\hat{\Sigma}_1^{-1}Y_{10} \\ &= Y_{01}\big(\hat{\Sigma}_1^{-\frac{1}{2}}\hat{L}_1\hat{R}_1\big)^{+}\hat{\Sigma}_1^{-\frac{1}{2}}Y_{10} \\ &= Y_{01}\big(\hat{\Sigma}_1^{-\frac{1}{2}}\hat{X}_{11}(k)\big)^{+}\hat{\Sigma}_1^{-\frac{1}{2}}Y_{10}.\end{aligned} \tag{18}$$

The third equality follows from Theorem 5.1. □

Next, we define the cross-validation prediction average squared error for block Y_{00} as

$$\widehat{\mathrm{PE}}_k(Y_{00}) = \frac{1}{n_0N_0}\big\|Y_{00} - \hat{X}_{00}(k)\big\|_F^2.$$

Notice that as the partition is random, we have:

$$\mathbb{E}\big(\widehat{\mathrm{PE}}_k(Y_{00})\big) = \mathbb{E}\left\{\frac{1}{n_0N_0}\mathrm{Err}_{X_{00}}\big(\hat{X}_{00}(k)\big)\right\} + \frac{1}{N}\sum_{i=1}^{N}\sigma_i^2,$$

where $\mathrm{Err}_X(\hat{X})$ is the loss defined at (2). The expectation is over the noise and the random partition, for a fixed signal matrix.

The above random partitioning step is repeated independently R times, yielding the average BCV mean squared prediction error for Y,

$$\widehat{\mathrm{PE}}(k) = \frac{1}{R}\sum_{r=1}^{R}\widehat{\mathrm{PE}}_k(Y_{00}^{(r)}).$$

The BCV estimate of k is then

$$\hat{k}^* = \mathrm{argmin}_k\,\widehat{\mathrm{PE}}(k). \tag{19}$$

We investigate integer values of k from 0 to some maximum. We cannot take k as large as $\min(n_1, N_1)$ where $n_1 = n - n_0$ and $N_1 = N - N_0$, for then we will surely get $\sigma_i = 0$ even with early stopping. We impose an additional constraint on k to keep the diagonal of $\hat{\Sigma}_1$ away from zero. If for some k we observe that

$$\frac{1}{N_1}\sum_{i=1}^{N_1}\log_{10}\big(|\hat{\sigma}_{i,1}^{(k)}|\big) < -6 + \log_{10}\big(\max_i|\hat{\sigma}_{i,1}^{(k)}|\big) \tag{20}$$

where $\hat{\Sigma}_1(k) = \mathrm{diag}\big(\hat{\sigma}_{1,1}^{(k)}, \hat{\sigma}_{2,1}^{(k)}, \cdots, \hat{\sigma}_{N_1,1}^{(k)}\big)$, then we do not consider any larger values of k. The condition (20) means that the geometric mean of the variance estimates is below 10^{-6} times the largest one.

Remark 5.1. Owen and Perry [28] mentioned that BCV can miss large but very sparse components in the SVD in a white noise model, and they suggested rotating the data matrix as a remedy. However, in our problem where the noise is heteroscedastic, there will be an identifiability issue between factors and noise if the factors are too sparse and the support of the low rank matrix is concentrated in a few locations (see for example [10]). Thus, we only investigate cases where the factor matrix X is not sparse, and do not use rotation to remove sparseness.

5.2 Choosing the size of the holdout Y_{00}

We define the true prediction error for ESA as:

$$\mathrm{PE}(k) = \frac{1}{nN}\big\|X - \hat{X}(k)\big\|_F^2 + \frac{1}{N}\sum_i \sigma_i^2$$

and then the oracle rank is

$$k^*_{\mathrm{ESA}} = \mathrm{argmin}_k\, \mathrm{PE}(k).$$

Ideally, we would like $\widehat{\mathrm{PE}}(k)$ be a good estimate of $\mathrm{PE}(k)$. For good estimation of X it suffices to have $\hat{k}^*$ (defined in (19)) be a good estimate of k^*_{ESA}.

When it is known that $\Sigma = \sigma^2 I$, we can use the truncated SVD to estimate X and for BCV the estimate of X_{00} simplifies to

$$\hat{X}_{00}(k) = Y_{01}\big(Y_{11}(k)\big)^{+} Y_{10}, \tag{21}$$

where $Y_{11}(k)$ is the truncated SVD in (8). Perry [32] proved that $\hat{k}^*$ and k^*_{ESA} track each other asymptotically if the relative size of the held-out matrix Y_{00} satisfies the following theorem.

Theorem 5.3. *For model (5), if k_0 is fixed and $\frac{N}{n} \to \gamma$ as $n \to \infty$, then $\hat{k}^*$ and* $\mathrm{argmin}_k\, \mathbb{E}(\widehat{\mathrm{PE}}_k(Y_{00}))$ *converge to the same value if*

$$\sqrt{\rho} = \frac{\sqrt{2}}{\sqrt{\bar{\gamma}} + \sqrt{\bar{\gamma} + 3}} \tag{22}$$

holds, where

$$\bar{\gamma} = \left(\frac{\gamma^{1/2} + \gamma^{-1/2}}{2}\right)^2, \quad \textit{and} \quad \rho = \frac{n - n_0}{n} \cdot \frac{N - N_0}{N}.$$

Here ρ is the fraction of entries from Y in the held-in block Y_{11}. The larger $\bar{\gamma}$ is, the smaller ρ will be, thus ρ reaches its maximum when Y is square with $\gamma = 1$. For example, when $\gamma = 1$, then $\rho \approx 22\%$. In contrast, if $\gamma = 50$ or 0.02, ρ then drops to only 3.5%.

Theorem 5.3 compares the best k for $\mathbb{E}(\widehat{\mathrm{PE}}_k)$ to the best k for the true error. [28] found that the BCV curve under repeated subsampling was remarkably stable for large matrices, and so ordinarily the best rank per sample will be close to the one that is best on average.

In our simulations, we use (22) to determine the size of Y_{00}. Further, to determine n_0 and N_0 individually, we make Y_{11} as square as possible as long as $n_0 \geq 1$ and $N_0 \geq 1$. For instance, with $\gamma = 1$ we hold in just under half the rows and columns of the data.

6 Simulation Results

We use simulation scenarios described in Section 3.3 and the Appendix. Those simulations have $\mathbb{E}(\sigma_i^2) = 1$ but fall into three different groups: white noise with $\mathrm{Var}(\sigma_i^2) = 0$, mild heteroscedasticity with $\mathrm{Var}(\sigma_i^2) = 1$ and strong heteroscedasticity with $\mathrm{Var}(\sigma_i^2) = 10$. In this section we begin by summarizing the mild heteroscedastic case. The other cases are similar and we give some results for them later.

To measure the loss in estimating X due to using an estimate $\hat{k}$ instead of the optimal choice k^*_{ESA} we use a relative estimation error (REE) given by

$$\mathrm{REE}(\hat{k}) = \frac{\|\hat{X}(\hat{k}) - X\|_F^2}{\|\hat{X}(k^*_{\mathrm{ESA}}) - X\|_F^2} - 1.$$

REE is zero if $\hat{k}$ is the best possible rank for the specific data matrix shown, that is, if $\hat{k}$ is the same rank an oracle would choose.

We compare our BCV technique with the three most widely used methods to choose the number of factors: Horn's Gaussian Monte Carlo version of PA [18], Buja and Eyuboglu's random permutation version of PA [6], and Kaiser's rule [21]. We also include in the comparison the use of the true number of factors as well as the oracle's number of factors k^*_{ESA} defined in (4). We use each method to estimate the number of factors. All the methods employ the same X estimation technique, ESA with $m = 3$. If they have chosen a good value for k, then they will attain a small error. We do not compare to the scree plot, as that method involves a subjective human opinion as to where the dividing line is between large and small eigenvalues.

Table 2 shows the worst case relative estimating error for each method. We see that BCV is the minimax method for all 10 matrix sizes, and has a worst case REE substantially smaller than the other three methods. The hardest situation for BCV is when there is a mixture of giant factors and weak useful factors. Kaiser's rule has difficulty on the hard case with lots of harmful factors. The two versions of PA perform similarly here. Tables 6 and 7 in the Appendix provide more details of the simulation results for this mildly heteroscedastic case.

$$\mathrm{Var}(\sigma_i^2) = 1$$

(N, n)	BCV		PA-Horn		PA-B&E		Kaiser	
	REE	Case	REE	Case	REE	Case	REE	Case
(20, 1000)	0.23	easy-1	0.56	hard-0	0.58	hard-0	1.32	hard-0
(100, 5000)	0.38	easy-3	0.74	hard-0	0.73	hard-0	6.18	hard-0
(20, 100)	0.65	easy-0	1.01	easy-6	1.04	easy-6	0.91	hard-0
(200, 1000)	0.18	easy-3	2.10	hard-0	2.15	hard-0	5.75	hard-1
(50, 50)	0.34	easy-3	1.12	easy-1	1.07	easy-1	2.40	hard-0
(500, 500)	0.06	easy-3	2.88	hard-0	2.92	hard-0	5.55	hard-1
(100, 20)	0.55	easy-1	1.31	easy-1	1.33	easy-1	2.57	hard-1
(1000, 200)	0.03	easy-3	2.25	hard-0	2.25	hard-0	5.55	hard-1
(1000, 20)	0.12	easy-3	0.78	hard-0	0.78	hard-0	4.51	hard-1
(5000, 100)	0.03	hard-0	1.23	hard-0	1.22	hard-0	5.68	hard-1

Table 2: Worst case REE($\hat{k}$) for each method. 'PA-Horn' is Horn's PA. 'PA-B&E' is Buja and Eyuboglu's permutation PA. For each method, the left column shows the worst case REE($\hat{k}$) and the right column identifies that case from Table 1, as either easy or hard plus the number of giant factors.

In 61.9% of the 6000 simulation samples, $\hat{k}^* = k^*_{\mathrm{ESA}}$, meaning that BCV gives same number of factors as an oracle would. Among the 3000 simulation samples of the larger sized matrices for each aspect ratio, the match rate was 78.2%. Figure 1 shows for different methods, the proportion of simulations with REE above certain values, both for all the samples and samples of the larger sized matrices. BCV does best by this measure. BCV is based on Perry's asymptotic Theorem 5.3 and the right panel of Figure 1 shows that BCV improves at larger sample sizes. PA performed roughly as well on large and small samples. The Kaiser method actually got worse on larger samples.

$$\mathrm{Var}(\sigma_i^2) = 1$$

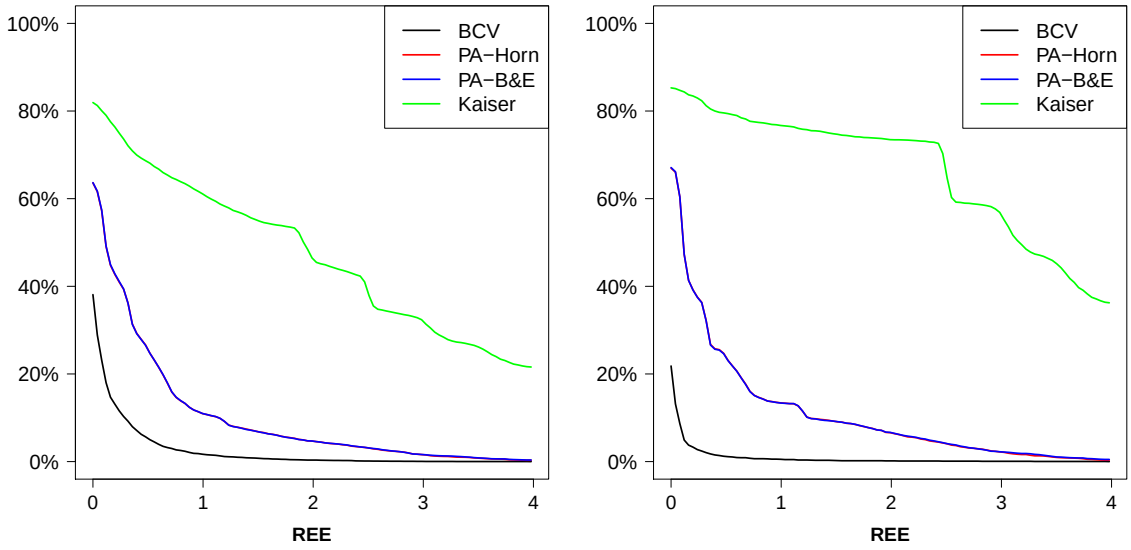

Figure 1: The vertical axis shows the proportion of samples with REE exceeding the number on the horizontal axis. The left panel shows all 6000 samples. The right panel shows only the 3000 simulations of larger matrices. As the two version of PA provide similar estimates, the blue line almost hides the red line.

To understand the difference between PA and BCV, see Figure 2, which plots the distribution of $\hat{k}$ for all methods and all 6 cases, on 500×500 data matrices. BCV closely tracks the oracle. The PA methods have trouble identifying the useful weak factors when giant factors are present. This is most extreme in the easy case with 3 giant factors. They also have trouble rejecting the detectable but not useful factors in the hard case with no giant factor. One sees a lot of variance in the number of factors chosen there by PA. As stated in [6], because of the summation constraint of the eigenvalues of the sample correlation matrix, "the larger eigenvalues of low order are, the smaller and less significant subsequent eigenvalues of higher order get", the performance of methods like PA and Kaiser's rule which compares the sample correlation eigenvalues with the null can be greatly influenced by the distribution of factor strength. In contrast, BCV seems to be much more robust against the change of factor strength distribution. The true number of factors was often much larger than the oracle's number and BCV was able to match the oracle.

Both versions of PA performed very similarly to each other. This is perhaps not

$\mathrm{Var}(\sigma_i^2)$	BCV	PA-Horm	PA-B&E	Kaiser
0	0.35	1.89	1.88	3.94
1	0.67	2.88	2.92	6.18
10	0.72	3.44	3.44	8.78

Table 3: Worst case REE values for each method of choosing k for white noise and two heteroscedastic noise settings.

surprising since the Gaussian noise we use is consistent with both models. But the Buja and Eyoboglu approach does not use the full power of the Gaussian assumption. It merely permutes data instead. That it closely matches Horn's parallel analysis in this case is a point in its favor.

6.1 Other cases

Table 2 shows that when $\mathrm{Var}(\sigma_i^2) = 1$ then BCV is minimax among the compared methods in all 12 sample sizes simulated. Figure 2 shows that BCV becomes more accurate on the larger sample sizes, Kaiser becomes less accurate and PA hardly changes. Both of those remain true in the white noise case as well as in the strongly heteroscedastic case.

Table 3 summarizes worst case REE values for the four methods we compare at each heteroscedasticity type. As the variance of σ_i^2 rises it becomes more difficult to attain a small REE. BCV is mildly affected, Kaiser is severely affected, while the two PA methods are intermediate and very close to each other.

6.2 Example trajectory

BCV chose the oracle's value of k more often than not in our simulations. Figure 3 depicts two simulation realizations where that happened. It plots the cross-validation error $\widehat{\mathrm{PE}}(k)$ and the ESA true prediction error $\mathrm{PE}(k)$ for two simulated data sets. Both are from the easy scenario of Table 1, with no giant factors. The setting was mildly heteroscedastic with $\mathbb{E}(\sigma_i^2) = \mathrm{Var}(\sigma_i^2) = 1$. One has size 500×500 and the other is 100×5000. In these instances, $\widehat{\mathrm{PE}}(k)$ (based on $R = 12$ replicates) estimates $\mathrm{PE}(k)$ quite well for $k < k^*_{\mathrm{ESA}}$. For higher k, the estimates are biased. The true and estimated prediction errors take their minima at the same ranks in these realizations, so in these cases BCV picks the best rank. Perry's remarkable Theorem 5.3 gives conditions for the minima to align asymptotically in the homoscedastic noise setting.

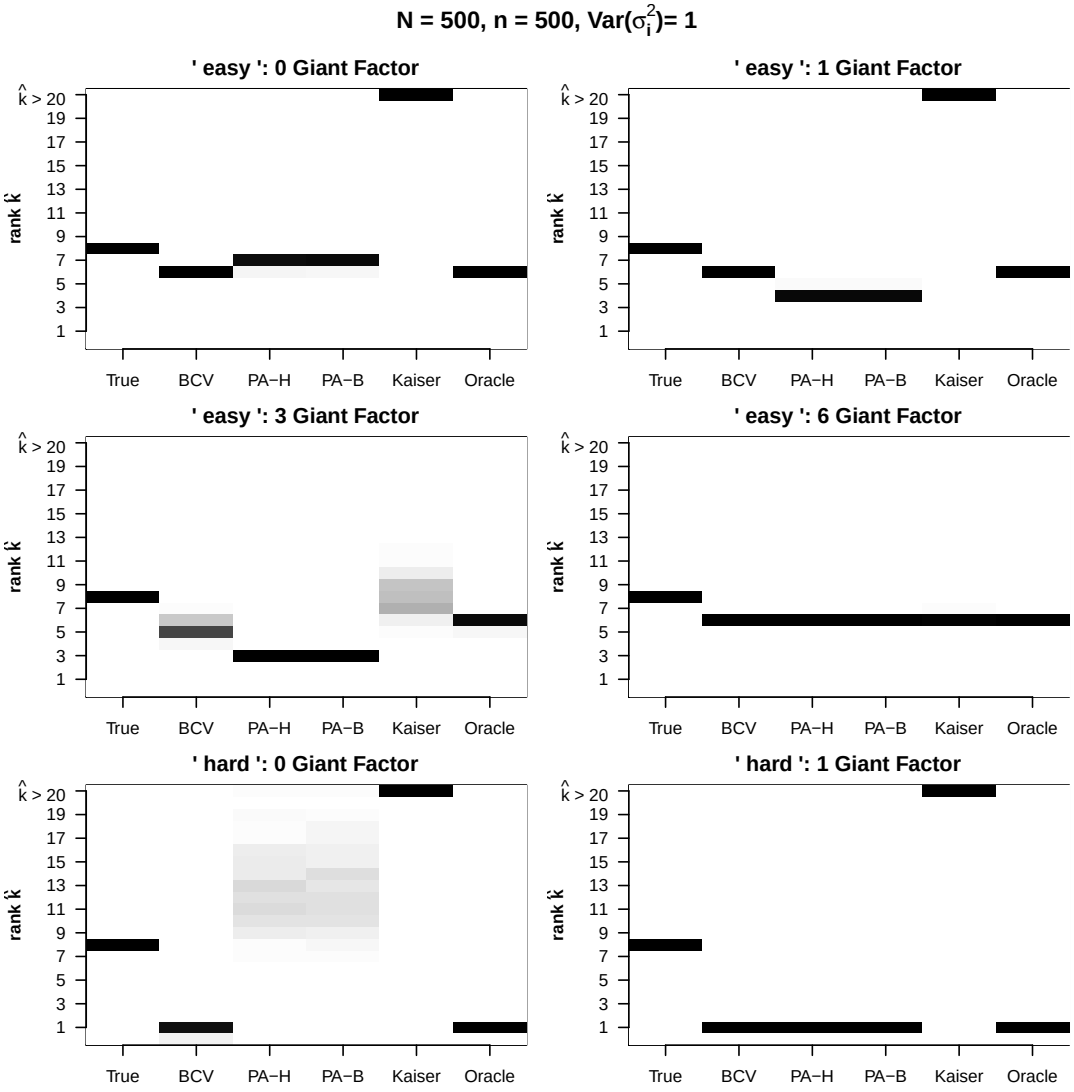

Figure 2: The distribution of $\hat{k}$ for each factor strength case when the matrix size is 500×500. Each image depicts 100 simulations with counts plotted in grey scale (larger equals darker) "PA-H" is short for "PA-Horn" and "PA-B" is short for "PA-B&E". The true k is always $k_0 = 8$. The "Oracle" method corresponds to k^*_{ESA}

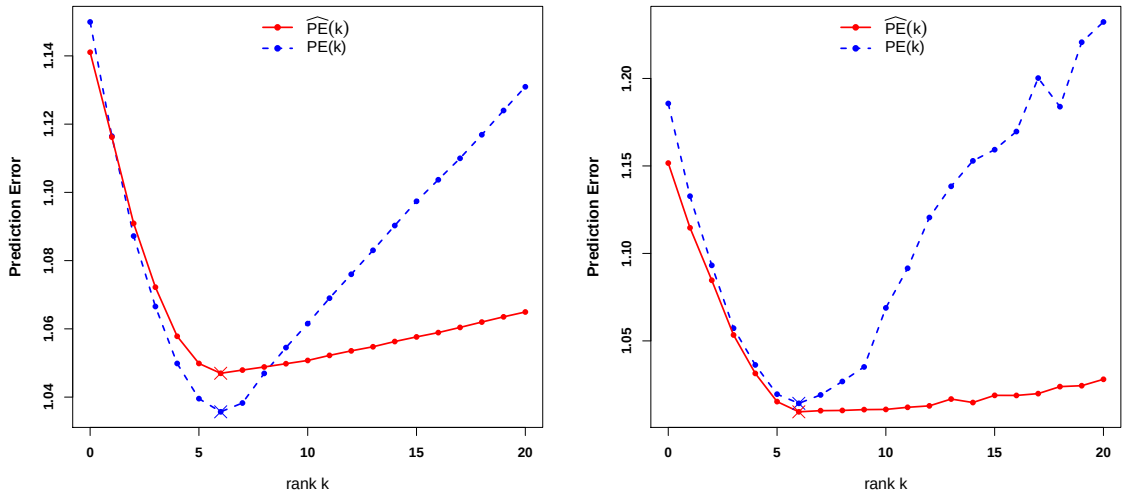

Figure 3: BCV prediction error $\widehat{\mathrm{PE}}(k)$ and true prediction error $\mathrm{PE}(k)$ vs k. The data are $Y \in \mathbb{R}^{500\times 500}$ (left) and $Y \in \mathbb{R}^{100\times 5000}$ (right). There are 8 factors corresponding to the easy scenario in Table 1 with no giant factors. The solid red line is the average over all held-out blocks. The crosses mark the positions of minimum $\widehat{\mathrm{PE}}(k)$ and $\mathrm{PE}(k)$ respectively.

7 Real Data Example

We investigate a real data example to show how our method works in practice. The observed matrix Y is 15×8192, where each row is a chemical element and each column represents a position on a 64×128 map of a meteorite. We thank Ray Browning for providing this data. Similar data are discussed in [29]. Each entry in Y is the amount of a chemical element at a grid point. The task is to analyze the distribution patterns of the chemical elements on that meteorite, helping us to further understand the composition.

A factor structure seems reasonable for the elements as various compounds are distributed over the map. The amounts of some elements such as Iron and Calcium are on a much larger scale than some other elements like Sodium and Potassium, and so it is necessary to assume a heteroscedastic noise model as (1). We center the data for each element before applying our method.

BCV choose $k = 4$ factors, while PA chooses $k = 3$. Figure 4 plots the BCV error for each rank, showing that among the selected factors, the first two factors

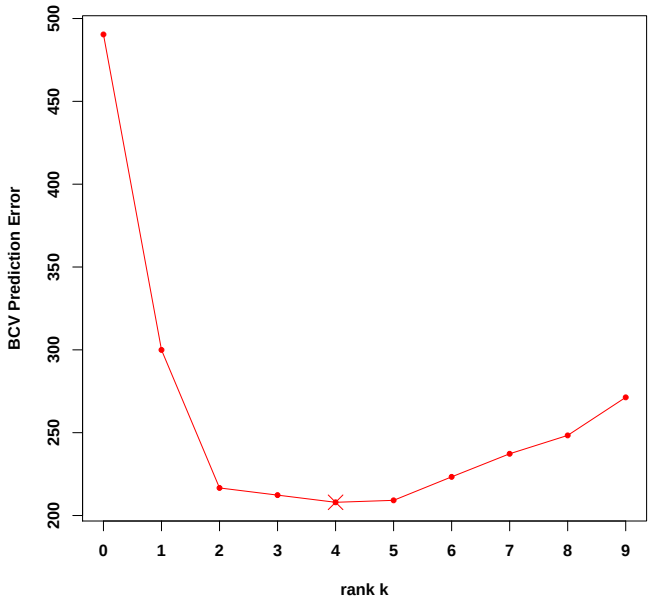

Figure 4: BCV prediction error for the meteorite. The BCV partitions have been repeated 200 times. The solid red line is the average over all held-out blocks, with the cross marking the minimum BCV error.

are much more influential than the last two. The first column of Figure 5 plots the four factors ESA has found at their positions. They represents four clearly different patterns.

As a comparison, we also apply a straight SVD on the centered data with and without standardization to analyze the hidden structure. The second and third columns of Figure 5 shows the first five factors of the locations that SVD finds for the original and scaled data respectively. If we do not scale the data, then the factor (F5) showing the concentration of Sulfur on some specific locations strangely comes after the factor (F4) which has no apparent pattern; F5 would have been neglected in a model of three or four factors as BCV or PA suggest.

Paque et al.[29] investigate this sort of data by clustering the pixels based on the values of the first two factors of a factor analysis. We apply such a clustering in Figure 6. Column (a) shows the resulting clusters. The factors found by ESA clearly divide the locations into five clusters, while the factors found by an SVD on the original data blur the boundary between clusters 1 and 5. An SVD on normalized

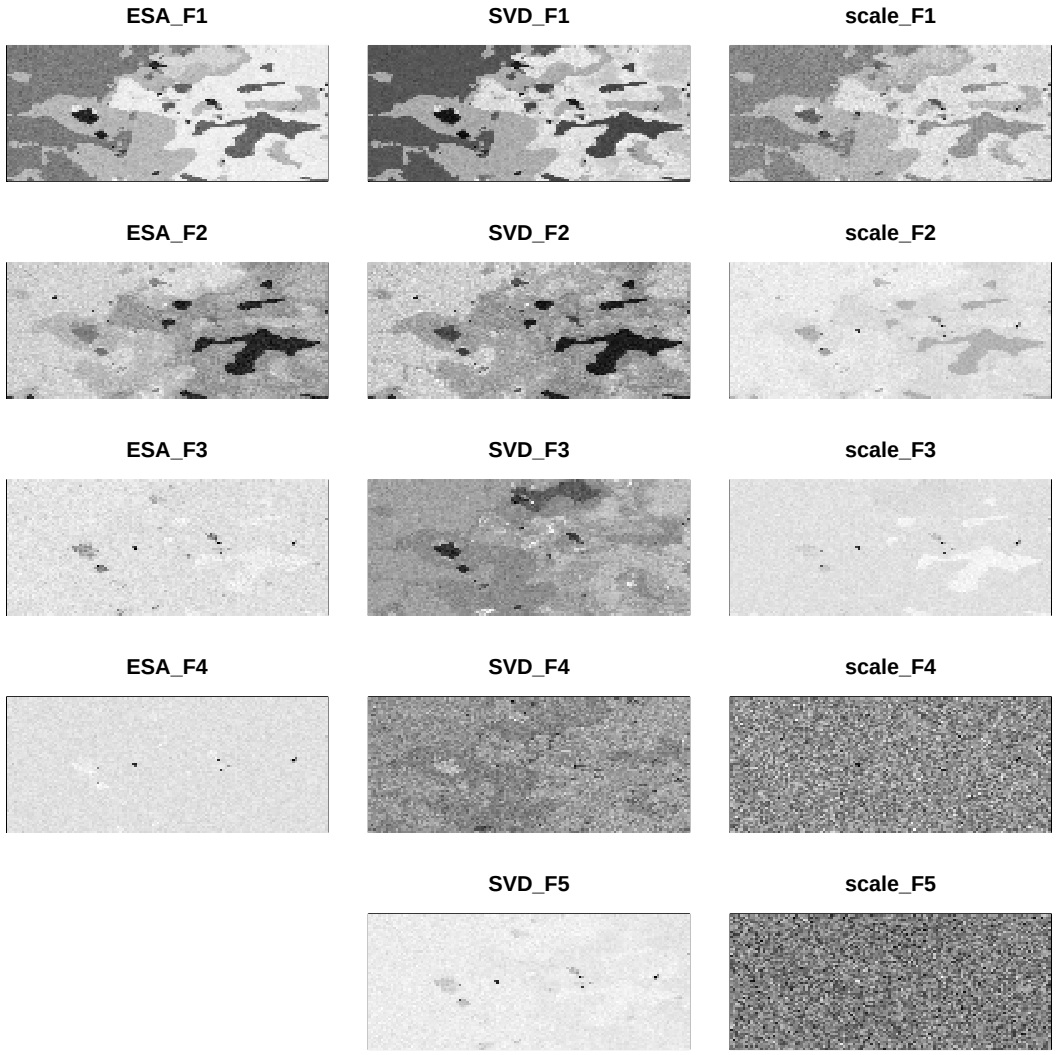

Figure 5: Distribution patterns of the estimated factors. The first column has the four factors found by ESA. The second column has the top five factors found by applying SVD on the unscaled data. The third column has the top five factors found by applying SVD on scaled data in which each element has been standardized. The values are plotted in grey scale, and a darker color indicates a higher value.

data (third plot in column (a)) blurs together three of the clusters. Columns (b) and (c) of Figure 6 show the quality of clustering using k-means based on the first two plots of Column (a). Clusters, especially C1 and C5, have much clearer boundaries and are less noisy if we are using ESA factors than using SVD factors. A k-means clustering depends on the starting points. For the ESA data the clustering was stable. For SVD the smallest group C3 was sometimes merged into one of the other clusters; we chose a clustering for SVD that preserved C3.

In this data the ESA based factor analysis found factors that, visually at least, seem better. They have better spatial coherence, and they provide better clusters than the SVD approaches do. For data of this type it would be reasonable to use spatial coherence of the latent variables to improve the fitted model. Here we have used those features instead as an informal confirmation that BCV is making a reasonable choice instead.

7.1 AGEMAP data

The meteorite data is the second of two real world data sets that we have tried BCV on. The first was the AGEMAP data used to study the LEAPP algorithm [36]. There, instead of a gold standard of a known signal matrix, the notion of ground truth is supplied by the idea that a better estimate of the signal in expression matrices for 16 different tissues should lead to greater overlap among the genes declared significant in those tissues. This is an indirect gold standard like the idea of positive controls in [14]. The LEAPP algorithm used parallel analysis as implemented in the SVA package of [23].

The SVA package implemented Buja and Eyuboglu's parallel analysis comparing eigenvalues of the unadjusted sample covariance matrix, which differs slightly from the original proposal that works on the unadjusted correlation matrix. Here adjustment refers to subtracting a diagonal term to account for noise. Working with adjusted matrices is generally considered less effective.

Placing BCV in LEAPP for the AGEMAP data yields a result similar to PA on the correlation matrix but is somewhat less effective than PA with the covariance matrix. All three are fairly close and all three gave better overlap than SVA did.

We do not understand why BCV failed to improve the overlap measure for the AGEMAP data. Here are some possibilities: We simulated Gaussian data using guidance from mostly Gaussian RMT, and the real data might not have been close enough to Gaussian. The noise covariance in AGEMAP might not have been nearly diagonal. There may not have been enough harmful factors in the AGEMAP data for the differences to be observed. LEAPP may be robust to missing weak factors.

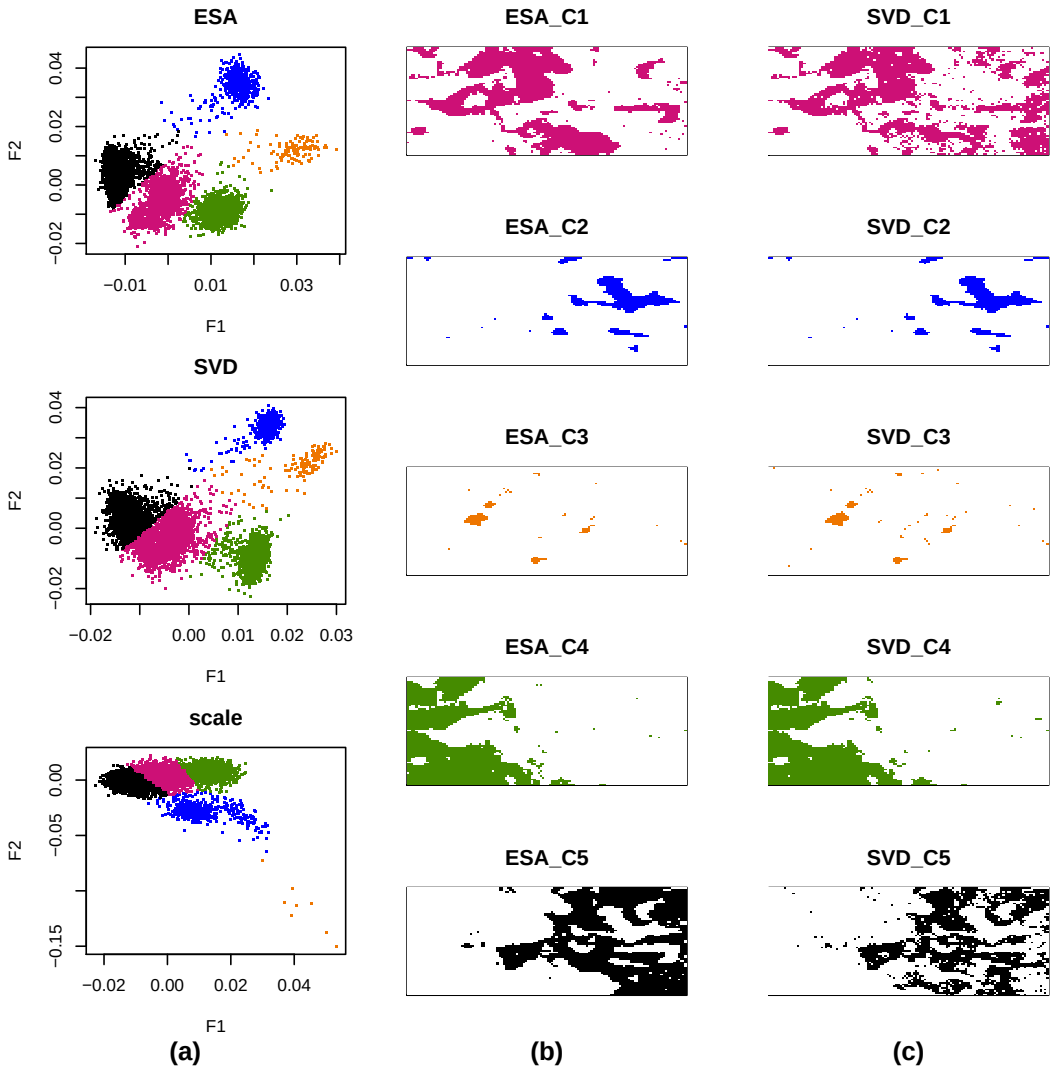

Figure 6: Plots of the first two factors and the location clusters. The three plots of column (a) are the scatter plots of pixels for the first two factors found by the three methods: ESA, SVD on the original data and SVD on normalized data. The coloring shows a k-means clustering result for 5 clusters. Column (b) has the five clustered regions based on the first two factors of ESA. Column (c) has the five clustered regions based on the first two factors of SVD on the original data after centering. The same color represents the same cluster.

Finally, there is no reason to expect that one method will be closer to an oracle on every data set.

8 Conclusion

In this paper, we have developed a bi-cross-validation algorithm to choose the number of factors in a heteroscedastic factor analysis and an early stopping alternation to estimate the model. Guided by random matrix theory, we have constructed a battery of test scenarios and found that stopping at three iterations is very effective. Using that early stopping rule we find that our bi-cross-validation proposal produces better recovery of the underlying signal matrix than the widely used method of parallel analysis and Kaiser's method.

Acknowledgments

This work was supported by the US National Science Foundation grant DMS-1407397.

References

[1] J. Bai and K. Li. Statistical analysis of factor models of high dimension. *The Annals of Statistics*, 40(1):436–465, 2012.

[2] J. Bai and S. Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002.

[3] M. S. Bartlett. Tests of significance in factor analysis. *British Journal of Statistical Psychology*, 3(2):77–85, 1950.

[4] M. S. Bartlett. A note on the multiplying factors for various χ^2 approximations. *Journal of the Royal Statistical Society, Series B*, pages 296–298, 1954.

[5] F. Benaych-Georges and R. R. Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, 2012.

[6] A. Buja and N. Eyuboglu. Remarks on parallel analysis. *Multivariate behavioral research*, 27(4):509–540, 1992.

[7] R. Caruana, S. Lawrence, and L. Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. *Advances in neural information processing systems*, pages 402–408, 2001.

[8] R. B. Cattell. The scree test for the number of factors. *Multivariate behavioral research*, 1(2):245–276, 1966.

[9] R. B. Cattell and S. Vogelmann. A comprehensive trial of the scree and KG criteria for determining the number of factors. *Multivariate Behavioral Research*, 12(3):289–325, 1977.

[10] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. *Annals of Statistics*, pages 1935–1967, 2012.

[11] E. F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417, 1970.

[12] H. E. Fleming. Equivalence of regularization and truncated iteration in the solution of ill-posed image reconstruction problems. *Linear Algebra and its applications*, 130:133–150, 1990.

[13] M. Forni and M. Lippi. *Aggregation and the microfoundations of dynamic macroeconomics*. Oxford University Press, 1997.

[14] J. A. Gagnon-Bartsch and T. P. Speed. Using control genes to correct for unwanted variation in microarray data. *Biostatistics*, 13(3):539–552, 2012.

[15] M. Gavish and D. L. Donoho. The optimal hard threshold for singular values is 4/sqrt (3). *arXiv preprint arXiv:1305.5870*, 2013.

[16] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning*. Springer, 2009.

[17] S. Hochreiter, D.-A. Clevert, and K. Obermayer. A new summarization method for Affymetrix probe level data. *Bioinformatics*, 22(8):943–949, 2006.

[18] J. L. Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185, 1965.

[19] R. Hubbard and S. J. Allen. An empirical comparison of alternative methods for principal component extraction. *Journal of Business Research*, 15(2):173–190, 1987.

[20] I. Jolliffe. *Principal component analysis*. New York: Springer-Verlag, 1986.

[21] H. F. Kaiser. The application of electronic computers to factor analysis. *Educational and psychological measurement*, 1960.

[22] S. Kritchman and B. Nadler. Determining the number of components in a factor model from limited noisy data. *Chemometrics and Intelligent Laboratory Systems*, 94(1):19–32, 2008.

[23] J. T. Leek and J. D. Storey. A general framework for multiple testing dependence. *Proceedings of the National Academy of Sciences*, 105(48):18718–18723, 2008.

[24] D. Love, D. Hallbauer, A. Amos, and R. Hranova. Factor analysis as a tool in groundwater quality management: two southern African case studies. *Physics and Chemistry of the Earth, Parts A/B/C*, 29(15):1135–1143, 2004.

[25] R. R. Nadakuditi. Optshrink: An algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage. *Information Theory, IEEE Transactions on*, 60(5):3002–3018, May 2014.

[26] R. R. Nadakuditi and A. Edelman. Sample eigenvalue based detection of high-dimensional signals in white noise using relatively few samples. *Signal Processing, IEEE Transactions on*, 56(7):2625–2638, 2008.

[27] B. Nadler. Finite sample approximation results for principal component analysis: A matrix perturbation approach. *The Annals of Statistics*, pages 2791–2817, 2008.

[28] A. B. Owen and P. O. Perry. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *The Annals of Applied Statistics*, 3(2):564–594, 06 2009.

[29] J. M. Paque, R. Browning, P. L. King, and P. Pianetta. Quantitative information from x-ray images of geological materials. *Proceedings of the XIIth International Congress for Electron Microscopy*, 2:244, 1990.

[30] D. Paul. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, 17(4):1617, 2007.

[31] P. R. Peres-Neto, D. A. Jackson, and K. M. Somers. How many principal components? Stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis*, 49(4):974–997, 2005.

[32] P. O. Perry. Cross-validation for unsupervised learning. *arXiv preprint arXiv:0909.3052*, 2009.

[33] A. L. Price, N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick, and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904–909, 2006.

[34] S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *The Journal of Machine Learning Research*, 5:941–973, 2004.

[35] C. Spearman. "General Intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, 1904.

[36] Y. Sun, N. R. Zhang, and A. B. Owen. Multiple hypothesis testing adjusted for latent variables, with an application to the AGEMAP gene expression data. *The Annals of Applied Statistics*, 6(4):1664–1688, 2012.

[37] W. F. Velicer. Determining the number of components from the matrix of partial correlations. *Psychometrika*, 41(3):321–327, 1976.

[38] W. F. Velicer, C. A. Eaton, and J. L. Fava. Construct explication through factor or component analysis: A review and evaluation of alternative procedures for determining the number of factors or components. In *Problems and solutions in human assessment*, pages 41–71. Springer, 2000.

[39] M. Wax and T. Kailath. Detection of signals by information theoretic criteria. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(2):387–392, 1985.

[40] Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

[41] T. Zhang and B. Yu. Boosting with early stopping: Convergence and consistency. *Annals of Statistics*, pages 1538–1579, 2005.

[42] W. R. Zwick and W. F. Velicer. Comparison of five rules for determining the number of components to retain. *Psychological bulletin*, 99(3):432, 1986.

Appendix

A.1: Simulation test cases

Our model is a low rank signal plus heteroscedastic noise. The formulation $Y = \sqrt{n}UDV^{\mathsf{T}} + E$ does not make it easy to take account of random matrix theory. We write our model as

$$Y = \Sigma^{1/2}(\sqrt{n}UDV^{\mathsf{T}} + E) \tag{23}$$

where $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_N^2)$ and $\sqrt{n}UDV^{\mathsf{T}}$ is the SVD for $\Sigma^{-1/2}X$. For constant $\sigma_i^2 = \sigma^2$ RMT can be used to choose the entries of $D = \mathrm{diag}(d_1, d_2, \cdots, d_{k_0})$ where $d_1 > d_2 > \cdots > d_{k_0} > 0$.

A straightforward implementation of (23) would have uniformly distributed U. In that case however the mean square signal per row would be simply proportional to the noise mean square per row. We think this would make the problem unrealistically easy: the relative sizes of the noise variances would be well estimated by corresponding sample variances within rows. Our simulation chooses a non-uniform U in order to decouple the mean square signal of the rows from the mean square noise in the rows. Below are the rules for generating the simulated data.

Generating the noise

Recall that the noise matrix is $\Sigma^{1/2}E$. The steps are as follows.

1. $E = (e_{ij})_{N\times n}$: here $e_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$.

2. $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_N^2)$: $\sigma_i^2 \overset{\text{iid}}{\sim} \mathrm{InvGamma}(\alpha, \beta)$. Therefore $\mathbb{E}(\sigma_i^2) = \beta/\alpha - 1$ and $\mathrm{Var}(\sigma_i^2) = \beta^2/(\alpha-1)^2(\alpha-2)$. Parameters α and β are chosen so that $\mathbb{E}(\sigma_i^2) = 1$. We consider two heteroscedastic noise cases: $\mathrm{Var}(\sigma_i^2) = 1$ and $\mathrm{Var}(\sigma_i^2) = 10$. We also include a homoscedastic case with all $\sigma_i^2 = 1$.

Generating the signal

The signal matrix is $X = \sqrt{n}\Sigma^{1/2}UDV^{\mathsf{T}}$, where Σ is the same matrix used to generate the noise. Entries in D specify the strength of signals of the reweighted matrix $\Sigma^{-1/2}X$. As we discussed in Section 3.3, for high-dimensional white noise models [32], there are two thresholds of signal strength for truncated SVD: a detection threshold

and an estimation threshold. From [32] and [15] we know that the detection threshold is $\mu_F = \sqrt{\gamma}$ and the estimation threshold is

$$\mu_F^* = \frac{1+\gamma}{2} + \sqrt{\left(\frac{1+\gamma}{2}\right)^2 + 3\gamma},$$

in the homoscedastic $\sigma = 1$ case. For general σ we multiply the thresholds by σ. In the heteroscedastic case we replace σ^2 by $\bar{\sigma}^2 = (1/N)\sum_{i=1}^N \sigma_i^2$. Recall that our four categories are:

a) Undetectable, $d_i^2 < \mu_F$,
b) Harmful, $\mu_F < d_i^2 < \mu_F^*$,
c) Useful, $\mu_F^* < d_i^2 = o(N)$, and
d) Giant, $d_i^2 \sim O(N)$.

The signal simulation is as follows.

1. We include the 6 scenarios from Table 1. For easy cases, $d_1^2 = 0.5\mu_F$, $d_2^2 = (\mu_F + \mu_F^*)/2$ and for $i = 3, 4, \ldots, 8$

$$d_i^2 = (i - 1.5)(\mu_F^* \mathbb{1}_{d_i \text{ is weak}} + N\mathbb{1}_{d_i \text{ is giant}}).$$

For hard cases, $d_1^2 = 0.5\mu_F$, $d_8^2 = 1.5(\mu_F^* \mathbb{1}_{d_8 \text{ is weak}} + N\mathbb{1}_{d_8 \text{ is giant}})$ and for $i = 2, 3, \ldots, 7$

$$d_i^2 = \frac{8-i}{7}\mu_F + \frac{i-1}{7}\mu_F^*.$$

2. U and V: First V is sampled uniformly from the Stiefel manifold $V_k(\mathbb{R}^n)$. See Appendix A.1.1 in [32] for a suitable algorithm. Then an intermediate matrix U^* is sampled uniformly from the Stiefel manifold $V_k(\mathbb{R}^N)$. Using the previously generated V and Σ we solve

$$\Sigma^{-1/2} U^* D V^\mathsf{T} = U \tilde{D} \tilde{V}^\mathsf{T}$$

for U. Now U is non-uniformly distributed on on the Stiefel manifold in such a way that rows of U with large L^2 norm are not necessarily those with large σ_i^2.

Data dimensions

We consider 5 different N/n ratios: $0.02, 0.2, 1, 5, 50$ and for each ratio consider a small matrix size and a larger matrix size, thus there are in total 10 (N, n) pairs. The specific sample sizes appear in Table 2. In total there are $6 \times 3 \times 5 \times 2 = 180$ scenarios. Each of them was simulated 100 times, for a total of 18,000 simulated data sets.

A.2: Early stopping

To study the effects of early stopping, we investigated the cases from Appendix A.1, varying the number k of factors and varying the number m of steps. In these simulations we know the true signal X and so we can measure the errors. We use the five measurements below to study the effectiveness of ESA with $m = 3$:

1. $\mathrm{Err}_X(m = 3)\big/\mathrm{Err}_X(m = m_{\mathrm{Opt}})$:

   this compares $m = 3$ to the optimal m defined in (14).

2. $\mathrm{Err}_X(m = 3)\big/\mathrm{Err}_X(m = 1)$:

   this measures the advantage of doing some iterations beyond standardization.

3. $\mathrm{Err}_X(m = 3)\big/\mathrm{Err}_X(m = 50)$:

   this measures the advantage of stopping early, using $m = 50$ as proxy for iteration to convergence.

4. $\mathrm{Err}_X(m = 3)\big/\mathrm{Err}_X(\mathrm{SVD})$:

   this compares ESA ($m = 3$) to the truncated SVD one would do for homoscedastic data.

5. $\mathrm{Err}_X(m = 3)\Big/\mathrm{Err}_X(\mathrm{oSVD})$:

   this compares ESA ($m = 3$) to the truncated SVD that an oracle which knew Σ could use on $\Sigma^{-1/2}Y$. It measures the relative inaccuracy in $\hat{X}$ arising from the inaccuracy of $\hat{\Sigma}$.

Table 4 summarizes the mean and standard deviation of each measurement over 6000 simulations each, for $\mathrm{Var}(\sigma_i^2) = 0$, 1 and 10. There are several conclusions we can make from Table 4. From row 1 we see that stopping at $m = 3$ was almost identical to stopping at the unknown optimal m in terms of the oracle estimating error, as the mean is nearly 1 and the standard deviation is negligible. From row 2, we see that taking $m = 3$ steps brought an improvement compared with truncated SVD on standardized data. Row 3 shows that taking $m = 3$ brought an improvement compared to using $m = 50$, our proxy for iterating to convergence. The latter is highly variable. Row 4 shows that using the truncated SVD is better than ESA with $m = 3$ when the noise is homoscedastic. But even a noise level as small as $\mathrm{Var}(\sigma_i^2) = \mathbb{E}(\sigma_i^2) = 1$ reverses the preference sharply. Row 5 shows that an oracle which knew Σ and used it to reduce the data to the homoscedastic case would gain only 4% over ESA with $m = 3$.

Measurements	White Noise	Heteroscedastic Noise	
	$\mathrm{Var}(\sigma_i^2) = 0$	$\mathrm{Var}(\sigma_i^2) = 1$	$\mathrm{Var}(\sigma_i^2) = 10$
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(m=m_{\mathrm{Opt}})}$	1.01 ± 0.01	1.00 ± 0.01	1.00 ± 0.01
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(m=1)}$	0.92 ± 0.11	0.89 ± 0.13	0.88 ± 0.14
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(m=50)}$	0.88 ± 0.20	0.87 ± 0.21	0.87 ± 0.22
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(\mathrm{SVD})}$	1.04 ± 0.09	0.84 ± 0.20	0.78 ± 0.23
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(\mathrm{rSVD})}$	1.04 ± 0.09	1.05 ± 0.13	1.05 ± 0.14

Table 4: ESA using five measurements. For each of $\mathrm{Var}(\sigma_i^2) = 0, 1$ and 10, the average for every measurement is the average over $10 \times 6 \times 100 = 6000$ simulations, and the standard deviation is the standard deviation of these 6000 simulations.

Table 5 gives the average value of each measurement over 100 replications for all of the simulations with mild heteroscedasticity ($\mathrm{Var}(\sigma_i^2) = 1$). The first panel confirms that $m = 3$ is broadly effective. The second panel shows that SVD on standardized data (i.e., $m = 1$) is very ineffective when γ is small. The problem is more severe at large sample sizes. The third panel shows a sharp disadvantage to $m = 50$ iterations when γ is small and the scenarios are easy. That problem is more severe at the smaller sample sizes. The fourth panel shows that ignoring heteroscedasticity causes the greatest losses at small γ and large sample sizes. Easy scenarios with few giant factors are most affected. The fifth panel shows where ESA with $m = 3$ does worst compared to an oracle that knows Σ. The greatest losses are on the easy scenarios with few giant factors and small γ.

It remains an interesting puzzle that heteroscedasticity is less of a problem when the aspect ratio is higher. In those settings there are more nuisance σ_i^2 to estimate.

A.3: further simulation results

Here we present more detailed simulation results for the comparisons among BCV, PA and Kaiser's method. All methods used the $m = 3$ steps found to be an effective stopping rule. These results are all for the mildly heteroscedastic case, $\mathrm{Var}(\sigma_i^2) = 1$.

	$\gamma = 0.02$		$\gamma = 0.2$		$\gamma = 1$		$\gamma = 5$		$\gamma = 50$	
	(20, 1000)	(100, 5000)	(20, 100)	(200, 1000)	(50, 50)	(500, 500)	(100, 20)	(1000, 200)	(1000, 20)	(5000, 100)
$\mathrm{Err}_X(m=3)\big/\mathrm{Err}_X(m=m_{\mathrm{Opt}})$										
easy-0	1.005	1.000	1.012	1.000	1.006	1.000	1.004	1.000	1.000	1.000
easy-1	1.025	1.003	1.021	1.001	1.007	1.000	1.004	1.000	1.000	1.000
easy-3	1.012	1.009	1.018	1.007	1.016	1.001	1.006	1.000	1.000	1.000
easy-6	1.000	1.000	1.004	1.000	1.004	1.000	1.004	1.000	1.000	1.000
hard-0	1.008	1.000	1.010	1.000	1.006	1.000	1.004	1.000	1.001	1.000
hard-1	1.002	1.000	1.005	1.000	1.002	1.000	1.002	1.000	1.000	1.000
$\mathrm{Err}_X(m=3)\big/\mathrm{Err}_X(m=1)$										
easy-0	0.861	0.932	0.895	0.954	0.919	0.977	0.972	0.987	0.995	0.998
easy-1	0.828	0.561	0.857	0.626	0.783	0.749	0.924	0.892	0.988	0.984
easy-3	0.813	0.687	0.853	0.601	0.766	0.630	0.901	0.848	0.983	0.977
easy-6	0.702	0.596	0.771	0.643	0.830	0.774	0.950	0.919	0.993	0.990
hard-0	0.986	0.994	0.997	0.996	0.998	0.998	1.000	0.999	0.999	1.000
hard-1	0.904	0.900	0.942	0.932	0.965	0.964	0.987	0.986	0.997	0.998
$\mathrm{Err}_X(m=3)\big/\mathrm{Err}_X(m=50)$										
easy-0	0.490	0.772	0.514	0.984	0.746	1.000	0.569	1.000	1.000	1.000
easy-1	0.561	0.779	0.544	0.987	0.708	1.000	0.617	1.000	1.000	1.000
easy-3	0.593	0.953	0.567	0.980	0.762	1.001	0.692	1.000	1.000	1.000
easy-6	0.339	0.761	0.376	0.989	0.804	1.000	0.520	1.000	1.000	1.000
hard-0	0.946	0.987	0.984	0.999	0.989	1.000	0.994	1.000	1.000	1.000
hard-1	0.818	1.000	0.850	1.000	0.981	1.000	0.998	1.000	1.000	1.000
$\mathrm{Err}_X(m=3)\big/\mathrm{Err}_X(\mathrm{SVD})$										
easy-0	0.621	0.345	0.748	0.366	0.726	0.468	0.887	0.726	0.977	0.967
easy-1	0.876	0.384	0.924	0.397	0.739	0.488	0.895	0.736	0.978	0.969
easy-3	0.958	0.788	0.982	0.656	0.803	0.600	0.912	0.790	0.984	0.975
easy-6	0.819	0.713	0.880	0.755	0.893	0.857	0.977	0.951	0.996	0.994
hard-0	0.935	0.905	0.976	0.923	0.974	0.938	0.984	0.963	0.992	0.992
hard-1	0.916	0.903	0.949	0.936	0.959	0.967	0.984	0.986	0.998	0.998
$\mathrm{Err}_X(m=3)\big/\mathrm{Err}_X(\mathrm{rSVD})$										
easy-0	1.033	0.994	1.070	0.997	1.032	1.000	1.026	1.001	1.003	1.000
easy-1	1.451	1.003	1.325	0.999	1.043	1.001	1.030	1.001	1.002	1.000
easy-3	1.461	1.473	1.370	1.214	1.080	1.004	1.029	1.001	1.003	1.000
easy-6	1.021	1.002	1.042	1.002	1.024	1.001	1.018	1.001	1.001	1.000
hard-0	1.017	1.001	1.022	1.002	1.015	1.002	1.012	1.001	1.002	1.000
hard-1	1.002	1.000	1.008	1.000	1.004	1.000	1.005	1.000	1.001	1.000

Table 5: Comparison of ESA results for various (N, n) pairs and number of giant factors in the scenarios with $\mathrm{Var}(\sigma_i^2) = 1$.

$\mathrm{Var}(\sigma_i^2) = 1$

Factor Type	Method	$\gamma = 0.02$				$\gamma = 0.2$				$\gamma = 1$			
		(20, 1000)		(100, 5000)		(20, 100)		(200, 1000)		(50, 50)		(500, 500)	
Easy - 0	True	0.70	8	0.51	8	0.43	8	0.39	8	0.33	8	0.41	8
	BCV	0.17	6	**0.06**	6	0.65	4	**0.00**	6	0.14	5	**0.00**	6
	PA-Horn	**0.01**	6	0.07	7	0.10	5	0.09	7	**0.05**	6	0.12	7
	PA-B&E	**0.01**	6	0.07	7	0.09	5	0.09	7	**0.05**	6	0.12	7
	Kaiser	**0.01**	6	5.71	20	**0.04**	6	3.78	> 20	1.17	13	3.56	> 20
	Oracle	–	6	–	6	–	6	–	6	–	6	–	6
Easy - 1	True	0.64	8	1.12	8	0.40	8	0.42	8	0.38	8	0.42	8
	BCV	0.23	5	**0.06**	6	0.47	4	**0.00**	6	**0.27**	4	**0.00**	6
	PA-Horn	0.28	3	0.11	5	0.88	2	0.27	4	1.12	2	0.29	4
	PA-B&E	0.28	3	0.11	5	0.86	2	0.26	4	1.07	3	0.29	4
	Kaiser	**0.16**	4	0.16	6	**0.16**	5	3.58	18	0.29	8	3.64	> 20
	Oracle	–	5	–	6	–	5	–	6	–	6	–	6
Easy - 3	True	0.52	8	1.24	8	0.35	8	1.01	8	0.41	8	0.54	8
	BCV	0.20	5	0.38	5	**0.22**	4	**0.18**	5	0.34	4	**0.06**	5
	PA-Horn	0.20	3	0.18	3	0.41	3	0.47	3	0.69	3	0.69	3
	PA-B&E	0.20	3	0.18	3	0.41	3	0.47	3	0.69	3	0.69	3
	Kaiser	**0.19**	3	**0.17**	3	0.37	3	0.30	3	**0.25**	4	0.54	8
	Oracle	–	5	–	4	–	5	–	5	–	6	–	6
Easy - 6	True	0.77	8	1.65	8	0.60	8	1.52	8	0.56	8	1.02	8
	BCV	0.05	6	**0.00**	6	0.10	6	**0.00**	6	**0.00**	6	**0.00**	6
	PA-Horn	**0.00**	6	**0.00**	6	1.01	6	**0.00**	6	0.04	6	**0.00**	6
	PA-B&E	**0.00**	6	**0.00**	6	1.04	6	**0.00**	6	**0.00**	6	**0.00**	6
	Kaiser	**0.00**	6	**0.00**	6	**0.00**	6	**0.00**	6	**0.00**	6	0.01	6
	Oracle	–	6	–	6	–	6	–	6	–	6	–	6
Hard - 0	True	1.52	8	0.89	8	0.81	8	1.23	8	0.92	8	1.36	8
	BCV	**0.07**	0	**0.04**	1	**0.03**	0	**0.02**	1	**0.03**	0	**0.00**	1
	PA-Horn	0.56	6	0.74	7	0.25	4	2.10	11	0.36	4	2.88	13
	PA-B&E	0.58	6	0.73	7	0.25	4	2.15	11	0.38	4	2.92	13
	Kaiser	1.32	8	6.18	> 20	0.91	8	5.16	> 20	2.40	18	5.16	> 20
	Oracle	–	1	–	1	–	1	–	1	–	1	–	1
Hard - 1	True	2.16	8	1.81	8	1.18	8	1.34	8	1.10	8	1.46	8
	BCV	**0.00**	1	**0.00**	1	0.01	1	**0.00**	1	**0.00**	1	**0.00**	1
	PA-Horn	0.05	1	0.07	2	**0.00**	1	**0.00**	1	**0.00**	1	**0.00**	1
	PA-B&E	0.05	1	0.07	2	**0.00**	1	**0.00**	1	**0.00**	1	**0.00**	1
	Kaiser	0.75	4	4.53	12	0.70	5	5.75	> 20	2.01	13	5.55	> 20
	Oracle	–	1	–	1	–	1	–	1	–	1	–	1

Table 6: Comparison of REE and $\hat{k}$ of rank selection methods for various (N, n) pairs, and scenarios. For the columns of each (N, n) pair, the first column is the REE and the second column is $\hat{k}$. The values are averages over 100 simulations.

$\mathrm{Var}(\sigma_i^2) = 1$

Factor Type	Method	$\gamma = 5$				$\gamma = 50$			
		(100, 20)		(1000, 200)		(1000, 20)		(5000, 100)	
Easy - 0	True	0.27	8	0.36	8	0.28	8	0.30	8
	BCV	0.53	4	**0.00**	6	0.06	5	**0.01**	6
	PA-Horn	**0.07**	5	0.11	7	**0.00**	6	0.10	7
	PA-B&E	**0.07**	5	0.11	7	**0.00**	6	0.10	7
	Kaiser	1.38	18	3.04	> 20	1.92	19	2.48	> 20
	Oracle	–	6	–	6	–	6	–	6
Easy - 1	True	**0.27**	8	0.36	8	0.28	8	0.30	8
	BCV	0.55	4	**0.00**	6	**0.07**	5	**0.02**	6
	PA-Horn	1.31	2	0.21	4	0.49	3	0.12	4
	PA-B&E	1.33	1	0.20	4	0.49	3	0.12	4
	Kaiser	1.02	14	3.06	> 20	1.92	19	2.49	> 20
	Oracle	–	6	–	6	–	6	–	6
Easy - 3	True	**0.29**	8	0.37	8	0.28	8	0.30	8
	BCV	0.32	4	**0.03**	5	**0.12**	4	**0.02**	5
	PA-Horn	0.57	3	0.50	3	0.35	3	0.34	3
	PA-B&E	0.57	3	0.50	3	0.35	3	0.34	3
	Kaiser	0.30	8	3.14	> 20	1.92	19	2.50	> 20
	Oracle	–	6	–	6	–	6	–	6
Easy - 6	True	0.33	8	0.41	8	0.29	8	0.30	8
	BCV	**0.05**	6	**0.00**	6	**0.01**	6	**0.00**	6
	PA-Horn	1.08	6	**0.00**	6	0.18	6	**0.00**	6
	PA-B&E	1.08	6	**0.00**	6	0.18	6	**0.00**	6
	Kaiser	**0.05**	6	1.34	12	1.95	19	2.53	> 20
	Oracle	–	6	–	6	–	6	–	6
Hard - 0	True	0.97	8	1.44	8	1.47	8	1.55	8
	BCV	**0.02**	0	**0.01**	1	**0.01**	0	**0.03**	1
	PA-Horn	0.29	3	2.25	10	0.78	5	1.23	7
	PA-B&E	0.32	4	2.25	10	0.78	5	1.22	7
	Kaiser	2.42	19	5.29	> 20	4.46	19	5.62	> 20
	Oracle	–	1	–	1	–	1	–	1
Hard - 1	True	1.09	8	1.52	8	1.49	8	1.57	8
	BCV	0.01	1	**0.00**	1	0.01	1	0.01	1
	PA-Horn	**0.00**	1	**0.00**	1	**0.00**	1	**0.00**	1
	PA-B&E	**0.00**	1	**0.00**	1	**0.00**	1	**0.00**	1
	Kaiser	2.57	19	5.55	> 20	4.51	19	5.68	> 20
	Oracle	–	1	–	1	–	1	–	1

Table 7: Like Table 6, but for larger γ.