Statistical Science
2016, Vol. 31, No. 1, 119–139
DOI: 10.1214/15-STS539

Bi-Cross-Validation for Factor Analysis

Art B. Owen and Jingshu Wang

Abstract. Factor analysis is over a century old, but it is still problematic to choose the number of factors for a given data set. We provide a systematic review of current methods and then introduce a method based on bi-cross-validation, using randomly held-out submatrices of the data to choose the optimal number of factors. We find it performs better than many existing methods especially when both the number of variables and the sample size are large and some of the factors are relatively weak. Our performance criterion is based on recovery of an underlying signal, equal to the product of the usual factor and loading matrices. 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 chose are designed using guidance from random matrix theory. In particular, we include factors which are asymptotically too small to detect, factors large enough to detect but not large enough to improve the estimate, and two classes of factors (weak and strong) large enough to be useful. We also find that a form of early stopping regularization improves the recovery of the signal matrix.

Key words and phrases: Parallel analysis, random matrix theory, scree plot, unwanted variation.

1. INTRODUCTION

Factor analysis is a core technology for handling large data matrices, with applications in signal processing [59, 25], bioinformatics [52, 48, 37, 56, 21], finance and econometrics [20, 6] and other areas [38, 26, 33]. In psychology, the factor model dates back at least to the paper of Spearman [55] in 1904. A basic factor analysis model assumes that the data matrix $Y \in \mathbb{R}^{N\times n}$ with n observations and N variables is represented as a matrix X of some low-rank k (the signal) plus independent heteroscedastic noise. The signal X in turn can be factored into an $N \times k$ matrix times a $k \times n$ matrix and this (nonunique) 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. In traditional factor analysis problems which have a small N but a relatively large n, there is no widely agreed best performing methods (see, e.g., [50]) and recommendations among them are based largely on simulation studies [13, 58]. Classical methods such as hypothesis testing based on likelihood ratios [36] or methods based on information theoretic criteria [59] assume homoscedastic noise while heteroscedastic noise is more common in applications. In addition, they are derived in an asymptotic regime with a growing number of observations and fixed number of variables and do not perform well on matrices where both dimensions are large. Special methods for big data matrices where both N and n are large have been proposed recently in the econometrics community [5, 42, 32, 2, 1]. They are derived in an asymptotic framework where the factor strength grows as N and n both tend to infinity. However, these methods may not work well on weaker factors and that is a potential flaw when the strong factors are already well known and we are trying to discover the weaker ones. The random matrix theory (RMT) literature by contrast focuses on weak factors,

Art B. Owen is Professor, Department of Statistics, Stanford University, Stanford, California 94305-4065, USA (e-mail: owen@stanford.edu). Jingshu Wang is Ph.D. candidate, Department of Statistics, Stanford University, Stanford, California 94305-4065, USA (e-mail: jingshuw@stanford.edu).

but their methods 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 factor analysis in many applications depends critically on the number of factors chosen [21, 29].

In this paper, we develop an approach to choosing the number of factors using bi-cross-validation (BCV) [45]. 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 RMT. 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 where both n and N go to infinity. To overcome identifiability problems, we assume that the factors are orthogonal to each other. Our factors may thus be linear combinations of some real world factors. The four factor levels are as follows: undetectable, harmful, helpful, and strong.

Strong factors are those that asymptotically explain a fixed percentage of variance in the matrix Y. They become easy to detect as the corresponding singular values go to infinity under the asymptotics, but their presence causes difficulties for some methods of choosing k when there are also weak factors. 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 a singular value decomposition (SVD) based method can not 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 because detecting them can lead to worse performance. If c is still larger, then we can not only detect the factor but including it in $\hat{X}$ also yields an improvement. We call those factors helpful. Strong factors are also helpful, but "helpful" by itself will refer to helpful weak factors. This taxonomy is based on homoscedastic Gaussian noise. A similar idea of the taxonomy also appeared in Onatski [43, 44]. In [43], he proposed the model with both strong and weak factors, and an "effective number" of factors which is the number of detectable factors in our taxonomy. In [44], there is a concept of optimal loss efficiency which is attained by estimating the number of useful factors.

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. It defines the boundaries in our four-level taxonomy of factor sizes. Section 4 describes our early stopping alternation (ESA) algorithm to estimate the low-rank signal matrix with a given target k for the number of factors. Section 5 introduces the BCV technique to determine the number of factors. Section 6 summarizes extensive simulation results. In those cases BCV is more reliably close to an oracle's performance than all the other methods compared, including parallel analysis (PA), several leading methods in the econometrics literature, and the information criteria based method [40] using RMT assuming white noise. Also, unlike other 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 bioinformatics problems we have worked on, it is usual to have $N > n$ or even $N \gg n$, but this is not assumed. In a factor model, Y can be decomposed into a low-rank signal matrix plus noise:

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

where the low-rank signal matrix $X \in \mathbb{R}^{N\times n}$ 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) entries with mean 0 and variance 1. Each variable has its own noise variance given by $\Sigma = \operatorname{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_N^2)$. The signal matrix X is a signal that we wish to recover despite the heteroscedastic noise.

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, as some factors may be too small to estimate. We should instead seek the relatively important ones, which are the factors that are strong enough to contribute most to the signals and be accurately estimated.

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. A typical factor analysis aims to estimate the individual factors L and R. To avoid identification problems, our goal is to recover X, seeking to minimize

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

This criterion was used for factor models in [44] and for truncated SVDs and nonnegative matrix factorizations in [45]. The estimate $\hat{X}$ can be factored into $\hat{L}$ and $\hat{R}$ using rotations for greater interpretability.

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 rank for M is

$$k_M^* = \underset{k}{\mathrm{argmin}}(\|\hat{X}^M(k) - X\|_F^2), \tag{3}$$

and the corresponding oracle estimate of X is

$$\hat{X}_{\mathrm{opt}}^M = \hat{X}^M(k_M^*). \tag{4}$$

If all the factors are strong enough, then for a good method M, we anticipate that k_M^* should equal the true number of factors k_0. With weak enough factors we will have $k_M^* < k_0$.

Our algorithm has two steps. First, we need to devise a method M to effectively estimate X given the oracle rank k_M^*. Then with such a method in hand, we need a means to estimate k_M^*. Section 4 describes our early stopping alternation (ESA) algorithm for estimating X at a given k, which has the best performance compared with other methods given their own oracle ranks. Then Section 5 describes our BCV for estimating $k_{\mathrm{ESA}}^{\star}$ for the ESA algorithm. First we describe previous methods and the relevant RMT that motivates our comparisons.

3. LITERATURE REVIEW AND FACTOR TAXONOMY

Here we review the most commonly used methods for choosing the number of factors. We begin with some classical methods in factor analysis which are typically based on a limit with $n \to \infty$ while N is fixed. Then we consider some recently developed methods from the econometrics community for large matrices with strong factors and methods. The third source of methods is based on RMT which emphasizes weak factors with noise of constant variance. We use the recent work in RMT to develop the four-level taxonomy of factor sizes that guides our simulations.

3.1 Classical Methods for Factor Analysis

The most widely used classical methods for determining the number of factors or principal components include the scree test [12, 13], sphericity tests based on likelihood ratio [8, 36], parallel analysis (PA) [27, 10], the minimum average partial test of [57] and information criteria based methods such as minimum description length (MDL) [59, 17]. Those methods are aimed at estimating the true number k 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, thus, $k_M^* = k_0$ asymptotically for a reasonable estimation method M.

Regarding classical methods, we should mention the conceptual difference between determining the number of principal components for principal component analysis (PCA) and determining the number of factors for factor analysis. Factor analysis has additive heteroscedastic noise that is not present in PCA. Though many of the above methods have been modified to be applied to both problems, theoretical guarantees were only derived for PCA assuming white and Gaussian noise. Many researchers [30, 10, 62, 58] have found out that those methods usually perform much better for estimating the principal components than for factor analysis. Some of them [62, 58] suggest that even for factor analysis, one should perform PCA first in the initial stage to determine the number of factors before estimating the factors. We adopt this suggestion in this paper later when comparing these methods in Section 6.

There is a large amount of evidence [62, 28, 58, 50] that PA is one of the most accurate of the above classical methods for determining the number of factors. Parallel analysis compares the observed eigenvalues of the correlation matrix to those obtained in a Monte Carlo simulation. The first factor is retained if and only if its associated eigenvalue is larger than the 95th percentile of simulated first eigenvalues. For $k \geq 2$, the kth factor is retained when the first $k-1$ factors were retained and the observed kth eigenvalue is larger than the 95th percentile of simulated kth factors. The permutation version of PA was introduced by [10]. There the eigenvalues are simulated by applying independent uniform random permutations to each of the variables stored in Y. The earlier method of Horn [27] resamples from a Gaussian distribution. Parallel analysis has been used recently in bioinformatics [37, 56]. Though there exist no theoretical results to guarantee the accuracy of PA, it performs very well in practice.

3.2 Methods for Large Matrices and Strong Factors

This collection of methods is designed for an asymptotic regime where both $n, N \to \infty$ while k is fixed. For strong factors, it is usually assumed that $RR^T/n \to \Sigma_R$ and $L^T L/N \to \Sigma_L$ for some $k_0 \times k_0$ positive definite matrices Σ_R and Σ_L. In that case, the singular values of X are $O(\sqrt{nN})$. The methods are designed to estimate the true number of factors. In the above framework, the factors can be estimated consistently, and we should expect $k_M^* = k_0$. This was proved when M is the SVD by Onatski [44].

Some of the most popular methods to estimate the number of factors under the above scenario are based on the information criteria developed by Bai and Ng [5], with later improvements in [2]. It has been shown that these information criteria based rules are asymptotically consistent. Kapetanios [31, 32] proposed several methods assuming strong factors but making use of the RMT results on the sample eigenvalue distribution of pure white noise. However, the theoretical guarantees for his methods require homescedastic noise. Ahn and Horenstein [1] recently proposed two estimators for determining the number of factors by simply maximizing the ratio of two adjacent eigenvalues of the sample covariance matrix. The idea of maximizing such a ratio to estimate the number of factors can be also found in [34, 35]. Onatski [42] developed an estimator (ED) based on the difference of two adjacent eigenvalues of the sample covariance matrix, and has proved its consistency under a weaker assumption of the factor strength: instead of growing in the order of $O(\sqrt{N})$, the singular values of $X/\sqrt{n}$ are just required to diverge in probability as $N \to \infty$. For econometrics applications, there are more methods to estimate the number of factors [19, 3, 23] for dynamic factor models. These models are designed for a time series structure on the factors. Such dependency models are beyond the scope of this paper.

3.3 Methods for Large Matrices and Weak Factors

Here we review methods to estimate the number of weak factors in white noise, based on results in RMT. In this asymptotic regime, n and N diverge to infinity, while k is fixed and the singular values of X are $O(1)$. The model is commonly framed as

$$Y = \sqrt{n}UDV^{\mathsf{T}} + \sigma E, \tag{5}$$

where $\sqrt{n}UDV^{\mathsf{T}}$ is the SVD of X, so that $U \in \mathbb{R}^{N\times k_0}$ and $V \in \mathbb{R}^{N\times k_0}$ satisfy $U^{\mathsf{T}}U = V^{\mathsf{T}}V = I_{k_0\times k_0}$. The matrix $D = \operatorname{diag}(d_1, d_2, \dots, d_{k_0})$ defines the strength of each signal. Asymptotically, $d_i^2 \to u_i$ for some constants u_i. The noise matrix $E \in \mathbb{R}^{N\times n}$ is usually taken to have IID entries with mean 0, variance 1 and finite fourth moment [7].

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 $n, N \to \infty$ and $N/n \to \gamma$, there is a well-known phase transition for signal detection. If $u_i^2 < \sigma^2\sqrt{\gamma}$, then the corresponding factor is asymptotically not detectable using SVD-based methods, while if $u_i^2 > \sigma^2\sqrt{\gamma}$, the factor can be detected. See [49, 9, 51] for statements of this result. Simulations [40, 22] have also confirmed this result.

A principled way to select the rank is to estimate the number of factors with u_i above the asymptotic detection threshold $\sigma^2\sqrt{\gamma}$. Nadakuditi and Edelman [40] used an information criteria based method modified from the classical MDL estimator [59]. Kritchman and Nadler [33] developed an algorithm based on a sequence of hypothesis tests which are connected with the Roy's classical largest root test [54] to check for sphericity of a covariance matrix. Both methods will consistently estimate the number of detectable factors under weak factor asymptotics. Similar to [33, 15] provides a sequential hypotheses testing method which is not based on asymptotics.

Neither the true rank nor the number of detectable factors will necessarily optimize our criterion (3). 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 [45] propose a BCV algorithm to choose k for the truncated SVD, motivated by the loss (2). Perry's work [51] on BCV identifies a higher threshold for u_i 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. This second estimation threshold was later derived by [22] and by [44].

The above results are only valid in the white noise model (5), which is much simpler than the heteroscedastic model (1). For more general noise covariance structures, there are several recent theoretical results, but none of them solve our problem. For example, Nadler [41] considered a general spiked covariance model with the eigenvalues corresponding to the noise in the population covariance matrix converging to some limiting distribution. However, our heteroscedastic model (1) is not directly related to a spiked covariance model. Nadakuditi [39] 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 he did not show how to estimate the rank. Onatski [43] considered noise whose covariance structure can be represented by a Kronecker product, which includes the heteroscedastic noise case. However, his theory depends on the strong assumption that the factors and the noise covariance have the same eigenvectors. He suggested using the ED estimator mentioned in Section 3.2 to estimate the number of weak detectable factors, which works well in his simulations.

3.4 Factor Categories and Test Cases

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

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

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 nonuniform 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 3.3 and under the asymptotics that $n, p \to \infty$, we may place each factor into a category depending on the size of d_i^2. The categories are as follows:

1. Undetectable: d_i^2 is below the detection threshold, thus the factor is asymptotically undetectable by SVD based methods.
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. Strong: d_i^2 grows proportionally to N. The factor sum of squares is then proportional to the noise level.

Undetectable factors essentially add to the noise level. Asymptotically, no method based on sample eigenvalues 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 a factor number estimator to reduce the loss (2). 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 strong 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. Given an estimation method, the total number of useful weak factors and strong factors is the same as the oracle rank.

Real data often include factors that fit the asymptotic strong factor category. In a matrix of dimensional measurements on animals, there is likely to be a strong factor for the overall size of those animals. In educational testing data where n students each answer N questions, there is very often a strong 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 strong factors should be easy to detect, they can cause severe difficulties for some algorithms as illustrated in Section 6. Useful weak factors may appear negligible in comparison to the strong ones. In each of these examples one can envision settings where the strongest factors are obvious and uninteresting, while the weak factors have useful insights.

Strong factors resemble the giant components commonly found in networks [16]. Network theory has several well understood mechanisms which lead to giant components. A mechanism for strong versus weak

TABLE 1
Six factor strength scenarios considered in our simulations

	Scenario					
	1	**2**	**3**	**4**	**5**	**6**
# Undetectable	1	1	1	1	1	1
# Harmful	1	1	1	3	3	6
# Useful	6	4	3	1	3	1
# Strong	0	2	3	3	1	0

factors seems to be missing. Suppose that one keeps adding measurements, increasing N, and perhaps doing so by adjoining additional features that are less and less important to one's primary scientific goals. A factor that strongly predicts the first few variables but is only weakly related to subsequent ones might become a weak factor in such a limit. A factor related to all of the variables we add would ordinarily be a strong one.

In the following sections we compare methods using the six testing scenarios described in Table 1. They have been customized based on our goals and our understanding of the problem. All of these cases have eight nonzero factors of which one is undetectable. We anticipate that the number of harmful factors is an important variable, and so it generally increases with scenario number, ranging from 1 to 6. The remaining factors are split between strong and merely useful. By including several scenarios with equal numbers of harmful factors, we can vary the ratio of strong to useful factors at high and low numbers of harmful factors.

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. The goal in this section is to find a method that has good performance when given its oracle rank. Assuming Gaussian noise, we get the log-likehihood function:

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

If Σ were known, it would be straightforward to estimate X using an SVD, 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 of Σ using the sample variance:

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

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

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

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

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

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 of the σ_i decreases to zero. Such a degenerate problem is similar to the degenerate problem 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 (the variance of one component goes to 0).

It is not straightforward to prevent σ_i from approaching 0. Imposing a bound $\sigma_i \ge \varepsilon > 0$ 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 toward some zero estimates.

A second, related approach is to employ a penalized likelihood

$$\begin{aligned} & L_{\text{reg}}(Y, \lambda, \hat{X}, \hat{\Sigma}) \\ (12) \qquad & = -n \log\det \hat{\Sigma} + \operatorname{tr}\big[\Sigma^{-1}(Y - \hat{X})(Y - \hat{X})^{\mathsf{T}}\big] \\ & \quad + \lambda P(\hat{\Sigma}), \end{aligned}$$

where P penalizes small components σ_i. This approach has two challenges. It is hard to select a penalty P that is strong enough to ensure boundedness of the likelihood, without introducing too much bias. 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 surprisingly 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

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

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

$$(14) \qquad \begin{aligned} m_{\text{opt}} &= \operatorname*{argmin}_m \operatorname{Err}_X(m) \\ &= \operatorname*{argmin}_m \min_k \|\hat{X}^m(k) - X\|_F^2. \end{aligned}$$

We have found that $m = 3$ is very nearly optimal in almost all cases. We find that $\operatorname{Err}_X(3)/\operatorname{Err}_X(m_{\text{opt}})$ is on average less than 1.01, with a standard deviation of 0.01 (see Appendix). Using $m = 3$ steps with the best k is nearly as good as using the best possible combination of m and k. We have tested early stopping on other data sizes, factor strengths and noise distributions, and find that $m = 3$ is a robust choice. Early stopping is also much faster than iterating until a convergence criterion has been met.

In the Appendix, we compare ESA to other methods for estimating X, including SVD, PCA (SVD after data standardization) and the quasi maximum-likelihood method (QMLE). The QMLE is derived by a classical factor analysis approach and it gives consistent estimation for strong factors and large data sets [4]. For the heteroscedastic noise cases and given the oracle rank of each method, ESA performs better than SVD and PCA in most cases. It also performs better than QMLE on average and when the aspect ratio N/n is not too small. Comparing ESA with an oracle SVD method that knows the noise variance, we find that they have comparable performance.

Given the above findings, we turn our attention to estimating the oracle k for ESA in Section 5.

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 [60, 61, 24, 11]. An equivalence between early stopping and adding a penalty term has been demonstrated in some settings [18, 53].

REMARK 4.2. ESA starting from (9) with $m = 1$ is equivalent to PCA. 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 $m = 1$.

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 [51].

5.1 Bi-Cross-Validation to Estimate k^*_{ESA}

We want k to minimize the squared estimating error (3) in $\hat{X}$. We adapt the BCV technique of Owen and Perry [45] 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:

$$\begin{aligned} &\begin{pmatrix} Y_{00} & Y_{01} \\ Y_{10} & Y_{11} \end{pmatrix} \\ &\quad = \begin{pmatrix} X_{00} & X_{01} \\ X_{10} & X_{11} \end{pmatrix} + \begin{pmatrix} \Sigma_0 & 0 \\ 0 & \Sigma_1 \end{pmatrix}^{1/2} \begin{pmatrix} E_{00} & E_{01} \\ E_{10} & E_{11} \end{pmatrix} \\ &\quad = \begin{pmatrix} L_0R_0 & L_0R_1 \\ L_1R_0 & L_1R_1 \end{pmatrix} + \begin{pmatrix} \Sigma_0^{1/2}E_{00} & \Sigma_0^{1/2}E_{01} \\ \Sigma_1^{1/2}E_{10} & \Sigma_1^{1/2}E_{11} \end{pmatrix}, \end{aligned}$$

where $L = \binom{L_0}{L_1}$ and $R = (R_0 \;\; R_1)$ are decompositions of the factors.

The held-in block

$$Y_{11} = X_{11} + \Sigma_1^{1/2}E_{11} = L_1R_1 + \Sigma_1^{1/2}E_{11}$$

has the low-rank 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 $k < \operatorname{rank}(Y_{11})$ 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 $Y_{01}^{\mathsf{T}} = \hat{R}_1^{\mathsf{T}}L_0^{\mathsf{T}} + E_{01}^{\mathsf{T}}\Sigma_0^{1/2}$, and estimate R_0 by solving n_0 weighted linear regression models $Y_{10} = \hat{L}_1R_0 + \hat{\Sigma}_1^{1/2}E_{10}$. These least square solutions are

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

and

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

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 [45]. □

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*

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

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

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

The third equality follows from Theorem 5.1. □

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

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

Notice that as the partition is random, we have

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

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

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

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

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

where $\hat{\Sigma}_1(k) = \mathrm{diag}(\hat{\sigma}^{(k)}_{1,1}, \hat{\sigma}^{(k)}_{2,1}, \dots, \hat{\sigma}^{(k)}_{N_1,1})$, 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 [45] 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, e.g., [14]). Thus, we only investigate cases where the signal 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

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

and then the oracle rank is $k^*_{\mathrm{ESA}} = \mathrm{argmin}_k\, \mathrm{PE}(k)$.

Ideally, we would like $\widehat{\mathrm{PE}}(k)$ to 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 [51] 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 $N/n \to \gamma \in (0,\infty)$ 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. Owen and Perry [45] found that the BCV curve under repeated subsampling was remarkably stable for large matrices, and then 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 $n_0 \ge 1$ and $N_0 \ge 1$. For instance, with $\gamma = 1$ as $\rho \approx 22\%$, we hold out roughly half the rows and columns of the data.

6. SIMULATION RESULTS

We use simulation scenarios described in Section 3.4 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.

Let $m = \min(n, N)$ and the singular values of the data matrix Y be $\sqrt{n}\lambda_1, \sqrt{n}\lambda_2, \dots, \sqrt{n}\lambda_m$ in nonincreasing order. We compare BCV with 5 representative methods reviewed in Section 3. They are as follows:

1. PA: the permutation version of the parallel analysis [10]. The Gaussian version of [27] has a nearly identical performance in our test cases.
2. ED: the eigenvalue difference method [42] which estimates the number of factors as

$$\hat{k} = \max\{i \le k_{\max} : \lambda_i^2 - \lambda_{i+1}^2 \ge \delta\},$$

where asymptotically $k_{\max}$ should be a slowly increasing function of n, and δ is calculated via a calibration method described in [42]. If $\{i \le k_{\max} : \lambda_i^2 - \lambda_{i+1}^2 \le \delta\}$ is empty, then we take $\hat{k} = 0$.

3. ER: the eigenvalue ratio method [1] which is the maximizer of sequential eigenvalue ratios

$$\hat{k} = \underset{0 \le i \le k_{\max}}{\mathrm{argmax}} \frac{\lambda_i^2}{\lambda_{i+1}^2},$$

where $\lambda_0^2 = \sum_{i=1}^m \lambda_i^2/\log(m)$. Also, $k_{\max}$ is suggested to be determined as $\min(|\{j \geq 1 : \lambda_j^2 \geq \sum_{i=1}^m \lambda_i^2/m\}|, 0.1m)$.

4. IC1: one of the rules based on information criteria developed in [5]. It is the k value that minimizes the criterion function

$$\mathrm{IC}_1(k) = \log(V(k)) + k\left(\frac{n+N}{nN}\right)\log\left(\frac{nN}{n+N}\right),$$

where $V(k) = \|Y - \hat{Y}(k)\|_F^2/(nN)$.

5. NE: Nadakuditi and Edelman's information criteria based estimator [40] which aims to estimate the number of weak factors in the white noise model. Set

$$t_i = N\left[(N-i)\frac{\sum_{j=i+1}^N \lambda_j^4}{(\sum_{j=i+1}^N \lambda_j^2)^2} - \left(1 + \frac{N}{n}\right)\right] - \frac{N}{n},$$

and then choose

$$\hat{k} = \operatorname*{argmin}_{0 \leq i < \min(N,n)} \left[\frac{1}{2}\left(\frac{n}{N}\right)^2 t_i^2 + 2(i+1)\right].$$

Of these methods, ER and IC1 are designed for models with strong factors only. ED does not require strong factors to work. NE has theoretical guarantees for estimating the number of detectable weak factors in the white noise model. Finally, PA was designed and tested under the small N and large n scenarios. We want to compare the finite sized data set performance of these methods in settings with both strong and weak factors. In applications one cannot be sure that only the desired factor strengths are present. In an earlier version of the paper [46], we also compared with Kaiser's rule [30], which estimates the number of factors as the number of eigenvalues of the sample correlation matrix above 1. However, Kaiser's rule is likely to overestimate the number of factors and does not perform well. 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 (3). Methods that choose a value closer to k_{ESA}^* should attain a small error using ESA.

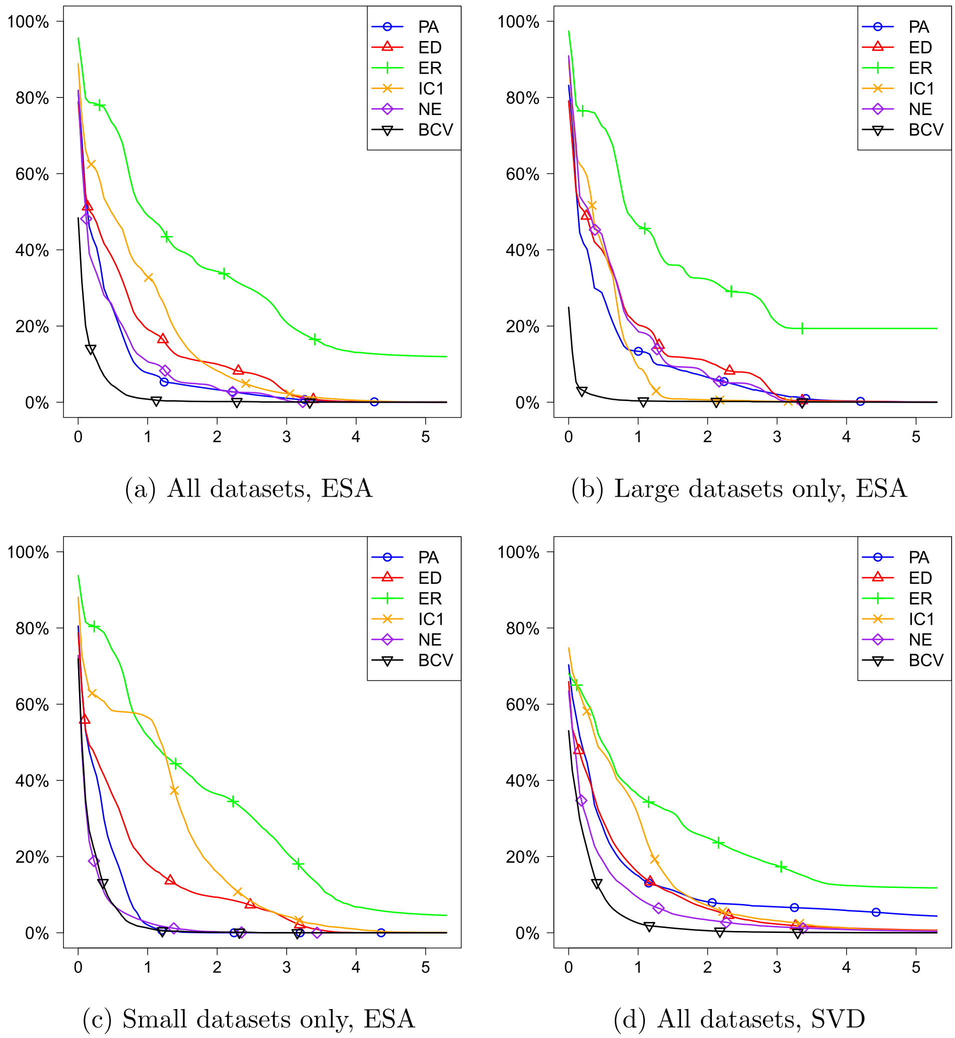
Figure 1 shows for different methods the proportion of simulations with REE above certain values for the mild heteroscedastic case $\mathrm{Var}(\sigma_i^2) = 1$. Figure 1(a) shows that BCV is overall best at recovering the signal matrix X. BCV is based on Perry's asymptotic Theorem 5.3. Figure 1(b) shows that BCV becomes far better than alternatives when we just compare the larger sample sizes from each aspect ratio. Figure 1(c) shows that at smaller sample sizes NE is competitive with BCV. The large data case is more important given the recent emphasis on large data problems.

Our goal is to find the best k for ESA, but the methods ED, ER. IC1 and NE are designed assuming that the SVD will be used to estimate the factors. To study them in the setting they were designed for, we include Figure 1(d), which calculates REE using SVD to estimate $\hat{X}(k)$ and compares with the oracle rank of SVD. For Figure 1(d), the BCV method also uses the SVD instead of ESA. Though the results in Table A.1 (Appendix) suggest that SVD is in general not recommended for heteroscedastic noise data, if one does use SVD, BCV is still the best method for choosing k to recover X.

The proportion of simulations with REE $= 0$ (matching the oracle's rank) for BCV was 51.6%, 75.1%, 28.1% and 47.0% in the four scenarios in Figure 1. BCV's percentage was always highest among the six methods we used. The fraction of REE $= 0$ sharply increases with sample size and is somewhat better for ESA than for SVD.

Table 2 briefly summarizes the REE values for all three noise variance cases. It shows the worst case REE over all the 10 matrix sizes and 6 factor strength scenarios. As the variance of σ_i^2 rises, it becomes more difficult to attain a small REE. BCV has substantially smaller worst case REE for heterscedastic noise than all other methods, but is slightly worse than NE for the white noise case. This is not surprising, as NE is designed for the white noise model.

To better understand the differences among the methods, we compare them directly in estimating the number of factors with the oracle. As an example, Figure 2 plots the distribution of $\hat{k}$ for all methods and all 6 cases, on 5000 × 100 data matrices with $\mathrm{Var}(\sigma_i^2) = 1$. The results of other cases are summarized in Tables A.3 and A.4 in the Appendix. In Figure 2, BCV closely tracks the oracle. For other methods, ED performs the best in estimating the oracle rank, though it is more variable and less accurate than BCV. ER is the most conservative method, trying to estimate at most the number of strong factors. IC1 also tries to estimate the number of strong factors, but is less conservative than ER. NE estimates some number between the number of strong factors and the number of useful (including strong) factors. PA has trouble identifying the useful weak factors when strong factors are present, and also has trouble rejecting the detectable but not useful factors in the hard case with no strong factor. This is due the fact that PA is using the sample correlation matrix which has a fixed sum of eigenvalues, thus, the magnitude of the each eigenvalue is influenced by every other one.



(a) All datasets, ESA

(b) Large datasets only, ESA

(c) Small datasets only, ESA

(d) All datasets, SVD

FIG. 1. *REE survival plots: the proportion of samples with REE exceeding the number on the horizontal axis. Figures* (a)–(c) *are for REE calculating using the method ESA. Figure* (a) *shows all* 6000 *samples. Figure* (b) *shows only the* 3000 *simulations of larger matrices of each aspect ratio. Figure* (c) *shows only the* 3000 *simulations of smaller matrices. For comparison, figure* (d) *is the REE plot for all samples calculating REE using the method SVD.*

Tables A.3 and A.4 in the Appendix provide more details of the simulation results for this mildly heteroscedastic case $\mathrm{Var}(\sigma_i^2) = 1$. We can see that some methods behave very differently for different sized data sets. For example, IC1 is very nonrobust and sharply overestimates the number of factors for small data sets, and ED will tend to estimate only the number of strong factors when the aspect ratio γ is small. Overall, BCV has the most robust and accurate performance in estimating k^*_{ESA} of the methods we investigated.

TABLE 2
Worst case REE values for each method of choosing k for white noise and two heteroscedastic noise settings

$\mathbf{Var}(\sigma_i^2)$	PA	ED	ER	IC1	NE	BCV
0	1.99	1.41	49.61	1.13	0.12	0.29
1	2.89	2.42	25.02	3.11	2.45	0.37
10	3.66	2.28	15.62	4.46	2.10	0.62

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 [47]. Each entry in Y is the amount of a chemical element at a grid point. The task is to analyze the distribution patterns of

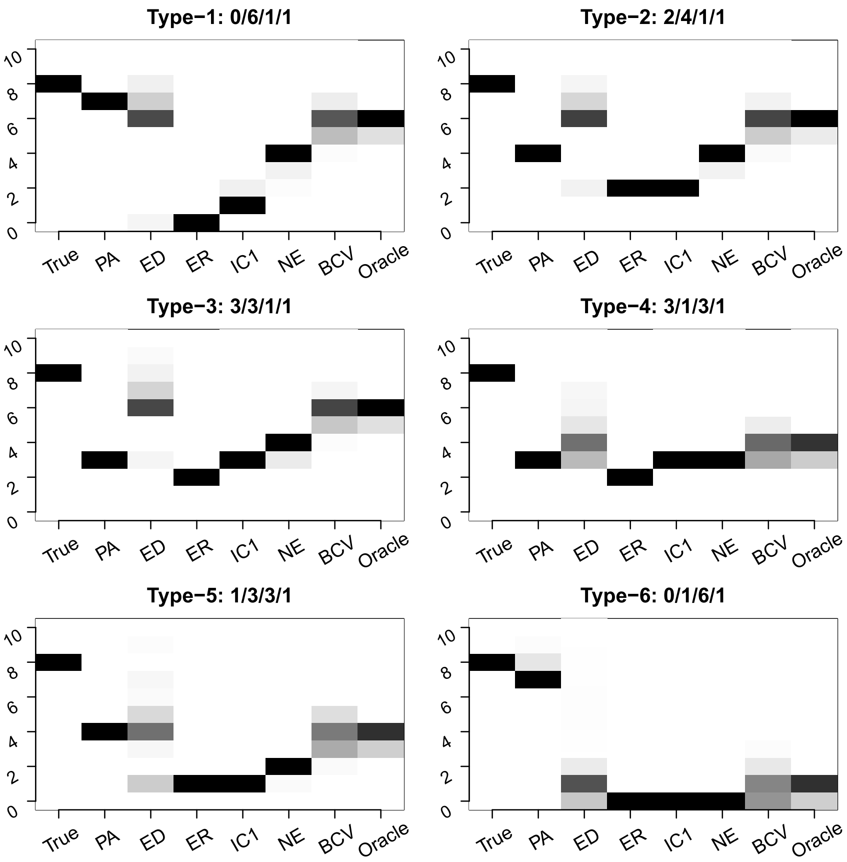


FIG. 2. *The distribution of $\hat{k}$ for each factor strength case when the matrix size is* 5000 × 100. *The y axis is $\hat{k}$. Each image depicts* 100 *simulations with counts plotted in grey scale* (*larger equals darker*). *For different scenarios*, *the factor strengths are listed as the number of "strong/useful/harmful/undetectable" factors in the title of each subplot. The true k is always $k_0 = 8$. The "Oracle" method corresponds to k^*_{ESA}.*

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 heteoroscedastic noise model as (1). We center the data for each element before applying our method.

BCV chooses $k = 4$ factors, while PA chooses $k = 3$. Figure 3 plots the BCV error for each rank, showing that, among the selected factors, the first two factors are much more influential than the last two. The first column of Figure 4 plots the four factors ESA has found at their positions. They represent 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

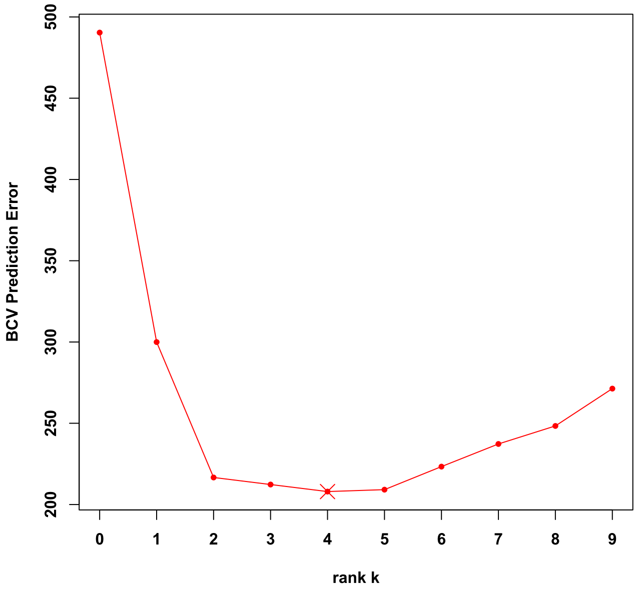


FIG. 3. *BCV prediction error for the meteorite. The BCV partitions have been repeated* 200 *times. The solid red line is the average over all held-out blocks, with the cross marking the minimum BCV error.*

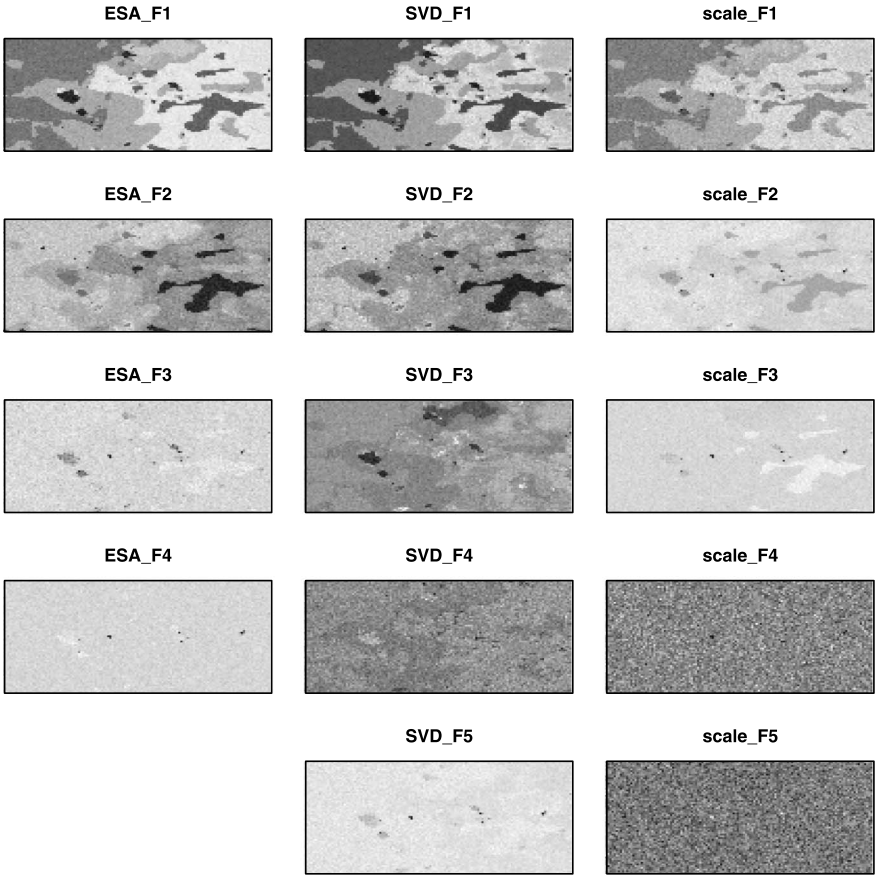


FIG. 4. *Distribution patterns of the estimated factors. The first column has the four factors found by ESA. The second column has the top five factors found by applying SVD on the unscaled data. The third column has the top five factors found by applying SVD on scaled data in which each element has been standardized. The values are plotted in grey scale, and a darker color indicates a higher value.*

columns of Figure 4 show 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. [47] 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 5. 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 data [third plot in column (a)] blurs together three of the clusters. Columns (b) and (c) of Figure 5 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 rather 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 spatial coherence as an informal confirmation that BCV is making a reasonable choice, which we could not do if we had exploited spatial coherence in estimating our factors.

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

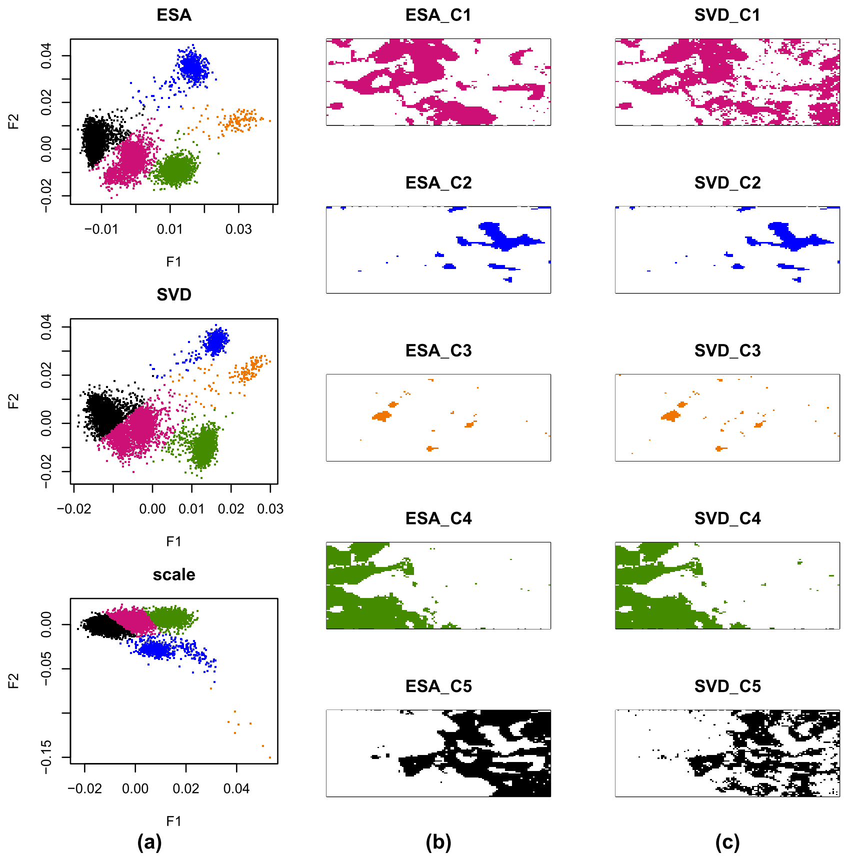


FIG. 5. *Plots of the first two factors and the location clusters. The three plots of column* (a) *are the scatter plots of pixels for the first two factors found by the three methods: ESA, SVD on the original data and SVD on normalized data. The coloring shows a k-means clustering result for* 5 *clusters. 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.*

the AGEMAP data used to study the LEAPP algorithm [56]. 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 [21]. The LEAPP algorithm used parallel analysis as implemented in the SVA package of [37].

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. 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 other widely used methods. It also improves markedly with sample size.

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

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

where $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_N^2)$ 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 $D = \mathrm{diag}(d_1, d_2, \dots, d_{k_0})$ where $d_1 > d_2 > \cdots > d_{k_0} > 0$.

A straightforward implementation of (A.1) 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 nonuniform 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 $e_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$.
2. $\Sigma = \mathrm{diag}(\sigma_1^2, \dots, \sigma_N^2)$: $\sigma_i^2 \overset{\text{i.i.d.}}{\sim} \mathrm{InvGamma}(\alpha, \beta)$. 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 3.4, for high-dimensional white noise models [51], there are two thresholds of signal strength for truncated SVD: a detection threshold and an estimation threshold. From [51] 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. Recall that based on the asymptotic thresholds, our four categories for a data set are roughly the following:

(a) Undetectable, $d_i^2 < \mu_F$,
(b) Harmful, $\mu_F < d_i^2 < \mu_F^*$,
(c) Useful, $\mu_F^* < d_i^2 = o(1)$, and
(d) Strong, $d_i^2 \sim O(N)$.

The signal simulation is as follows:

1. We include the 6 scenarios from Table 1. For the d_i^2 values we take, the strong factors take values at $1.5N$, $2.5N$, $3.5N, \dots$. The useful factors take values at $1.5\mu_F^\star$, $2.5\mu_F^\star$, $3.5\mu_F^\star, \dots$. The harmful factors take values at equally spaced interior points of the interval $[\mu_F, \mu_F^\star]$ and the undetectable factors take values at equally spaced interior points of the interval $[0, \mu_F]$.
2. U and V: First V is sampled uniformly from the Stiefel manifold $V_k(\mathbb{R}^n)$. See Appendix A.1.1 in [51] 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

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

for U. Now U is nonuniformly distributed 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 at the top of Table A.2. In total there are $6 \times 3 \times 5 \times 2 = 180$ scenarios. Each 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 six measurements below to study the effectiveness of ESA with $m = 3$:

1. $\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(m = m_{\mathrm{Opt}})$: this compares $m = 3$ to the optimal m defined in (14).
2. $\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(m = 1)$: this measures the advantage of ESA beyond PCA.

3. $\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(m = 50)$: this measures the advantage of stopping early, using $m = 50$ as proxy for iteration to convergence.

4. $\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(\mathrm{SVD})$: this compares ESA to the truncated SVD one would do for homoscedastic data.

5. $\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(\mathrm{QMLE})$: this compares ESA to the quasi maximum likelihood method, which is solved using the EM algorithm with principal component estimates as starting values.

6. $\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(\mathrm{oSVD})$: this compares ESA 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 $\hat{\Sigma}$.

For QMLE, R and Σ are estimated via maximizing the quasi-loglikelihood [4]:

$$\text{(A.2)}\qquad \begin{aligned} &-\frac{1}{2N}\log\det\left(RR^T + \Sigma\right) \\ &\quad -\frac{1}{2N}\mathrm{tr}\left\{\frac{YY^T}{n}\left[RR^T + \Sigma\right]^{-1}\right\}, \end{aligned}$$

then $\hat{L}$ is estimated via a generalized linear regression of Y on $\hat{R}$ with estimated variance $\hat{\Sigma}$ and $\hat{X} = \hat{R}\hat{L}$.

Table A.1 summarizes the mean and standard deviation of each measurement over 6000 simulations each, for $\mathrm{Var}(\sigma_i^2) = 0$, 1 and 10. Row 1 shows that ESA stopping at $m = 3$ steps 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. Row 2 indicates that taking $m = 3$ steps brought an improvement compared with PCA (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 to the local minimum of loss. The latter is highly variable. Row 4 shows that truncated SVD is better than ESA 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 ESA beats QMLE on average for the heteroscedastic noise case, though the latter has a theoretical guarantee for the strong factor scenario. Row 6 shows that an oracle which knew Σ and used it to reduce the data to the homoscedastic case would gain only 3% over ESA.

TABLE A.1
ESA using six 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*

	White noise	**Heteroscedastic noise**	
Measurements	$\mathbf{Var(\sigma_i^2) = 0}$	$\mathbf{Var(\sigma_i^2) = 1}$	$\mathbf{Var(\sigma_i^2) = 10}$
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(m=m_{\mathrm{Opt}})}$	1.01 ± 0.01	1.00 ± 0.01	1.00 ± 0.01
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(m=1)}$	0.93 ± 0.09	0.90 ± 0.11	0.89 ± 0.12
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(m=50)}$	0.87 ± 0.21	0.87 ± 0.21	0.87 ± 0.21
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(\mathrm{SVD})}$	1.03 ± 0.06	0.81 ± 0.20	0.75 ± 0.22
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(\mathrm{QMLE})}$	1.02 ± 0.05	0.95 ± 0.15	0.91 ± 0.19
$\frac{\mathrm{Err}_X(m=3)}{\mathrm{Err}_X(\mathrm{oSVD})}$	1.03 ± 0.06	1.03 ± 0.07	1.03 ± 0.08

Table A.2 gives the average value of each measurement over 100 replications for all of the simulations with mild heteroscedasticity [$\mathrm{Var}(\sigma_i^2) = 1$]. “Type-1” to “Type-6” correspond to the six cases of factor strengths listed in Table 1. The first panel confirms that $m = 3$ is broadly effective. The second panel shows that the problem of PCA is more severe at large sample sizes. The third panel shows in contrast that the disadvantage to $m = 50$ iterations is more severe at the smaller sample sizes. The fourth panel shows similar to the second panel that SVD causes greatest losses at large sample sizes. The fifth panel shows that ESA has great advantage over QMLE when the variable size is large, even at a low aspect ratio γ.

It remains an interesting puzzle that heteroscedasticity is less of a problem when the aspect ratio is higher for all the methods. In those settings there are actually more nuisance σ_i^2 to estimate. One explanation is that no matter what method used, the right factor R of size $r \times n$ can be accurately estimated if γ is large enough. Then the estimate of the left factor L is done via an ordinary linear regression of Y on R which is not affected by the heterscedastic noise. This explanation can also work for our observation that heteroscedasticity becomes a more severe problem for small γ, as given L, it is important to take into consideration different noise variance when estimating R.

A.3 Further Simulation Results

Here we present more detailed simulation results for the comparisons among the methods we compare. All methods used the $m = 3$ steps found to be an effective stopping rule.

ACKNOWLEDGMENTS

Supported in part by NSF Grants DMS-14-07397 and DMS-09-06056.

We thank the reviewers for comments that led to an improved paper.

TABLE A.2

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

Factor	$\gamma = 0.02$		$\gamma = 0.2$		$\gamma = 1$		$\gamma = 5$		$\gamma = 50$	
scenario	$(20, 1000)$	$(100, 5000)$	$(20, 100)$	$(200, 1000)$	$(50, 50)$	$(500, 500)$	$(100, 20)$	$(1000, 200)$	$(1000, 20)$	$(5000, 100)$
					$\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(m = m_{\mathrm{Opt}})$					
Type-1	1.011	1.000	1.011	1.000	1.004	1.000	1.003	1.000	1.000	1.000
Type-2	1.013	1.002	1.012	1.001	1.006	1.000	1.004	1.000	1.000	1.000
Type-3	1.016	1.006	1.014	1.005	1.010	1.000	1.002	1.000	1.000	1.000
Type-4	1.002	1.002	1.009	1.001	1.008	1.000	1.006	1.000	1.000	1.000
Type-5	1.008	1.001	1.011	1.001	1.007	1.000	1.006	1.000	1.000	1.000
Type-6	1.007	1.000	1.011	1.000	1.006	1.000	1.003	1.000	1.001	1.000
$\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(m = 1)$										
Type-1	0.900	0.936	0.913	0.957	0.924	0.977	0.967	0.987	0.995	0.998
Type-2	0.819	0.626	0.844	0.680	0.833	0.785	0.942	0.909	0.990	0.987
Type-3	0.827	0.613	0.840	0.616	0.801	0.739	0.925	0.887	0.987	0.984
Type-4	0.781	0.723	0.837	0.751	0.864	0.833	0.947	0.926	0.990	0.990
Type-5	0.854	0.789	0.904	0.834	0.911	0.899	0.962	0.956	0.993	0.994
Type-6	0.987	0.993	0.997	0.996	0.997	0.998	0.999	0.999	0.999	1.000
$\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(m = 50)$										
Type-1	0.441	0.802	0.473	0.985	0.759	1.000	0.590	1.000	1.000	1.000
Type-2	0.472	0.839	0.486	0.984	0.765	1.000	0.605	1.000	1.000	1.000
Type-3	0.501	0.918	0.463	0.994	0.751	1.000	0.626	1.000	1.000	1.000
Type-4	0.560	0.975	0.541	0.989	0.899	1.000	0.854	1.000	1.000	1.000
Type-5	0.604	0.907	0.671	0.992	0.821	1.000	0.842	1.000	1.000	1.000
Type-6	0.947	0.982	0.981	0.999	0.988	1.000	0.997	1.000	1.000	1.000
$\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(\mathrm{SVD})$										
Type-1	0.638	0.348	0.740	0.366	0.722	0.466	0.882	0.727	0.977	0.966
Type-2	0.785	0.450	0.829	0.451	0.749	0.525	0.898	0.754	0.980	0.972
Type-3	0.870	0.611	0.896	0.548	0.772	0.599	0.903	0.791	0.983	0.976
Type-4	0.872	0.810	0.923	0.809	0.893	0.872	0.960	0.942	0.991	0.990
Type-5	0.704	0.542	0.798	0.552	0.770	0.605	0.888	0.779	0.978	0.972
Type-6	0.935	0.906	0.972	0.925	0.971	0.943	0.985	0.966	0.993	0.991
$\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(\mathrm{QMLE})$										
Type-1	0.915	0.633	0.966	0.677	0.985	0.858	0.997	0.988	1.000	1.000
Type-2	1.104	0.672	1.058	0.725	1.000	0.863	0.999	0.989	1.000	1.000
Type-3	1.199	0.826	1.129	0.766	1.008	0.878	0.997	0.990	1.000	1.000
Type-4	1.035	0.991	1.033	0.954	1.005	0.973	1.002	0.997	1.000	1.000
Type-5	0.966	0.661	0.996	0.744	0.989	0.885	0.998	0.991	1.000	1.000
Type-6	0.971	0.912	0.993	0.942	0.999	0.974	0.999	0.999	1.000	1.000
$\mathrm{Err}_X(m = 3)/\mathrm{Err}_X(\mathrm{oSVD})$										
Type-1	1.029	0.994	1.064	0.998	1.036	1.001	1.026	1.001	1.003	1.000
Type-2	1.220	1.014	1.156	0.999	1.040	1.001	1.027	1.001	1.002	1.000
Type-3	1.298	1.150	1.223	1.020	1.053	1.001	1.026	1.001	1.002	1.000
Type-4	1.087	1.067	1.095	1.013	1.036	1.002	1.021	1.001	1.002	1.000
Type-5	1.075	0.998	1.087	1.000	1.029	1.002	1.027	1.001	1.003	1.000
Type-6	1.011	1.000	1.023	1.002	1.016	1.002	1.006	1.001	1.002	1.000

TABLE A.3

Comparison of REE and $\hat{k}$ for rank selection methods with various (N, n) pairs, and scenarios. For each different scenario, the factors' strengths are listed as the number of "strong/useful/harmful/undetectable" factors. For each (N, n) pair, the first column is the REE and the second column is $\hat{k}$. Both values are averages over 100 *simulations.* $\mathrm{Var}(\sigma_i^2) = 1$

Factor type	**Method**	$\gamma = 0.02$				$\gamma = 0.2$				$\gamma = 1$			
		(20, 1000)		**(100, 5000)**		**(20, 100)**		**(200, 1000)**		**(50, 50)**		**(500, 500)**	
Type-1	PA	**0.04**	5.5	0.07	7.0	**0.12**	4.9	0.10	6.9	**0.05**	5.4	0.13	7.0
0/6/1/1	ED	1.93	1.7	2.29	1.3	2.27	1.3	2.40	1.0	2.42	1.2	2.40	0.6
	ER	2.19	0.9	2.80	0.1	1.68	1.8	2.92	0.1	1.35	2.5	2.72	0.0
	IC1	2.30	16.0	0.69	3.3	1.44	16.0	0.61	3.5	0.10	5.6	0.69	3.1
	NE	0.23	6.3	1.82	1.3	0.16	5.0	2.45	0.6	0.08	5.4	2.36	0.5
	BCV	0.16	5.9	**0.03**	5.8	0.33	4.5	**0.01**	5.9	0.12	5.0	**0.00**	6.0
	Oracle	–	6.0	–	6.0	–	5.9	–	6.0	–	6.0	–	6.0
Type-2	PA	0.27	3.7	0.15	4.6	0.55	3.4	0.34	4.0	0.69	3.2	0.31	3.9
2/4/1/1	ED	0.61	3.5	1.03	2.9	0.95	3.0	1.18	2.5	1.00	3.0	1.03	2.6
	ER	1.52	1.8	1.21	2.0	1.64	1.9	1.33	2.0	1.34	2.0	1.23	2.0
	IC1	1.87	16.0	0.58	3.6	1.34	16.0	0.57	3.7	**0.09**	5.8	0.66	3.2
	NE	0.42	6.6	0.87	2.7	**0.12**	5.3	1.13	2.4	0.10	5.6	1.11	2.2
	BCV	**0.26**	5.4	**0.12**	5.7	0.24	4.5	**0.00**	5.9	0.19	4.7	**0.00**	6.0
	Oracle	–	5.1	–	5.8	–	5.5	–	6.0	–	5.9	–	6.0
Type-3	PA	0.35	3.2	0.46	3.1	0.62	3.1	0.72	3.0	0.76	3.0	0.69	3.0
3/3/1/1	ED	0.30	4.0	0.55	4.0	0.46	3.8	0.54	3.5	0.56	3.7	0.56	3.5
	ER	4.15	1.8	16.18	2.2	3.40	1.9	13.62	2.6	0.78	3.0	0.69	3.0
	IC1	1.70	16.0	0.41	4.2	1.23	16.0	0.41	4.1	0.11	5.9	0.52	3.5
	NE	0.41	6.8	0.41	3.7	**0.14**	5.5	0.56	3.4	**0.10**	5.6	0.60	3.2
	BCV	**0.17**	5.1	**0.26**	5.3	0.26	4.5	**0.08**	5.8	0.21	4.6	**0.01**	5.9
	Oracle	–	5.0	–	4.8	–	5.5	–	5.8	–	5.9	–	6.0
Type-4	PA	**0.01**	3.0	**0.02**	3.0	**0.03**	3.0	0.07	3.0	**0.05**	3.0	0.06	3.0
3/1/3/1	ED	0.11	3.3	0.81	4.4	0.08	3.3	0.29	3.9	0.07	3.3	0.08	3.8
	ER	5.10	1.8	19.24	2.2	3.50	1.9	16.79	2.5	3.33	2.3	0.50	3.0
	IC1	2.62	16.0	0.66	4.1	1.60	16.0	0.33	4.1	0.10	3.7	0.06	3.5
	NE	0.63	5.7	0.54	3.8	0.13	3.7	0.14	3.6	0.09	3.9	0.05	3.3
	BCV	0.02	3.1	0.19	3.5	**0.03**	3.3	**0.05**	3.7	**0.05**	3.1	**0.01**	3.9
	Oracle	–	3.2	–	3.2	–	3.5	–	3.9	–	3.8	–	4.0
Type-5	PA	**0.02**	3.4	**0.01**	4.3	**0.08**	3.0	**0.01**	3.8	0.10	2.9	0.02	3.7
1/3/3/1	ED	0.40	2.0	0.58	1.9	0.54	1.6	0.56	1.6	0.57	1.6	0.45	2.0
	ER	0.69	1.0	0.78	1.0	0.70	1.0	0.79	1.0	0.71	1.0	0.72	1.0
	IC1	2.63	16.0	0.41	2.1	1.53	16.0	0.45	2.0	0.10	3.3	0.55	1.5
	NE	0.40	5.3	0.48	1.9	0.13	3.2	0.59	1.5	**0.08**	3.5	0.62	1.2
	BCV	0.12	3.1	0.04	3.9	0.27	2.4	**0.01**	3.9	0.16	2.8	**0.00**	4.0
	Oracle	–	3.7	–	4.0	–	4.0	–	4.0	–	4.0	–	4.0
Type-6	PA	0.45	5.6	0.68	7.3	0.22	4.0	2.00	10.4	0.34	4.5	2.89	12.8
0/1/6/1	ED	0.07	0.8	0.11	1.8	0.06	0.7	0.12	1.4	0.06	0.4	0.09	1.1
	ER	0.07	0.1	0.09	0.1	**0.03**	0.2	0.08	0.1	0.05	0.1	0.06	0.1
	IC1	3.11	13.6	0.06	1.1	1.74	16.0	0.07	1.0	0.05	0.5	0.06	0.5
	NE	0.21	3.2	0.06	1.0	0.05	0.8	0.06	0.7	0.06	0.9	0.05	0.3
	BCV	**0.06**	0.2	**0.04**	1.0	**0.03**	0.1	**0.02**	0.8	**0.03**	0.0	**0.00**	1.0
	Oracle	–	1.0	–	1.0	–	0.8	–	1.0	–	0.8	–	1.0

TABLE A.4
Like Table A.3, but for larger γ

Factor		$\gamma = 5$				$\gamma = 50$			
type	**Method**	**(100, 20)**		**(1000, 200)**		**(1000, 20)**		**(5000, 100)**	
Type-1	PA	**0.05**	5.0	0.11	6.9	**0.01**	5.7	0.10	7.0
0/6/1/1	ED	1.89	1.2	1.57	1.6	0.43	4.7	0.10	6.1
	ER	2.23	0.3	2.18	0.0	1.69	0.0	1.68	0.0
	IC1	1.23	16.0	0.86	2.2	0.04	5.0	1.10	1.1
	NE	0.14	4.9	1.17	1.7	0.20	4.2	0.14	3.9
	BCV	0.37	4.1	**0.00**	6.0	0.10	4.9	**0.01**	5.8
	Oracle	–	5.9	–	6.0	–	5.8	–	5.9
Type-2	PA	0.68	2.8	0.23	3.9	0.32	3.1	0.12	4.0
2/4/1/1	ED	0.83	2.9	0.65	3.2	0.17	5.2	0.06	6.0
	ER	1.05	2.0	0.94	2.0	0.95	1.9	0.68	2.0
	IC1	1.24	16.0	0.86	2.2	**0.05**	5.0	0.68	2.0
	NE	**0.07**	5.2	0.77	2.4	0.08	4.5	0.13	4.0
	BCV	0.31	4.2	**0.00**	6.0	0.09	4.9	**0.01**	5.8
	Oracle	–	5.9	–	6.0	–	5.7	–	5.9
Type-3	PA	0.59	3.0	0.51	3.0	0.35	3.0	0.35	3.0
3/3/1/1	ED	0.48	3.6	0.36	3.9	0.11	5.5	0.06	6.2
	ER	3.51	1.9	22.02	2.1	3.33	2.0	15.40	2.0
	IC1	1.27	16.0	0.48	3.1	**0.04**	5.0	0.35	3.0
	NE	**0.09**	5.2	0.47	3.1	0.05	4.7	0.14	3.9
	BCV	0.25	4.5	**0.01**	5.8	0.09	4.6	**0.01**	5.8
	Oracle	–	5.9	–	6.0	–	5.8	–	5.9
Type-4	PA	**0.03**	3.0	0.03	3.0	**0.01**	3.0	**0.01**	3.0
3/1/3/1	ED	0.05	3.2	0.05	3.6	**0.01**	3.3	0.03	4.0
	ER	3.36	1.8	25.02	2.1	3.67	2.0	18.55	2.0
	IC1	1.53	16.0	0.03	3.1	**0.01**	3.0	**0.01**	3.0
	NE	0.04	3.4	0.03	3.2	**0.01**	3.0	**0.01**	3.0
	BCV	**0.03**	3.2	**0.01**	3.8	**0.01**	3.2	**0.01**	3.7
	Oracle	–	3.8	–	4.0	–	3.6	–	3.8
Type-5	PA	**0.11**	2.7	**0.01**	3.6	**0.01**	3.1	**0.00**	4.0
1/3/3/1	ED	0.42	1.8	0.32	2.1	0.31	1.9	0.12	3.7
	ER	0.57	1.0	0.57	1.0	0.43	1.0	0.42	1.0
	IC1	1.45	16.0	0.54	1.1	0.34	1.3	0.42	1.0
	NE	0.12	2.8	0.53	1.1	0.08	2.5	0.15	2.0
	BCV	0.22	2.4	**0.01**	3.9	0.12	2.6	0.02	3.8
	Oracle	–	3.9	–	4.0	–	3.7	–	3.8
Type-6	PA	0.29	3.4	2.27	10.5	0.77	5.4	1.24	7.1
0/1/6/1	ED	0.03	0.2	0.04	0.6	0.02	0.5	0.03	0.9
	ER	**0.02**	0.0	0.04	0.0	**0.01**	0.0	**0.01**	0.0
	IC1	1.00	7.4	0.03	0.1	**0.01**	0.0	**0.01**	0.0
	NE	0.03	0.2	0.03	0.2	**0.01**	0.0	**0.01**	0.0
	BCV	**0.02**	0.1	**0.01**	0.8	**0.01**	0.1	0.02	0.7
	Oracle	–	0.5	–	0.9	–	0.6	–	0.8

REFERENCES

[1] AHN, S. C. and HORENSTEIN, A. R. (2013). Eigenvalue ratio test for the number of factors. *Econometrica* **81** 1203–1227. MR3064065

[2] ALESSI, L., BARIGOZZI, M. and CAPASSO, M. (2010). Improved penalization for determining the number of factors in approximate factor models. *Statist. Probab. Lett.* **80** 1806–1813. MR2734245

[3] AMENGUAL, D. and WATSON, M. W. (2007). Consistent estimation of the number of dynamic factors in a large N and T panel. *J. Bus. Econom. Statist.* **25** 91–96. MR2338873

[4] BAI, J. and LI, K. (2012). Statistical analysis of factor models of high dimension. *Ann. Statist.* **40** 436–465. MR3014313

[5] BAI, J. and NG, S. (2002). Determining the number of factors in approximate factor models. *Econometrica* **70** 191–221. MR1926259

[6] BAI, J. and NG, S. (2008). *Large Dimensional Factor Analysis*. Now Publishers, Hanover.

[7] BAIK, J. and SILVERSTEIN, J. W. (2006). Eigenvalues of large sample covariance matrices of spiked population models. *J. Multivariate Anal.* **97** 1382–1408. MR2279680

[8] BARTLETT, M. S. (1954). A note on the multipying factors for various χ^2 approximations. *J. Roy. Statist. Soc. Ser. B.* **16** 296–298. MR0069457

[9] BENAYCH-GEORGES, F. and NADAKUDITI, R. R. (2012). The singular values and vectors of low rank perturbations of large rectangular random matrices. *J. Multivariate Anal.* **111** 120–135. MR2944410

[10] BUJA, A. and EYUBOGLU, N. (1992). Remarks on parallel analysis. *Multivar. Behav. Res.* **27** 509–540.

[11] CARUANA, R., LAWRENCE, S. and GILES, L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems* 13: *Proceedings of the* 2000 *Conference* **13** 402–408. MIT Press, Cambridge, MA.

[12] CATTELL, R. B. (1966). The scree test for the number of factors. *Multivar. Behav. Res.* **1** 245–276.

[13] CATTELL, R. B. and VOGELMANN, S. (1977). A comprehensive trial of the scree and KG criteria for determining the number of factors. *Multivar. Behav. Res.* **12** 289–325.

[14] CHANDRASEKARAN, V., PARRILO, P. A. and WILLSKY, A. S. (2012). Latent variable graphical model selection via convex optimization. *Ann. Statist.* **40** 1935–1967. MR3059067

[15] CHOI, Y., TAYLOR, J. and TIBSHIRANI, R. (2014). Selecting the number of principal components: Estimation of the true rank of a noisy matrix. Preprint. Available at arXiv:1410.8260.

[16] EASLEY, D. and KLEINBERG, J. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge Univ. Press, Cambridge. MR2677125

[17] FISHLER, E., GROSMANN, M. and MESSER, H. (2002). Detection of signals by information theoretic criteria: General asymptotic performance analysis. *IEEE Trans. Signal Process.* **50** 1027–1036. MR1929130

[18] FLEMING, H. E. (1990). Equivalence of regularization and truncated iteration in the solution of ill-posed image reconstruction problems. *Linear Algebra Appl.* **130** 133–150. MR1057805

[19] FORNI, M., HALLIN, M., LIPPI, M. and REICHLIN, L. (2000). The generalized dynamic-factor model: Identification and estimation. *Rev. Econom. Statist.* **82** 540–554.

[20] FORNI, M. and REICHLIN, L. (2001). Federal policies and local economies: Europe and the US. *Eur. Econ. Rev.* **45** 109–134.

[21] GAGNON-BARTSCH, J. A. and SPEED, T. P. (2012). Using control genes to correct for unwanted variation in microarray data. *Biostatistics* **13** 539–552.

[22] GAVISH, M. and DONOHO, D. L. (2014). The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Trans. Inform. Theory* **60** 5040–5053. MR3245370

[23] HALLIN, M. and LIŠKA, R. (2007). Determining the number of factors in the general dynamic factor model. *J. Amer. Statist. Assoc.* **102** 603–617. MR2325115

[24] HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2009). *The Elements of Statistical Learning*, 2nd ed. Springer, New York. MR2722294

[25] HERMUS, K., WAMBACQ, P. and VAN HAMME, H. (2007). A review of signal subspace speech enhancement and its application to noise robust speech recognition. *EURASIP J. Adv. Signal Process.* Art. ID 45821, 15. MR2366601

[26] HOCHREITER, S., CLEVERT, D.-A. and OBERMAYER, K. (2006). A new summarization method for Affymetrix probe level data. *Bioinformatics* **22** 943–949.

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

[28] HUBBARD, R. and ALLEN, S. J. (1987). An empirical comparison of alternative methods for principal component extraction. *J. Bus. Res.* **15** 173–190.

[29] JOLLIFFE, I. T. (1986). *Principal Component Analysis*. Springer, New York. MR0841268

[30] KAISER, H. F. (1960). The application of electronic computers to factor analysis. *Educ. Psychol. Meas.* **20** 141-151.

[31] KAPETANIOS, G. (2004). A new method for determining the number of factors in factor models with large datasets. Technical report, Working Paper, Dept. of Economics, Queen Mary University of London.

[32] KAPETANIOS, G. (2010). A testing procedure for determining the number of factors in approximate factor models with large datasets. *J. Bus. Econom. Statist.* **28** 397–409. MR2723608

[33] KRITCHMAN, S. and NADLER, B. (2008). Determining the number of components in a factor model from limited noisy data. *Chemom. Intell. Lab. Syst.* **94** 19–32.

[34] LAM, C. and YAO, Q. (2012). Factor modeling for high-dimensional time series: Inference for the number of factors. *Ann. Statist.* **40** 694–726. MR2933663

[35] LAN, W. and DU, L. (2014). A factor-adjusted multiple testing procedure with application to mutual fund selection. Preprint. Available at arXiv:1407.5515.

[36] LAWLEY, D. N. (1956). Tests of significance for the latent roots of covariance and correlation matrices. *Biometrika* **43** 128–136. MR0078610

[37] LEEK, J. T. and STOREY, J. D. (2008). A general framework for multiple testing dependence. *Proc. Natl. Acad. Sci. USA* **105** 18718–18723.

[38] LOVE, D., HALLBAUER, D., AMOS, A. and HRANOVA, R. (2004). 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** 1135–1143.

[39] NADAKUDITI, R. R. (2014). OptShrink: An algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage. *IEEE Trans*. *Inform*. *Theory* **60** 3002–3018. MR3200641

[40] NADAKUDITI, R. R. and EDELMAN, A. (2008). Sample eigenvalue based detection of high-dimensional signals in white noise using relatively few samples. *IEEE Trans*. *Signal Process*. **56** 2625–2638. MR1500236

[41] NADLER, B. (2008). Finite sample approximation results for principal component analysis: A matrix perturbation approach. *Ann*. *Statist*. **36** 2791–2817. MR2485013

[42] ONATSKI, A. (2010). Determining the number of factors from empirical distribution of eigenvalues. *Rev*. *Econ*. *Stat*. **92** 1004–1016.

[43] ONATSKI, A. (2012). Asymptotics of the principal components estimator of large factor models with weakly influential factors. *J*. *Econometrics* **168** 244–258. MR2923766

[44] ONATSKI, A. (2015). Asymptotic analysis of the squared estimation error in misspecified factor models. *J*. *Econometrics* **186** 388–406. MR3343793

[45] OWEN, A. B. and PERRY, P. O. (2009). Bi-cross-validation of the SVD and the nonnegative matrix factorization. *Ann*. *Appl*. *Stat*. **3** 564–594. MR2750673

[46] OWEN, A. B. and WANG, J. (2015). Bi-cross-validation for factor analysis (v1). Preprint. Available at arXiv:1503.03515.

[47] PAQUE, J. M., BROWNING, R., KING, P. L. and PIANETTA, P. (1990). Quantitative information from x-ray images of geological materials. In *Proceedings of the XIIth International Congress for Electron Microscopy* **2** 244–247. San Francisco Press, San Francisco, CA.

[48] PATTERSON, N., PRICE, A. L. and REICH, D. (2006). Population structure and eigenanalysis. *PLoS Genetics* **2** e190.

[49] PAUL, D. (2007). Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statist*. *Sinica* **17** 1617–1642. MR2399865

[50] PERES-NETO, P. R., JACKSON, D. A. and SOMERS, K. M. (2005). How many principal components? Stopping rules for determining the number of non-trivial axes revisited. *Comput*. *Statist*. *Data Anal*. **49** 974–997. MR2143053

[51] PERRY, P. O. (2009). Cross-validation for unsupervised learning. Preprint. Available at arXiv:0909.3052.

[52] PRICE, A. L., PATTERSON, N. J., PLENGE, R. M., WEINBLATT, M. E., SHADICK, N. A. and REICH, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nat*. *Genet*. **38** 904–909.

[53] ROSSET, S., ZHU, J. and HASTIE, T. (2004). Boosting as a regularized path to a maximum margin classifier. *J*. *Mach*. *Learn*. *Res*. **5** 941–973. MR2248005

[54] ROY, S. N. (1953). On a heuristic method of test construction and its use in multivariate analysis. *Ann*. *Math*. *Stat*. **24** 220–238. MR0057519

[55] SPEARMAN, C. (1904). "General intelligence," objectively determined and measured. *Am*. *J*. *Psychol*. **15** 201–292.

[56] SUN, Y., ZHANG, N. R. and OWEN, A. B. (2012). Multiple hypothesis testing adjusted for latent variables, with an application to the AGEMAP gene expression data. *Ann*. *Appl*. *Stat*. **6** 1664–1688. MR3058679

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

[58] VELICER, W. F., EATON, C. A. and FAVA, J. L. (2000). 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* 41–71. Springer, Berlin.

[59] WAX, M. and KAILATH, T. (1985). Detection of signals by information theoretic criteria. *IEEE Trans*. *Acoust*. *Speech Signal Process*. **33** 387–392. MR0788604

[60] YAO, Y., ROSASCO, L. and CAPONNETTO, A. (2007). On early stopping in gradient descent learning. *Constr. Approx*. **26** 289–315. MR2327601

[61] ZHANG, T. and YU, B. (2005). Boosting with early stopping: Convergence and consistency. *Ann*. *Statist*. **33** 1538–1579. MR2166555

[62] ZWICK, W. R. and VELICER, W. F. (1986). Comparison of five rules for determining the number of components to retain. *Psychol*. *Bull*. **99** 432–442.