Macroeconomic Forecasting using Approximate Factor Models with Outliers

RAY YEUTIEN CHOU,* TSO-JUNG YEN,† YU-MIN YEN‡

September 10, 2017

Abstract

Approximate factor models and their extensions are widely used in forecasting and economic analysis due to their ability to extracting useful information from a large number of relevant variables. In these models, candidate predictors are typically subject to some common components. In this paper, we consider to efficiently estimate an approximate factor model in which the candidate predictors are additionally subject to idiosyncratic large uncommon components such as jumps or outliers. By assuming that occurrences of the uncommon components are rare, we propose an estimation procedure to simultaneously disentangle and estimate the common and uncommon components. We formulate the estimation problem as a penalized least squares problem in which a norm penalty function is imposed on the uncommon components. To solve the estimation problem, we propose an algorithm, which iteratively solves a principal component analysis (PCA) problem and a one dimensional shrinkage estimation problem. The algorithm is flexible in incorporating methods for selecting the number of common components. We then compare finite-sample efficiency of the proposed method and traditional PCA method with simulations. We also demonstrate performances of the proposed method with empirical applications on predicting yearly growths of important macroeconomic indicators.

KEYWORDS: Approximate Factor Model, PCA, Norm Penalty

*Research fellow, Institute of Economics, Academa Sinica. Address: 128 Academia Road, Section 2, Nankang, Taipei 115, Taiwan. E-mail: rchou@econ,sinica.edu.tw.

†Assistant research fellow, Institute of Statistical Science, Academia Sinica. Address: 128 Academia Road, Section 2, Nankang, Taipei 11529, Taiwan. E-mail: tjyen@stat.sinica.edu.tw.

‡Assistant professor, Department of International Business, National Chengchi University, Address: NO.64,Sec.2,ZhiNan Rd.,Wenshan District,Taipei City 11605, Taiwan Email: yyu_min@nccu.edu.tw

1 Introduction

In this paper we consider robust estimations on a class of approximate factor models in which the data generating process is subject to large idiosyncratic uncommon components. Approximate factor models and their extensions are widely used in economic analysis and forecasting due to their ability to extracting useful information from a large number of relevant variables (e.g., Stock and Watson, 2002; Bernanke et al., 2005; Ludvigson and Ng, 2009). In such models, data generating process often are specified as a linear combination of relevant common factors and error terms. Due to the nature that some of the relevant common factors are not observable, an important goal for estimations of such models is to identify the latent common factors and their factor loadings. Methods such as maximum likelihood (MLE), Markov Chain Monte Carlo (MCMC) and Principal Component Analysis (PCA) have been shown to be useful on this task. Nevertheless in econometrics, researchers often estimate the models with high dimensional data and thus PCA method, which is much less computational intensive than MLE and MCMC, is often more preferred in practice.

Although PCA method has a computational advantage, it is widely known that the method may fail to yield accurate estimations on the latent factors and factor loadings when large idiosyncratic uncommon components are presented in the data (Jolliffe, 2002). In this paper, we propose a simple and efficient method to estimate a class of approximate factor models in which the data are additionally subject to the large idiosyncratic uncommon components. The estimation problem is formulated as a penalized least squares problem in which a norm penalty function is imposed on the uncommon components. The proposed estimation procedure can simultaneously disentangle and estimate the common and uncommon components and therefore can reduce estimation biases in the latent common factors and their factor loadings. To solve the estimation problem, we propose an algorithm, which iteratively solves a principal component analysis (PCA) problem and a one dimensional shrinkage estimation problem. The algorithm is flexible in incorporating methods for selecting the number of common components. We call the proposed estimation procedure P-PCA method (*Penalized least squares plus PCA method*).

Recently many different approximate factor models and their related estimation procedures are proposed. Moench et al. (2013) propose a multilevel factor model for large panel data with between-block variations and idiosyncratic noise. They propose an estimation procedure which can both separate block-level shocks and genuinely common factors and achieve dimension reduction. Ando and Bai (2013) propose a multifactor

model for data with a large number of observable factors and unobservable common and group-specific pervasive factors. Their proposed estimation procedure for such a model can simultaneously select relevant observable factors and determine the number of common and group-specific unobservable factors. Cheng et al. (2014) propose a factor model in which both factor loadings and number of factors can have a behaviour of structure break. They use a shrinkage estimator that can simultaneously and consistently estimate the number of common factors before and after the structure break. Their proposed estimation procedure can be implemented by solving a convex optimization with the principal components of data matrix as its inputs.

A main difference between the aforementioned research and our paper is that we consider a data generating process which is subject to the large idiosyncratic uncommon components. Such a data generating process is more appropriately to be viewed as an observation occasionally blurred by extreme large signals, like asset price jumps in financial data, rather than broken by a permanent change of common factors or factor loadings. Indeed, under suitable assumptions (e.g., Bai and Ng, 2002; Stock and Watson, 2002) on the idiosyncratic uncommon components, the factors and factor loadings might still be consistently estimated by using PCA method. However, in term of finite sample efficiency, we show that P-PCA estimation procedure can outperform PCA method on estimating the model parameters through intensive simulations under a wide range of model settings. In addition, we discuss how the proposed method can be used for a more general data structure, such as panel data. We also demonstrate how P-PCA method performs by empirical applications on predicting yearly growth of important macroeconomic variables and investigating how latent factors affect asset returns. Throughout these works, we believe the proposed method can serve as a complementary tool for robust estimations rather than a competitive approach to those established approximate factor models.

The rest of paper is organized as follows. In Section 2 we first review PCA method and then introduce P-PCA method. In Section 3.1 we discuss how to select number of the latent common factors in our estimation procedure. We then report simulation results in Section 4. In Section 5 we perform empirical applications. Section 6 is a conclusion.

2 Methodology

In this section we introduce our method for estimating an approximate factor model in which the data are subject to idiosyncratic uncommon components. Specifically we

assume the N dimensional time series of candidate predictors $\mathbf{X}_t$ and the variable to be forecast Y_t subject to the following data generating process:

$$\begin{aligned} \mathbf{X}_t &= \mathbf{\Lambda}\mathbf{F}_t + \mathbf{J}_t + \mathbf{e}_t, & (1) \\ Y_{t+h} &= \boldsymbol{\beta}_F^{\mathbf{T}}\mathbf{F}_t + \boldsymbol{\beta}_W^{\mathbf{T}}\mathbf{W}_t + \varepsilon_{t+h}, & (2) \end{aligned}$$

where $t = 1, \ldots, T$, $\dim(\mathbf{X}_t) = N \times 1$, $\dim(\mathbf{\Lambda}) = N \times r$, $\dim(\mathbf{F}_t) = r \times 1$, $\dim(\mathbf{J}_t) = N \times 1$, $\dim(\mathbf{e}_t) = N \times 1$, $\dim(\boldsymbol{\beta}_F) = r \times 1$, $\dim(\boldsymbol{\beta}_W) = m \times 1$ and $\dim(\mathbf{W}_t) = m \times 1$. In the data generating process, $\mathbf{\Lambda}$ is a factor loading matrix, $\mathbf{F}_t$ is a vector for latent factors and $\mathbf{e}_t$ is a vector for measurement errors. $\mathbf{J}_t$ is a vector for the idiosyncratic uncommon components. By assuming that occurrences of the uncommon components are rare, $\mathbf{J}_t$ is generically a sparse vector (some of its components are zero). $\mathbf{W}_t$ is a vector for observable exogenous variables. The index h denotes the forecast horizon. Y_{t+h} and ε_{t+h} are the variables to be forecast and error term h periods ahead respectively and they are scalars. The setting is similar to the dynamic factor model considered in Stock and Watson (2002) except $\mathbf{X}_t$ has an additional idiosyncratic uncommon component $\mathbf{J}_t$, which can be viewed as a jump or outlier in $\mathbf{X}_t$.

We first review PCA method for estimating the latent factors $\mathbf{F}_t$ and factor loadings $\mathbf{\Lambda}$. Let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_T)^{\mathbf{T}}$, $\mathbf{F} = (\mathbf{F}_1, \ldots, \mathbf{F}_T)^{\mathbf{T}}$ and $\mathbf{J} = (\mathbf{J}_1, \ldots, \mathbf{J}_T)^{\mathbf{T}}$. Suppose $N > T$ and the number of factors r is known. Without the term $\mathbf{J}_t$, we can solve the following optimization to estimate the factor matrix $\mathbf{F}$ and factor loading matrix $\mathbf{\Lambda}$:

$$\min_{\mathbf{F},\mathbf{\Lambda}} \frac{1}{TN} \left\| \mathbf{X} - \mathbf{F}\mathbf{\Lambda}^{\mathbf{T}} \right\|_F^2, \text{ subject to } \frac{\mathbf{F}^{\mathbf{T}}\mathbf{F}}{T} = \mathbf{I}_r. \quad (3)$$

Here $\|.\|_F$ denotes the Frobenius norm. The above optimization is closely related to the principal component analysis (PCA). The estimated $\mathbf{F}$, denoted by $\hat{\mathbf{F}}$, can be obtained by $\sqrt{T}$ times a matrix containing the eigenvectors corresponding to the largest r eigenvalues of the $T \times T$ matrix $\mathbf{X}\mathbf{X}^{\mathbf{T}}$. Given $\hat{\mathbf{F}}$, the factor loading matrix can be estimated by using the least squares method: $\hat{\mathbf{\Lambda}} = \left(\left(\hat{\mathbf{F}}^{\mathbf{T}}\hat{\mathbf{F}} \right) \hat{\mathbf{F}}^{\mathbf{T}}\mathbf{X} \right)^{\mathbf{T}} = \mathbf{X}^{\mathbf{T}}\hat{\mathbf{F}}/T$. When $T \geq N$, we can estimate the factor and factor loading matrices by solving the above optimization but with the constraint $\mathbf{F}^{\mathbf{T}}\mathbf{F}/T = \mathbf{I}_r$ replaced by $\mathbf{\Lambda}^{\mathbf{T}}\mathbf{\Lambda}/N = \mathbf{I}_r$. In this situation the estimated factor loading matrix, denoted by $\bar{\mathbf{\Lambda}}$, is given by $\sqrt{N}$ times a matrix containing the eigenvectors corresponding to the largest r eigenvalues of the $N \times N$ matrix $\mathbf{X}^{\mathbf{T}}\mathbf{X}$. Given $\bar{\mathbf{\Lambda}}$, the factor matrix can be estimated by using the least squares method: $\bar{\mathbf{F}} = \left(\left(\bar{\mathbf{\Lambda}}^{\mathbf{T}}\bar{\mathbf{\Lambda}} \right) \bar{\mathbf{\Lambda}}^{\mathbf{T}}\mathbf{X}^{\mathbf{T}} \right)^{\mathbf{T}} = \mathbf{X}\bar{\mathbf{\Lambda}}/N$. Let $\mathbf{Z} = \mathbf{F}\mathbf{\Lambda}^{\mathbf{T}}$, $\hat{\mathbf{Z}} = \hat{\mathbf{F}}\hat{\mathbf{\Lambda}}^{\mathbf{T}}$ and $\bar{\mathbf{Z}} = \bar{\mathbf{F}}\bar{\mathbf{\Lambda}}^{\mathbf{T}}$. The matrices $\hat{\mathbf{Z}}$ and $\bar{\mathbf{Z}}$ can be viewed as low rank approximations for the matrix $\mathbf{X}$. It

is known that $\hat{\mathbf{Z}} = \bar{\mathbf{Z}}$, and hence the objective function $\left\|\mathbf{X} - \mathbf{F}\mathbf{\Lambda}^{\mathbf{T}}\right\|_F^2 / (TN)$ has the same value under the two optimal solutions $\left(\hat{\mathbf{F}}, \hat{\mathbf{\Lambda}}\right)$ and $\left(\bar{\mathbf{F}}, \bar{\mathbf{\Lambda}}\right)$.

2.1 Penalized Least Squares plus PCA method

If the term $\mathbf{J}_t$ is in the data generating process of $\mathbf{X}_t$, directly apply PCA method to the matrix $\mathbf{X}\mathbf{X}^{\mathbf{T}}$ or $\mathbf{X}^{\mathbf{T}}\mathbf{X}$ may yield loss of efficiency in estimating $\mathbf{F}$ and $\mathbf{\Lambda}$. In this situation if $\mathbf{J}$ is known, we can apply PCA method to the matrix $\mathbf{C}\mathbf{C}^{\mathbf{T}}$ (or $\mathbf{C}^{\mathbf{T}}\mathbf{C}$), where $\mathbf{C} = \mathbf{X} - \mathbf{J}$, to obtain better estimations of $\mathbf{F}$ and $\mathbf{\Lambda}$. If $\mathbf{J}$ is unknown, we can try to estimate $\mathbf{J}$ and disentangle the estimated $\hat{\mathbf{J}}$ from $\mathbf{X}$, and apply PCA method to the matrix $\hat{\mathbf{C}}\hat{\mathbf{C}}^{\mathbf{T}}$ (or $\hat{\mathbf{C}}^{\mathbf{T}}\hat{\mathbf{C}}$), where $\hat{\mathbf{C}} = \mathbf{X} - \hat{\mathbf{J}}$.

Our strategy to estimate $\mathbf{J}$ is to use the property that $\mathbf{J}$ is a sparse matrix (by our assumption). To result in a sparse estimation for $\mathbf{J}$, a commonly adopted method is to impose an l_p norm penalty on $\mathbf{J}$, where $0 \leq p \leq 1$. In the following, we focus on using an l_1 norm penalty on the idiosyncratic uncommon component matrix $\mathbf{J}$.

Suppose $N > T$ and the number of factors r is known, to estimate $\mathbf{F}$, $\mathbf{\Lambda}$ and $\mathbf{J}$, we propose to solve the following penalized l_1 norm optimization:

$$\min_{\mathbf{F},\mathbf{\Lambda},\mathbf{J}} \frac{1}{TN} \left\|\mathbf{X} - \mathbf{F}\mathbf{\Lambda}^{\mathbf{T}} - \mathbf{J}\right\|_F^2 + \frac{\delta}{TN} \|\mathbf{J}\|_1, \text{ subject to } \frac{\mathbf{F}^{\mathbf{T}}\mathbf{F}}{T} = \mathbf{I}_r. \tag{4}$$

Here $\delta \in \mathbb{R}^+$ is penalty parameter and the l_1 norm penalty is imposed on each entry in the matrix $\mathbf{J}$ (i.e., $\|\mathbf{J}\|_1$ is sum of absolute value of each element in $\mathbf{J}$).

The l_1 norm penalty perhaps is the most frequently used norm penalty on sparse estimations. Famous examples of using the l_1 norm penalty on sparse estimations include the lasso of Tibshirani (1996) and the robust PCA method of Candès et al. (2011). The l_1 norm penalty is a convex function of $\mathbf{J}$, which makes the modified estimation problem still easily tractable when T and N becomes very large. In fact, except for the l_1 norm, there does not exist a norm penalty which can simultaneously produce a sparse estimation as well as being a convex function for the matrix $\mathbf{J}$.

In the following we provide a step-by-step description on how to implement an algorithm to solve the optimization of (4) given that the number of factors r is known and the penalty parameter δ is fixed.

Step 1. Set the initial value of $\mathbf{J}$, $\mathbf{J}^{(0)} = \mathbf{0}$.

Step 2. Given $\mathbf{J} = \mathbf{J}^{(0)}$, solve (4). When $\mathbf{J}^{(0)} = \mathbf{0}$, it is equivalent to solving (3) and as mentioned we can obtain the optimal solutions by using PCA method.

Let $\left(\mathbf{F}^{(1)}, \mathbf{\Lambda}^{(1)}\right)$ denote the optimal solutions from using PCA method and $\mathbf{Z}^{(1)} = \mathbf{F}^{(1)}\mathbf{\Lambda}^{(1)\mathbf{T}}$.

Step 3. Plug $\mathbf{Z}^{(1)}$ into (4) and solve the optimization with respect to $\mathbf{J}$, which is equivalent to solving

$$\min_{\mathbf{J}} \frac{1}{TN} \left\|\mathbf{X} - \mathbf{Z}^{(1)} - \mathbf{J}\right\|_F^2 + \frac{\delta}{TN} \|\mathbf{J}\|_1 . \tag{5}$$

Let $\mathbf{L}^{(1)} = \left(\mathbf{L}_1^{(1)}, \ldots, \mathbf{L}_T^{(1)}\right)^{\mathbf{T}} = \mathbf{X} - \mathbf{Z}^{(1)}$ and $L_{it}^{(1)}$ and J_{it} denote the ith elements in the vector $\mathbf{L}_t^{(1)}$ and $\mathbf{J}_t$, $i = 1, \ldots, N$ and $t = 1, \ldots, T$. The optimization of (5) can be reformulated as

$$\min_{J_{it}, i=1,\ldots,N, t=1,\ldots,T} -\frac{2}{NT} \sum_{i=1}^{N}\sum_{t=1}^{T} J_{it}L_{it}^{(1)} + \frac{1}{NT} \sum_{i=1}^{N}\sum_{t=1}^{T} J_{it}^2 + \frac{\delta}{NT} \sum_{i=1}^{N}\sum_{t=1}^{T} |J_{it}| . \tag{6}$$

The reformulated optimization of (6) is separable. It means that each optimal J_{it} can be obtained by separately solving the following one dimensional optimization:

$$\min_{J_{it}} -2J_{it}L_{it}^{(1)} + J_{it}^2 + \delta |J_{it}| .$$

The optimal J_{it}, denoted by $J_{it}^{(1)}$, is then given by

$$J_{it}^{(1)} = \begin{cases} L_{it}^{(1)} - \frac{\delta}{2}, & \text{if } L_{it}^{(1)} > \frac{\delta}{2}, \\ 0, & \text{if } -\frac{\delta}{2} \le L_{it}^{(1)} \le \frac{\delta}{2}, \\ L_{it}^{(1)} + \frac{\delta}{2}, & \text{if } L_{it}^{(1)} < -\frac{\delta}{2}, \end{cases}$$

or more concisely,

$$J_{it}^{(1)} = ST\left(L_{it}^{(1)}, \frac{\delta}{2}\right),$$

where $ST(x, y) := sign(x)(|x| - y)_+$ is the softthresholding function.

Step 4. Let $\mathbf{J}_t^{(1)}$ be the vector in which the ith element is $J_{it}^{(1)}$, $i = 1, \ldots, N$ and $t = 1, \ldots, T$ and $\mathbf{J}^{(1)} = \left(\mathbf{J}_1^{(1)}, \ldots, \mathbf{J}_T^{(1)}\right)^{\mathbf{T}}$. Update $\mathbf{J}$ with $\mathbf{J}^{(1)}$ and plug it into (4) and solve the optimization with respect to $(\mathbf{F}, \mathbf{\Lambda})$, which is equivalent to solving

$$\min_{\mathbf{F}, \mathbf{\Lambda}} \frac{1}{TN} \left\|\mathbf{C}^{(1)} - \mathbf{F}\mathbf{\Lambda}^{\mathbf{T}}\right\|_F^2, \text{ subject to } \frac{\mathbf{F}^{\mathbf{T}}\mathbf{F}}{T} = \mathbf{I}_r, \tag{7}$$

where $\mathbf{C}^{(1)} = \mathbf{X} - \mathbf{J}^{(1)}$. Again we solve (7) by using PCA method, and let $\left(\mathbf{F}^{(2)}, \mathbf{\Lambda}^{(2)}\right)$ be the optimal solutions and $\mathbf{Z}^{(2)} = \mathbf{F}^{(2)}\mathbf{\Lambda}^{(2)\mathbf{T}}$.

Step 5. Plug $\mathbf{Z}^{(2)}$ into (4) and solve the optimization with respect to $\mathbf{J}$ as we do in step 3. Let the optimal solution denoted by $\mathbf{J}^{(2)}$ and use it to update $\mathbf{J}$.

Step 6. Repeat step 4 and 5 to obtain $\left(\mathbf{F}^{(k)}, \mathbf{\Lambda}^{(k)}\right)$, $\mathbf{Z}^{(k)}$ and $\mathbf{J}^{(k)}$, $k = 1, \ldots,$ until the following convergence condition met:

$$\frac{\left\|\mathbf{Z}^{(\bar{k})} - \mathbf{Z}^{(\bar{k}-1)}\right\|_F}{\left\|\mathbf{Z}^{(\bar{k}-1)}\right\|_F} \leq \epsilon,$$

where $\bar{k} \geq 1$ is the number of iterations for step 4 and 5. The outputs $\left(\mathbf{F}^{(\bar{k})}, \mathbf{\Lambda}^{(\bar{k})}\right)$ and $\mathbf{J}^{(\bar{k})}$ are used as the estimations for $(\mathbf{F}, \mathbf{\Lambda})$ and $\mathbf{J}$.

The above algorithm can be summarized as the following.

Algorithm 1 Robust Approximate Factor Model Estimation with l_1 Norm Penalty

Input: Data matrix $\mathbf{X}$, penalty penalty parameter δ, tolerance ϵ, and number of factors r and maximum iteration k_{max}

Output: $\hat{\mathbf{F}}$, $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{J}}$

1: Set $\mathbf{J}^{(0)} = \mathbf{0}$ and $\delta > 0$
2: **for** $k = 1$ to k_{max} **do**
3: Given $\mathbf{C}^{(k-1)} = \mathbf{X} - \mathbf{J}^{(k-1)}$, where $\mathbf{J}^{(k-1)} = (\mathbf{J}_1^{(k-1)}, \ldots, \mathbf{J}_T^{(k-1)})^{\mathbf{T}}$, compute $\left(\mathbf{F}^{(k)}, \mathbf{\Lambda}^{(k)}\right)$ by using PCA method and least squares method
4: Given $\mathbf{L}^{(k)} = \left(\mathbf{L}_1^{(k)}, \ldots, \mathbf{L}_T^{(k)}\right)^{\mathbf{T}} = \mathbf{X} - \mathbf{Z}^{(k)}$, where $\mathbf{Z}^{(k)} = \mathbf{F}^{(k)}\mathbf{\Lambda}^{(k)\mathbf{T}}$, update $J_{it}^{(k)}$ the ith element of vector $\mathbf{J}_t^{(k)}$, $t = 1, \ldots, T$ as follows

$$J_{it}^{(k)} = ST\left(L_{it}^{(k)}, \frac{\delta}{2}\right).$$

Here $ST(x, y) := sign(x)(|x| - y)_+$ is the softthresholding function and $L_{it}^{(k)}$ is the ith element in vector $\mathbf{L}_t^{(k)}$, $t = 1, \ldots, T$
5: **if** $\left\|\mathbf{Z}^{(k)} - \mathbf{Z}^{(k-1)}\right\|_F / \left\|\mathbf{Z}^{(k-1)}\right\|_F \leq \epsilon$ **then**
6: **break**
7: **end if**
8: **end for**
9: Set outputs $\hat{\mathbf{F}} = \mathbf{F}^{(k)}$, $\hat{\mathbf{\Lambda}} = \mathbf{\Lambda}^{(k)}$ and $\hat{\mathbf{J}} = \mathbf{J}^{(k)}$

We call the proposed estimation procedure P-PCA method (Penalized least squares plus PCA method). Note that in the algorithm, to identify $\mathbf{J}$, it is not necessary to

identify the factor $\mathbf{F}$ and factor loading matrices $\mathbf{\Lambda}$ and only knowing their product $\mathbf{Z} = \mathbf{F}\mathbf{\Lambda}^{\mathbf{T}}$ is enough.

2.2 Convergence of the Algorithm

The algorithm can be viewed as iteratively choosing a low rank matrix and a sparse matrix to minimize the objective function. In the following we show that the algorithm indeed decreases the objective function in each iteration. We first show that using PCA and OLS to estimate the factors and factor loadings is equivalent to solving a low rank approximation problem.

2.2.1 Low Rank Matrix Approximation

From the theorem of singular value decomposition (SVD), a matrix $\mathbf{C} \in \mathbb{R}^{T\times N}$ admits a decomposition of the form:

$$\mathbf{C} = \mathbf{U}\tilde{\mathbf{L}}\mathbf{V}^{\mathbf{T}},\ \tilde{\mathbf{L}} = \begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where $\mathbf{U} \in \mathbb{R}^{T\times T}$, $\mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{I}_T$ and $\mathbf{V} \in \mathbb{R}^{N\times N}$, $\mathbf{V}^{\mathbf{T}}\mathbf{V} = \mathbf{I}_N$. The matrix $\tilde{\mathbf{L}} \in \mathbb{R}^{T\times N}$ and $\mathbf{L} = \mathbf{diag}\,(l_1, \ldots, l_q)$ is a $q \times q$ diagonal matrix and $q \leq \min(T, N)$ is rank of the matrix $\mathbf{C}$. The diagonal elements $l_1, \ldots, l_q$, called singular values of the matrix $\mathbf{C}$, are all positive and unique. The first q columns of $\mathbf{U}$ are called left singular vectors of $\mathbf{C}$ and the first q columns of $\mathbf{V}$ are called right singular vectors of $\mathbf{C}$. With the SVD, it can be shown that $\mathbf{C}\mathbf{C}^{\mathbf{T}} = \mathbf{U}\tilde{\mathbf{L}}\tilde{\mathbf{L}}^{\mathbf{T}}\mathbf{U}^{\mathbf{T}}$ and $\mathbf{C}^{\mathbf{T}}\mathbf{C} = \mathbf{V}\tilde{\mathbf{L}}^{\mathbf{T}}\tilde{\mathbf{L}}\mathbf{V}^{\mathbf{T}}$. It is known that nonzero eigenvalues of $\mathbf{C}\mathbf{C}^{\mathbf{T}}$ and $\mathbf{C}^{\mathbf{T}}\mathbf{C}$ are the same and are given by $l_1^2, \ldots, l_q^2$. The corresponding eigenvectors of $\mathbf{C}\mathbf{C}^{\mathbf{T}}$ and $\mathbf{C}^{\mathbf{T}}\mathbf{C}$ are $\mathbf{U}$ and $\mathbf{V}$ respectively (the left and right singular vectors of $\mathbf{C}$).

If the matrix $\mathbf{J}$ is known and $N > T$, solving (4) is equivalent to solving

$$\min_{\mathbf{F},\mathbf{\Lambda}} \frac{1}{TN} \left\|\mathbf{C} - \mathbf{F}\mathbf{\Lambda}^{\mathbf{T}}\right\|_F^2,\ \text{subject to}\ \frac{\mathbf{F}^{\mathbf{T}}\mathbf{F}}{T} = \mathbf{I}_r,$$

where $\mathbf{C} = \mathbf{X} - \mathbf{J}$. Using PCA method yields the estimated $\hat{\mathbf{F}} = \sqrt{T}\mathbf{U}_r$, where $\mathbf{U}_r$ is a matrix containing the eigenvectors corresponding to the largest r eigenvalues of the $T \times T$ matrix $\mathbf{C}\mathbf{C}^{\mathbf{T}}$. Using OLS method yields the estimated $\hat{\mathbf{\Lambda}} = \mathbf{C}^{\mathbf{T}}\hat{\mathbf{F}}/T$. By using

the fact $\mathbf{U}_r^{\mathbf{T}}\mathbf{U} = (\mathbf{I}_r, \mathbf{0})$, it can be shown that

$$\begin{aligned}\hat{\mathbf{Z}} &= \hat{\mathbf{F}}\hat{\mathbf{\Lambda}}^{\mathbf{T}} \\ &= \mathbf{U}_r\mathbf{U}_r^{\mathbf{T}}\mathbf{U}\tilde{\mathbf{L}}\mathbf{V}^{\mathbf{T}} \\ &= \mathbf{U}\tilde{\mathbf{L}}_r\mathbf{V}^{\mathbf{T}},\end{aligned}$$

where

$$\tilde{\mathbf{L}}_r = \begin{pmatrix} \mathbf{L}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

is a $T \times N$ matrix and $\mathbf{L}_r = \mathbf{diag}\,(l_1, \ldots, l_r)$. The matrix $\hat{\mathbf{Z}}$ is in fact the best r-rank approximation for $\mathbf{C}$ by solving

$$\min_{\mathbf{Z}} \|\mathbf{C} - \mathbf{Z}\|_F, \text{ subject to } rank\,(\mathbf{Z}) = r.$$

Note that if $\hat{\mathbf{Z}}$ minimizes $\|\mathbf{C} - \mathbf{Z}\|_F$, it also minimizes $\|\mathbf{C} - \mathbf{Z}\|_F^2$. Therefore the above procedures for estimating $\mathbf{F}$ and $\mathbf{\Lambda}$ is equivalent to finding an optimal r-rank matrix to approximate $\mathbf{C}$.

2.2.2 Descent Algorithm

Let

$$Q\,(\mathbf{Z}, \mathbf{J}) := \frac{1}{TN}\|\mathbf{X} - \mathbf{Z} - \mathbf{J}\|_F^2 + \frac{\delta}{TN}\|\mathbf{J}\|_1.$$

We next show that under our algorithm, each iteration indeed reduces the value of $Q\,(\mathbf{Z}, \mathbf{J})$. In our algorithm, given $\mathbf{J}^{(0)}$, we find an optimal r-rank matrix $\mathbf{Z}^{(1)}$ to minimize $Q\left(\mathbf{Z}, \mathbf{J}^{(0)}\right)$ and given $\mathbf{Z}^{(1)}$ we find an optimal $\mathbf{J}^{(1)}$ to minimize $Q\left(\mathbf{Z}^{(1)}, \mathbf{J}\right)$. Given $\mathbf{J}^{(1)}$ we then find an optimal r-rank matrix $\mathbf{Z}^{(2)}$ to minimize $Q\left(\mathbf{Z}, \mathbf{J}^{(1)}\right)$ and given $\mathbf{Z}^{(2)}$ we find an optimal $\mathbf{J}^{(2)}$ to minimize $Q\left(\mathbf{Z}^{(2)}, \mathbf{J}\right)$ and so on. By induction we can get

$$\begin{aligned}Q\left(\mathbf{Z}^{(k)}, \mathbf{J}^{(k)}\right) &\geq \min_{\mathbf{z}, rank(\mathbf{z})=r} Q\left(\mathbf{Z}, \mathbf{J}^{(k)}\right) \\ &= Q\left(\mathbf{Z}^{(k+1)}, \mathbf{J}^{(k)}\right) \\ &\geq \min_{\mathbf{J}} Q\left(\mathbf{Z}^{(k+1)}, \mathbf{J}\right) \\ &= Q\left(\mathbf{Z}^{(k+1)}, \mathbf{J}^{(k+1)}\right),\end{aligned}$$

which shows that $Q\left(\mathbf{Z}^{(k)}, \mathbf{J}^{(k)}\right)$ is a decreasing function of the number of iterations k.

2.2.3 Convergence Condition

The convergence condition in step 6 only considers convergence of $\mathbf{Z}^{(k)} = \mathbf{F}^{(k)}\mathbf{\Lambda}^{(k)\mathbf{T}}$. The reason is that, if $\mathbf{Z}^{(k)}$ converges, $\mathbf{J}^{(k)}$ can also converges. To see this, we show that $\left\|\mathbf{J}^{(k+1)} - \mathbf{J}^{(k)}\right\|_F \leq \left\|\mathbf{Z}^{(k+1)} - \mathbf{Z}^{(k)}\right\|_F$. Note that at the kth iteration, the following condition holds,

$$-\left(\mathbf{X} - \mathbf{Z}^{(k)} - \mathbf{J}^{(k)}\right) + \frac{\delta}{2}\mathbf{S}^{(k)} = \mathbf{0}, \tag{8}$$

where $\mathbf{S}^{(k)} = \left(\mathbf{S}_1^{(k)}, \ldots, \mathbf{S}_T^{(k)}\right)^{\mathbf{T}}$ is a $T \times N$ matrix. The ith element in vector $\mathbf{S}_t^{(k)}$, $t = 1, \ldots, T$ is $s_{it}^{(k)} \in [-1, 1]$, $i = 1, \ldots, N$. Equation (8) is the matrix form of the KKT conditions for solving (6) and it holds for each k. It then can be shown that

$$\left\langle \mathbf{J}^{(k+1)} - \mathbf{J}^{(k)}, \mathbf{Z}^{(k+1)} - \mathbf{Z}^{(k)} \right\rangle + \left\|\mathbf{J}^{(k+1)} - \mathbf{J}^{(k)}\right\|_F^2 + \frac{\delta}{2}\left\langle \mathbf{J}^{(k+1)} - \mathbf{J}^{(k)}, \mathbf{S}^{(k+1)} - \mathbf{S}^{(k)} \right\rangle = 0,$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes $Trace\left(\mathbf{a}^{\mathbf{T}}\mathbf{b}\right)$. The third term in the above equality is trace of an $N \times N$ matrix with the ith diagonal element

$$\frac{\delta}{2}\sum_{t=1}^{T}\left(J_{it}^{(k+1)} - J_{it}^{(k)}\right)\left(s_{it}^{(k+1)} - s_{it}^{(k)}\right).$$

It is known that if $J_{it}^{(k)} \neq 0$, $s_{it}^{(k)} = sign\left(J_{it}^{(k)}\right)$ and if $J_{it}^{(k)} = 0$, $s_{it}^{(k)} \in [-1, 1]$. Thus it can be proved that $\left(J_{it}^{(k+1)} - J_{it}^{(k)}\right)\left(s_{it}^{(k+1)} - s_{it}^{(k)}\right) \geq 0$ always holds[1], and

$$\frac{\delta}{2}\left\langle \mathbf{J}^{(k+1)} - \mathbf{J}^{(k)}, \mathbf{S}^{(k+1)} - \mathbf{S}^{(k)} \right\rangle \geq 0,$$

if δ is positive. It then follows that

$$\begin{aligned}
\left\|\mathbf{J}^{(k+1)} - \mathbf{J}^{(k)}\right\|_F^2 &\leq \left\|\mathbf{J}^{(k+1)} - \mathbf{J}^{(k)}\right\|_F^2 + \frac{\delta}{2}\left\langle \mathbf{J}^{(k+1)} - \mathbf{J}^{(k)}, \mathbf{S}^{(k+1)} - \mathbf{S}^{(k)} \right\rangle \\
&= -\left\langle \mathbf{J}^{(k+1)} - \mathbf{J}^{(k)}, \mathbf{Z}^{(k+1)} - \mathbf{Z}^{(k)} \right\rangle \\
&\leq \left\|\mathbf{J}^{(k)} - \mathbf{J}^{(k+1)}\right\|_F \left\|\mathbf{Z}^{(k+1)} - \mathbf{Z}^{(k)}\right\|_F.
\end{aligned}$$

[1]Let $JS_{it} = \left(J_{it}^{(k+1)} - J_{it}^{(k)}\right)\left(s_{it}^{(k+1)} - s_{it}^{(k)}\right)$. If $J_{it}^{(k)}, J_{it}^{(k+1)} \neq 0$ and have the same sign, $JS_{it} = 0$ by $s_{it}^{(k+1)} - s_{it}^{(k)} = 0$. If $J_{it}^{(k)}, J_{it}^{(k+1)} \neq 0$ and have different same signs, $JS_{it} > 0$ since $J_{it}^{(k+1)} - J_{it}^{(k)}$ and $s_{it}^{(k+1)} - s_{it}^{(k)}$ will have the same sign. If $J_{it}^{(k+1)} = 0$ and $J_{it}^{(k)} > 0$ $\left(J_{it}^{(k)} < 0\right)$, $JS_{it} = -J_{it}^{(k)} \times ([-1, 1] - 1) \geq 0$ $\left(JS_{it} = -J_{it}^{(k)} \times ([-1, 1] + 1) \geq 0\right)$. The same logic applies to the case of $J_{it}^{(k+1)} > 0$ $\left(J_{it}^{(k+1)} < 0\right)$ and $J_{it}^{(k)} = 0$. Finally, if $J_{it}^{(k)}, J_{it}^{(k+1)} = 0$, $JS_{it} = 0$. Thus we conclude that $\left(J_{it}^{(k+1)} - J_{it}^{(k)}\right)\left(s_{it}^{(k+1)} - s_{it}^{(k)}\right) \geq 0$ holds.

Note that

$$\left\|\mathbf{J}^{(k)}-\mathbf{J}^{(k+1)}\right\|_F=\left\|\mathbf{J}^{(k+1)}-\mathbf{J}^{(k)}\right\|_F.$$

Therefore

$$\left\|\mathbf{J}^{(k+1)}-\mathbf{J}^{(k)}\right\|_F\leq\left\|\mathbf{Z}^{(k+1)}-\mathbf{Z}^{(k)}\right\|_F,$$

which means convergence of $\mathbf{Z}^{(k)}$ implies convergence of $\mathbf{J}^{(k)}$.

3 Choices of r and δ

3.1 Selecting the Number of Factors

For estimating $\mathbf{F}$ and $\mathbf{\Lambda}$, so far we assume the number of factors r is known, but such assumption in general does not hold in real data applications. When r is unknown, several methods have been proposed to consistently estimate it. These methods rely on either minimizing certain loss functions (Bai and Ng, 2002; Alessi et al., 2010) or on using test statistics constructed from eigenvalues of the (transformed) data matrix $\mathbf{X}$ (Onatski, 2009, 2010; Ahn and Horenstein, 2013). Since our algorithm separately deals with estimations of $(\mathbf{F},\mathbf{\Lambda})$ and $\mathbf{J}$, usages of these methods for consistently estimating r can be easily incorporated into the algorithm when we estimate $(\mathbf{F},\mathbf{\Lambda})$.

For simulations in Section 4 and empirical applications in Section 5, we use the IC_p criteria proposed by Bai and Ng (2002) to consistently estimate the number of common factors r. Suppose $N>T$. Let $\hat{\mathbf{F}}(r)$ be the estimated $T\times r$ factor matrix (a matrix containing $\sqrt{T}$ times the first r eigenvectors of $\hat{\mathbf{C}}\hat{\mathbf{C}}^{\mathbf{T}}$), and let

$$V\left(r,\hat{\mathbf{F}}(r)\right)=\min_{\mathbf{\Lambda}}\frac{1}{NT}\left\|\hat{\mathbf{C}}-\hat{\mathbf{F}}(r)\mathbf{\Lambda}^{\mathbf{T}}\right\|_F^2.$$

The three IC_p criteria of Bai and Ng (2002) are defined as:

$$\begin{aligned}
IC_{p^1}(r) &= \ln V\left(r,\hat{\mathbf{F}}(r)\right)+r\left(\frac{N+T}{NT}\right)\ln\left(\frac{NT}{N+T}\right),\\
IC_{p^2}(r) &= \ln V\left(r,\hat{\mathbf{F}}(r)\right)+r\left(\frac{N+T}{NT}\right)\ln\left(\min(N,T)\right),\\
IC_{p^3}(r) &= \ln V\left(r,\hat{\mathbf{F}}(r)\right)+r\left(\frac{\ln(\min(N,T))}{\min(N,T)}\right).
\end{aligned}$$

Let $\hat{r}_i=\arg\min_r IC_{p^i}(r)$, $i=1,2$ and 3. We use $\hat{r}_{BN}=\min(\hat{r}_1,\hat{r}_2,\hat{r}_3)$ as an estimate for r. For each iteration of the algorithm, we implement the method of IC_p criteria. We then use $\hat{\mathbf{F}}(\hat{r}_{BN})$, which is a matrix containing the first $\hat{r}_{BN}$ eigenvectors of $\hat{\mathbf{C}}\hat{\mathbf{C}}^{\mathbf{T}}$, and

$\hat{\mathbf{\Lambda}}\left(\hat{r}_{BN}\right) = \hat{\mathbf{C}}^{\mathbf{T}}\hat{\mathbf{F}}\left(\hat{r}_{BN}\right)/T$ as the estimated factor and factor loading matrices.

3.2 Setting the Penalty Parameter

The penalty parameter δ is important for using P-PCA method, however, there is no rule of thumb to specify it. We propose to set the penalty parameter as $\delta^{naive} = \bar{\hat{\sigma}}\sqrt{8\ln T}$, where $\bar{\hat{\sigma}} = N^{-1}\sum_{i=1}^{N}\hat{\sigma}_i$ and $\hat{\sigma}_i$ is sample standard deviation of the residuals $\hat{e}_{it}$ from using PCA method. We call the setting of δ^{naive} as a "naive" setting due to its simplicity. The naive setting, however, works well over entire simulations. We give some intuitions on why we propose such a naive setting for the penalty parameter. The idea comes from that using a softthresholding estimator with a proper setting of δ to estimate J_{it} can attain an ideal loss (Donoho and Jonhstone, 1994). Let $\omega_{it} = J_{it} + e_{it}$ denote the part of uncommon idiosyncratic component and error term in data X_{it} and assume each e_{it}, $t = 1, \ldots, T$, is i.i.d. normally distributed with mean zero and variance σ^2. Consider estimating J_{it} with either ω_{it} or 0. An ideal mean squared loss of using such an estimator over $t = 1, \ldots, T$ is given by $Loss_i^{oracle} = \sum_{t=1}^{T}\min\left(J_{it}^2, \sigma^2\right)$ when $|J_{it}| > \sigma$ is known. Without such an information, however, it can be shown that if $\tilde{J}_{it} = ST\left(\omega_{it}, \sigma\sqrt{2\ln T}\right)$ is used to estimate J_{it}, mean squared loss of $\tilde{J}_{it}$ over $t = 1, \ldots, T$ can still approach closely to $Loss_i^{oracle}$ (Donoho and Jonhstone, 1994; Wasserman, 2006, pp.172). Let $\hat{L}_{it}$ denote that X_{it} subtracts the product of the estimated common factors and factor loadings and let $\hat{J}_{it} = ST\left(\hat{L}_{it}, \sigma\sqrt{2\ln T}\right)$, which is equivalent to setting $\delta = \sigma\sqrt{8\ln T}$ in P-PCA method. Now if the common factors and factor loadings are accurately estimated, $\hat{L}_{it} \approx \omega_{it}$ and $\hat{J}_{it} \approx \tilde{J}_{it}$. Then over $t = 1, \ldots, T$, mean squared loss of $\hat{J}_{it}$ may well approximate mean squared loss of $\tilde{J}_{it}$ and therefore may approach closely to $Loss_i^{oracle}$.

In order to guarantee that the naive setting works in theory, even though some technical conditions in data generating process mentioned above should be satisfied, we find the naive setting still works well for P-PCA method over various data generating process in our simulations. In addition, due to its simplicity, using the naive setting also avoids intensive computations. Theoretically the naive setting may not be the best choice, however, it indeed serves as an easily implemented guidance and a benchmark for further adjustments to obtain the best results.

4 Simulation Results

The model for generating data for the simulations is given by (1) and (2). Except for $\mathbf{F}_t$, $\mathbf{\Lambda}$ and $\mathbf{e}_t$, we uniformly use the following settings in the data generating process:

- $T = N = 50$, 100, 200 and 400, $r = 5$.
- $J_{it} \sim i.i.d.\ Pois(\nu) \times \mathcal{N}(0, \sigma_J^2)$, $\nu = 0$, 0.01, 0.05 and 0.1, $\sigma_J = 5 \times \sqrt{\theta}$ and $\theta = r$.
- Number of columns of $\mathbf{X}$ has the idiosyncratic jump components $= \lfloor a \times N \rfloor$, $a =$ 0, 0.1, 0.5 and 1. The $\lfloor a \times N \rfloor$ columns are randomly chosen from the N columns without replacement.
- $\boldsymbol{\beta}_F = (1, \ldots, 1)$, $\dim(\boldsymbol{\beta}_F) = r \times 1$ and $\boldsymbol{\beta}_W = \mathbf{0}$.
- $\varepsilon_{t+1} \sim i.i.d.\ \mathcal{N}(0, 1)$, $t = 1, \ldots, T$.

For generating $\mathbf{F}_t$, $\mathbf{\Lambda}$ and $\mathbf{e}_t$, we consider five different models as follows.

1. **Model 1 (i.i.d. error)**:
   - $\mathbf{F}_t \sim i.i.d.\ \mathcal{N}(\mathbf{0}, \mathbf{I}_{rr})$, $\lambda_{ij} \sim i.i.d.\ \mathcal{N}(0, 1)$, $i = 1, \ldots, N$ and $j = 1, \ldots, r$.
   - $\mathbf{e}_t \sim i.i.d.\ \mathcal{N}(\mathbf{0}, \theta \times \mathbf{I}_{NN})$ and $\theta = r$.
2. **Model 2 (AR(1) error)**:
   - $\mathbf{F}_t \sim i.i.d.\ \mathcal{N}(\mathbf{0}, \mathbf{I}_{rr})$, $\lambda_{ij} \sim i.i.d.\ \mathcal{N}(0, 1)$, $i = 1, \ldots, N$ and $j = 1, \ldots, r$.
   - $e_{it} = \sqrt{\theta} u_{it}$, $u_{it} = 0.5 u_{it-1} + v_{it}$, $v_{it} \sim i.i.d. \mathcal{N}(0, 1)$, $i = 1, \ldots, N$, $t = 1, \ldots, T$ and $\theta = r$.
3. **Model 3 (Large break Model, Bates et al. (2013))**:
   - $\mathbf{F}_t \sim i.i.d.\ \mathcal{N}(\mathbf{0}, \mathbf{I}_{rr})$.
   - For the factor loading λ_{ij}, we first randomly select a subset B of i (without replacement), $i = 1, \ldots, N$ with size $4\sqrt{N}$. If $i \notin B$, we set the factor loading as $\lambda_{ij} = (0.4/0.45) \times \bar{\lambda}_{ij} \tilde{\lambda}_i$, where $\bar{\lambda}_{ij} \sim i.i.d.\ \mathcal{N}(0, 1)$ and $\tilde{\lambda}_i \sim i.i.d. U(0.1, 0.8)$. If $i \in B$, we set the factor loading as
   $$\lambda_{ij}^{LB} = \begin{cases} \lambda_{ij} & \text{for } t \leq \lfloor 0.5T \rfloor, \\ \lambda_{ij} + \triangle_j & \text{for } t > \lfloor 0.5T \rfloor, \end{cases}$$
   where $\triangle_j \sim i.i.d.\ \mathcal{N}(0, 0.16)$ for $j = 1, \ldots, r$.
   - $\mathbf{e}_t \sim i.i.d.\ \mathcal{N}(\mathbf{0}, \theta \times \mathbf{I}_{NN})$ and $\theta = r$.
4. **Model 4 (Cross sectionally correlated error)**:

- $\mathbf{F}_t \sim i.i.d.\ \mathcal{N}(\mathbf{0}, \mathbf{I}_{rr})$, $\lambda_{ij} \sim i.i.d.\ \mathcal{N}(0,1)$, $i = 1, \ldots, N$ and $j = 1, \ldots, r$.
- $e_{it} = \sqrt{\theta} u_{it}$, $u_{it} = v_{it} + \sum_{l=i-L, l\neq i}^{i+L} (0.5)^{|l-i|} v_{lt}$, $v_{it} \sim i.i.d. \mathcal{N}(0,1)$,
- $L = \max(N/20, 10)$, $i = 1, \ldots, N$, $t = 1, \ldots, T$ and $\theta = r$.

5. **Model 5 (AR(1) factor)**:

- $\lambda_{ij} \sim i.i.d.\ \mathcal{N}(0,1)$, $i = 1, \ldots, N$ and $j = 1, \ldots, r$.
- $F_{it} = 0.5 F_{it-1} + v_{f,it}$, $v_{f,it} \sim i.i.d. \mathcal{N}(0,1)$.
- $\mathbf{e}_t \sim i.i.d.\ \mathcal{N}(\mathbf{0}, \theta \times \mathbf{I}_{NN})$ and $\theta = r$.

Some concerns on using IC_ρ in the simulations should be addressed. In Lemma 2 of Amengual and Watson (2007), it was showed that when the noise-contained data $\widetilde{X}_{it} = X_{it} + w_{it}$ are used for PCA, where w_{it} is an additive error, if $(NT)^{-1} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{it}^2 = O_p\left(C_{NT}^{-2}\right)$, where $C_{NT} = \min\left(\sqrt{N}, \sqrt{T}\right)$, then information criteria IC_p proposed by Bai and Ng (2002) can still consistently estimate the number of PC's. Without the jump terms, the data in Model 3 can be viewed as such noise-contained data, and the noise w_{it} has the following form

$$w_{it} = \begin{cases} 0, & \text{if } i \notin B, \\ 0, & \text{if } i \in B \text{ and } t \leq \lfloor 0.5T \rfloor, \\ \sum_{j=1}^{r} F_{it} \triangle_j, & \text{if } i \in B \text{ and } t > \lfloor 0.5T \rfloor. \end{cases}$$

Note that if $i \in B$ and $t > \lfloor 0.5T \rfloor$, $w_{it}^2 = \left(\sum_{j=1}^{r} F_{it} \triangle_j\right)^2 = O(1)$. Thus by setting $|B| = O\left(\sqrt{N}\right)$,

$$\begin{aligned} \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{it}^2 &= \frac{1}{NT} \sum_{t=\lfloor 0.5T \rfloor + 1}^{T} \sum_{i \in B} w_{it}^2 \\ &= O_p\left(N^{-\frac{1}{2}}\right), \end{aligned}$$

which has a rate greater than $O_p\left(C_{NT}^{-2}\right)$ when $N \asymp T$. It is still unknown whether the rate $O_p\left(C_{NT}^{-2}\right)$ can be improved (Bai and Ng, 2002). Hence for Model 3, in order to obtaining a fair comparison, we will assume the number of factors is know ($\hat{r} = 5$) rather than using IC_p on estimating the number of factors.

We report four performance measures:

1. Distance correlation (DCOR): The performance measure is proposed by Szekely et al. (2007). It measures dependence between two sets of random vectors and has a range from zero to one. The higher (lower) the DCOR, the higher (lower) the dependence between the two sets of random vectors. We apply the measure to gauge dependence between the true factors $\mathbf{F}$ and estimated factors $\hat{\mathbf{F}}$. Since PCA and P-PCA can only identify the factors up to a change of sign of the true factors, using a measure of dependence between $\mathbf{F}$ and $\hat{\mathbf{F}}$ is reasonable.

2. Squared predictive error: $\left(\hat{y}_{T+1|T} - \widetilde{y}_{T+1|T}\right)^2$, where $\hat{y}_{T+1|T} = \hat{\beta}_F \hat{\mathbf{F}}_T$ and $\widetilde{y}_{T+1|T} = \widetilde{\beta}_F \mathbf{F}_T$. $\hat{\beta}_F$ $\left(\widetilde{\beta}_F\right)$ is obtained by regressing y_t onto $\hat{\mathbf{F}}_t$ (y_t onto $\mathbf{F}_t$) with the OLS.

3. Trace R^2 between the true factors $\mathbf{F}$ and estimated factors $\hat{\mathbf{F}}$ (Stock and Watson, 2002):
$$R^2_{\hat{\mathbf{F}},\mathbf{F}} = \frac{\text{avg}\left(\left\|\mathbf{F}\left(\mathbf{F}^{\mathbf{T}}\mathbf{F}\right)^{-1}\mathbf{F}^{\mathbf{T}}\hat{\mathbf{F}}\right\|_F^2\right)}{\text{avg}\left(\left\|\hat{\mathbf{F}}\right\|_F^2\right)}.$$

4. Proportions of better performances in the 2000 simulations: Proportions that P-PCA method has a higher DCOR (or a lower squared predictive error) than does PCA method in the 2000 simulations.

Figure 1 to 4 show the simulation results. In each Figure, from top to bottom are plots corresponding to Model 1 to Model 5 and from left to right are plots corresponding to different T and N. In Figure1 and Figure 2, we show averages of the DCOR between the true and estimated factors and averages of the squared predictive errors together with their 99% confidence intervals obtained from the 2000 simulations. In Figure 3 and Figure 4 we show trace R-Square and proportions that P-PCA method has a higher DCOR or a lower squared predictive error than PCA method among the 2000 simulations. In x-axis of each plot is (a, ν), where $a = 0, 0.1, 0.5$ and $\nu = 0, 0.01, 0.05$ are two parameters for controlling proportion of entries in the data matrix $\mathbf{X}$ that have nonzero idiosyncratic jump components. We sort (a, ν) in the x-axis according to the value of $a \times \nu$.

From Figure 1 to 3, in terms of the DCOR, squared predictive error and trace R-Square, we can see that P-PCA method on average outperforms PCA method in almost cases in which the nonzero idiosyncratic jump components present (both a and ν are not zeros). Furthermore, as can be seen in Figure 4, among the 2000 simulations, a large proportion of results show that when the uncommon components present, ($(a, \nu) \neq (0, 0)$), the estimated factors from P-PCA method can have a higher DCOR with the

true factors than those from PCA method. This suggests that with a high probability, P-PCA can produce more accurate estimations for the factors than PCA method. As the data have the the uncommon components present, estimated factors from P-PCA method also can have a better chance to produce a lower predictive error as T, N and $a \times \nu$ increase.

However, the performance measures vary a lot with (a, ν), T and N. Given T and N, as $a \times \nu$ increases, in general the performance measures become worse. On contrary given (a, ν), the performance measures become better as T and N increase. Except in a few cases, the patterns of the performance measures v.s. (a, ν) are consistent over different model settings. The performance measures become worse as the proportion of entries in $\mathbf{X}$ having the idiosyncratic components $a \times \nu$ becomes higher, but they become better as the sample size T and number of variables N become larger.

5 Forecasting Yearly Growths of Important Macroeconomic Indicators

In this section, we demonstrate how P-PCA method performs with real data. The first application we consider is on forecasting yearly growths of important macroeconomic indicators with common factors extracted from a number of macroeconomic variables. Two data sets of such macroeconomic variables are used. The first one contains 131 monthly macroeconomic variables of the U.S. from July-1960 to December-2011, and is a subset of the one used in Jurado et al. (2013)[2]. We will use the data set on forecasting annual growth rate of industrial production index (IP) and annual change of civilian unemployment rate (UNR) of the U.S.. The second one contains 109 quarterly macroeconomic variables of the U.S. from Q2-1960 to Q4-2008, which was used in Stock and Watson (2012)[3]. We will use the data set on forecasting annual growth rates of three macroeconomic variables of the U.S.: real GDP (RGDP), industrial production

[2] In Jurado et al. (2013), the authors used the data set on constructing what so called "uncertainty index "of macro economy by using predictive errors induced from some commonly used method on forecasting macroeconomic variables. The data set of Jurado et al. (2013) actually have 132 macroeconomic variables, but one of them seems not to have a suitable transformation form. This is why we only use 131 of them. The data set can be downloaded from Professor Sydney Ludvigson $'s website

[3] The 109 variables are a subset of 143 variables used in Stock and Watson (2012) on investigating performances of the predictive regression estimated from different statistical methods. The 109 variables are lower-level disaggregated variables and are suitable for estimation of the common factor. They can be downloaded from Professor James Stock $'s website.

index (IP) and real gross private domestic investment (RPINV) and annual change of civilian unemployment rate (UNR) of the U.S..

For estimating the common factors, we first transform the raw data to stationary time series[4]. Then the transformed data are standardized before they are used to estimate the common factors with PCA or our method. The forecasts are real time with expanding window scheme and initial window length of the scheme is set to 131 and 109 for the monthly and quarterly data sets, respectively. We detail how to implement the real time forecasts in the Appendix.

We set the penalty parameter $\delta = \delta^{naive} = \bar{\hat{\sigma}}\sqrt{8\log(T_t)}$. For estimating the number of factors r, we consider two choices. The first choice is $\hat{r}_{BN}$ as defined in Section 3.1 and in the three IC_ρ criteria, the maximum r can be chosen is restricted to 8. The second choice is simply fixing r =2, 4, 6 and 8. Note that under the real time forecast with the expanding window scheme, $\bar{\hat{\sigma}}$ and $\hat{r}$ also need to be updated when new data are included. Finally, when our method is used, we also add $\hat{SJ}_t = N^{-1}\sum_{i=1}^{N}\hat{J}_{it}$ in the predictive regressions to see whether including the information of the uncommon jumps can improve the forecasts.

For comparing performances of PCA and our methods, we report out-of-sample R^2 of their forecasts from period $T+h$ to $\bar{T}$:

$$R^2_{oos} = 1 - \frac{\sum_{t=T+h}^{\bar{T}}\left(\hat{y}_{t-h,t} - y_t\right)^2}{\sum_{t=T+h}^{\bar{T}}\left(y_t - \bar{y}\right)^2},$$

where $\hat{y}_{t-h,t}$ is the oos forecast of y_t at time $t-h$ and $\bar{y}$ is the sample mean of y_t over period $T+h$ to $\bar{T}$[5].

Table 1 shows results for using the data set of 131 monthly macroeconomic variables. For forecasting the annual growth of IP, our method (P-PCA) can have higher R^2_{oos} than PCA method. Adding the uncommon jump components as predictors is not helpful on forecasting the annual growth of IP. For the annual change of UNR, however, adding the uncommon jump components can slightly improve the forecasts when our method is used. For both forecasts, fixing r often performs better than using the IC criteria. The R^2_{oos} of forecasting the two macroeconomic variables are quite different. Using the estimated common factors results in relatively more improvements on forecasting the annual change of UNR than on forecasting the annual growth of IP.

[4]On line supplement materials of Jurado et al. (2013) and Stock and Watson (2012) provide details on how to transform the variables in each of the two data sets.

[5]For the 131 monthly macroeconomic variables, T =May-1971, h = 12 months, $\bar{T}$ =Dec-2012 and total number of the oos forecasts generated is 488. For the 109 quarterly macroeconomic variables, T =Q2-1987, h = 4 quarters, $\bar{T}$ =Q4-2009 and total number of the oos forecasts generated is 87.

Table 2 shows the results for using the data set of 109 quarterly macroeconomic variables. Our method performs better than PCA on forecasting the annual growth of IP and PRINV, and depending on the settings for estimating r, differences between their performances can be significant. For example, R^2_{oos} of our method is more than two times larger than that of PCA when r is estimated by $\hat{r}_{BN}$. As for forecasting the annual growth of RGDP and annual change of UNR, when four or six common factors are used in the forecasts, our method fails to generate higher R^2_{oos} than PCA, even though in some cases our method still performs better. The estimated common factors have better performances on forecasting annual change of UNR and annual growth of PRINV than on forecasting annul growth of RGDP and IP. Unlike in cases of using the monthly data, the uncommon jump components cannot improve the forecasts when our method is used. To sum, the results shown here suggest that P-PCA method can delivery at least comparable performances as PCA method on forecasting these macroeconomic variables.

6 Conclusion

We propose a penalized least squares estimation method, called P-PCA method, to estimate an approximate factor model in which the candidate predictors are subject to idiosyncratic large uncommon components. The proposed algorithm for the method can be easily implemented and incorporated with methods for selecting the number of common factors. Simulation results indicate that the proposed method can have better finite-sample performances than PCA method when data have the uncommon components. Empirically we first show that P-PCA method can have comparable performances as PCA method when forecasting annual growth rates of important macroeconomic variables.

Appendix

A. Constructions of Macroeconomic Data

Suppose we have data from period 1 to T, and we want to predict a variable Y_{T+h} at period $T+h$. The real time forecast proceeds as follows. We first estimate the common factors with data from period 1 to T. To estimate the predictive regression, we use the estimated factors $\hat{\mathbf{F}}_t$, $t = 1, \ldots, T-h$ as the regressors and variable Y_t, $t = 1+h, \ldots, T$ as the regressand. Let $\hat{f}_T(.)$ denote the estimated predictive regression, the real time forecast for Y_{T+h} at time T is then given by $\hat{Y}_{T,T+h} = \hat{f}_T\left(\hat{\mathbf{F}}_T\right)$, i.e., the fitted value of Y_{T+h} given $\hat{f}_T$ and $\hat{\mathbf{F}}_T$.

Forecasts in subsequent periods $T+1, \ldots,$ are obtained with an expanding window scheme, i.e., adding new data to the data used for previous predictions without deleting any of them. Or simply to say, all data from period 1 to $T+l$ are used for constructing the forecasts. For example, at period $T+l$, $l \geq 1$, the common factors are estimated with data from period 1 to $T+l$. Accordingly in the predictive regression, $\hat{\mathbf{F}}_t$, $t = 1, \ldots, T+l-h$ are the regressors and Y_t, $t = 1+h, \ldots, T+l$ are the regressand, and at time $T+l$, the real time forecast for Y_{T+l+h} is then given by $\hat{Y}_{T+l,T+l+h} = \hat{f}_{T+l}\left(\hat{\mathbf{F}}_{T+l}\right)$,

To describe how the real time forecasts are implemented, we use predicting annual growth of IP of the U.S. with the 131 monthly macroeconomic variables as an example. We first standardize the (transformed) monthly 131 macroeconomic variables from July-1960 to May-1971 (131 months which equals to our initial window length) and then use the standardized data to estimate the factors $\mathbf{F}_t$ with PCA or our method. For estimating the predictive regressionwe use the estimated $\hat{\mathbf{F}}_t$ from July-1960 to May-1970 as the regressors and the annual growth of IP of U.S. from July-1961 to May-1971 (monthly data) as the regressand. The predictive regression is estimated with the OLS. Let $\hat{\alpha}_{May-1971}$ and $\hat{\beta}_{F,May-1971}$ be the estimated intercept and coefficient vector of the predictive regression. The first forecast is for the annual growth of IP of the U.S. at May-1972, which is constructed by using $\hat{\mathbf{F}}_t$ at May-1971:

$$\hat{Y}_{May-1971,May-1972} = \hat{\alpha}_{May-1971} + \hat{\beta}_{F,May-1971}\hat{\mathbf{F}}_{May-1971}.$$

Under the expanding window scheme described above, a subsequent forecast is constructed with all data from July-1960 to the month that the forecast is used. For example, for forecast at June-1971, we expand the data used for the first forecast to include new data in June-1971 (e.g.,131 variables and Ann. growth of IP of the U.S. in June-1971) and the expanded data are used to estimate the common factors. To estimate

the predictive regression, we now use estimated factor $\hat{\mathbf{F}}_t$ from July-1960 to June-1970 as the regressors and annual growth of IP of U.S. from July-1961 to June-1971 (monthly data) as the regressand. For forecast at July-1971 and afterwards, the procedures follow in the same way.

Table 1: The Table shows out-of-Sample R^2 for predictions of annual growth of industrial production index (IP) and annual change of civilian unemployment rate (UNR) of the U.S. with common factors extracted from 131 macroeconomic variables. Monthly data of the 131 macroeconomic variables from July-1960 to December-2011 are used to extract the common factors.

	Method	IP	UNR
$r = 2$	P-PCA	**0.2517**	0.3687
	PCA	0.2415	0.3650
	P-PCA-J	0.2262	**0.3714**
$r = 4$	P-PCA	**0.2619**	**0.3714**
	PCA	0.2568	0.3675
	P-PCA-J	0.2108	0.3467
$r = 6$	P-PCA	**0.2615**	0.3780
	PCA	0.2524	0.3731
	P-PCA-J	0.2591	**0.3799**
$r = 8$	P-PCA	**0.2774**	**0.3924**
	PCA	0.2715	0.3862
	P-PCA-J	0.2760	0.3921
$r = \hat{r}_{BN}$	P-PCA	**0.2602**	0.3772
	PCA	0.2456	0.3805
	P-PCA-J	0.2584	**0.3809**

Table 2: The Table shows out-of-Sample R^2 for predictions of annual growths of three macroeconomic variables of the U.S.: real GDP (RGDP), industrial production index (IP) and real gross private domestic investment (RPINV) and annual change of civilian unemployment rate (UNR) of the U.S.. with common factors extracted from 109 macroeconomic variables. Quarterly data of the 109 macroeconomic variables from Q2-1960 to Q4-2008 are used to extract the common factors.

	Method	RGDP	IP	UNR	RPINV
$r = 2$	P-PCA	**0.0835**	**0.0800**	**0.3219**	**0.2100**
	PCA	0.0535	0.0577	0.2889	0.1635
	P-PCA-J	0.0290	0.0000	0.2544	0.1543
$r = 4$	P-PCA	0.0791	**0.1605**	0.3830	**0.1859**
	PCA	0.0888	0.1542	0.3878	0.1751
	P-PCA-J	0.0366	0.1152	0.3480	0.1455
$r = 6$	P-PCA	**0.1600**	**0.1822**	0.3834	**0.2224**
	PCA	0.1137	0.1385	0.3864	0.1776
	P-PCA-J	0.1380	0.1584	0.3614	0.1970
$r = 8$	P-PCA	**0.1778**	**0.1582**	**0.3853**	**0.2553**
	PCA	0.1081	0.1095	0.3551	0.1827
	P-PCA-J	0.1692	0.1487	0.3789	0.2323
$r = \hat{r}_{BN}$	P-PCA	**0.1295**	**0.1100**	**0.3721**	**0.2168**
	PCA	0.0718	0.0457	0.3042	0.1959
	P-PCA-J	0.0980	0.0677	0.3413	0.1818

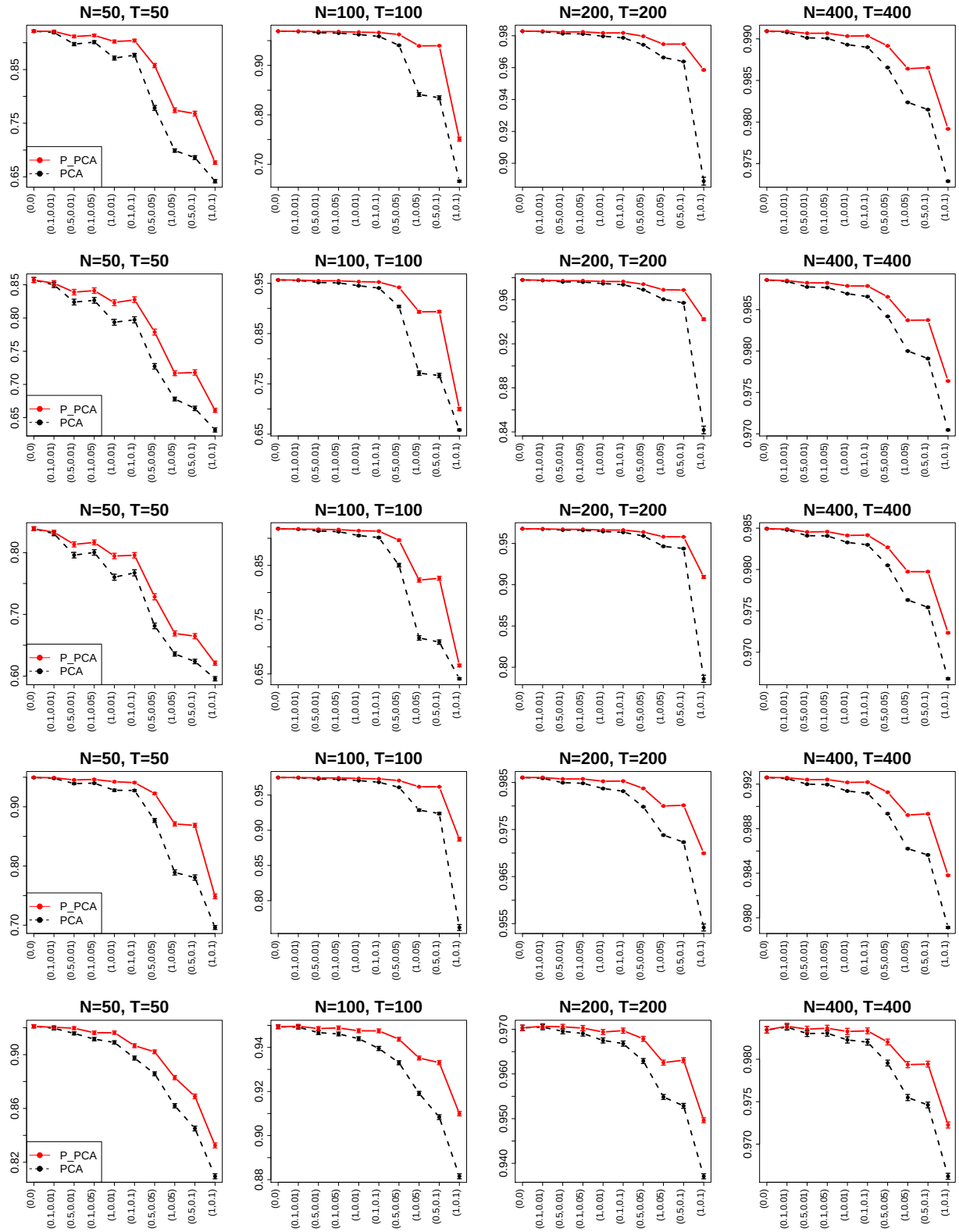

Figure 1: The Figure shows averages of Distance Correlations between the true and estimated factors (DCOR's) together with their 99% confidence intervals for each combination of (a, ν) over 2000 simulations. Plots in the first row to fifth row are corresponding to Model 1 to Model 5.

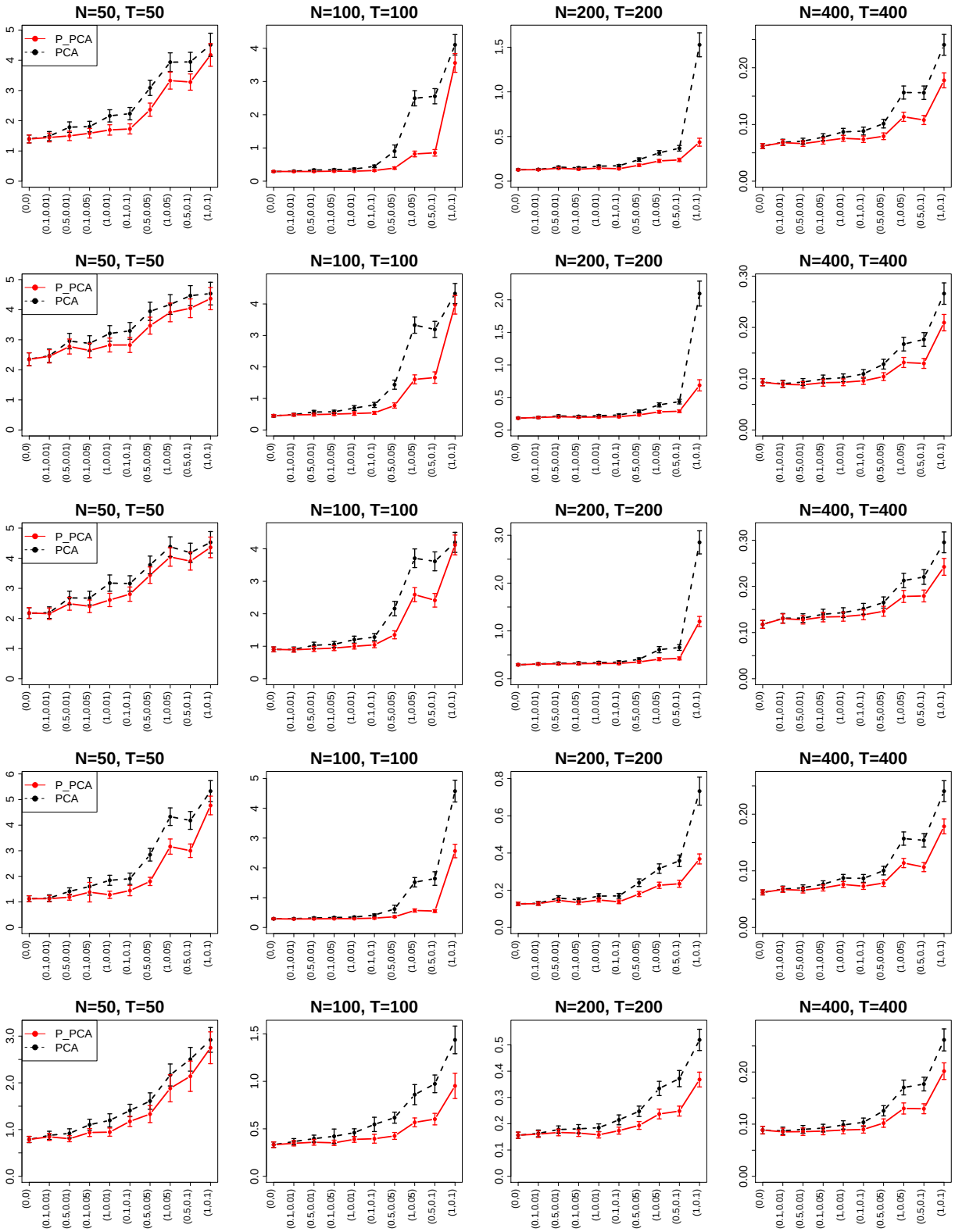

Figure 2: The Figure shows averages of squared predictive errors together with their 99% confidence intervals for each combination of (a, ν) over 2000 simulations. Plots in the first row to fifth row are corresponding to Model 1 to Model 5.

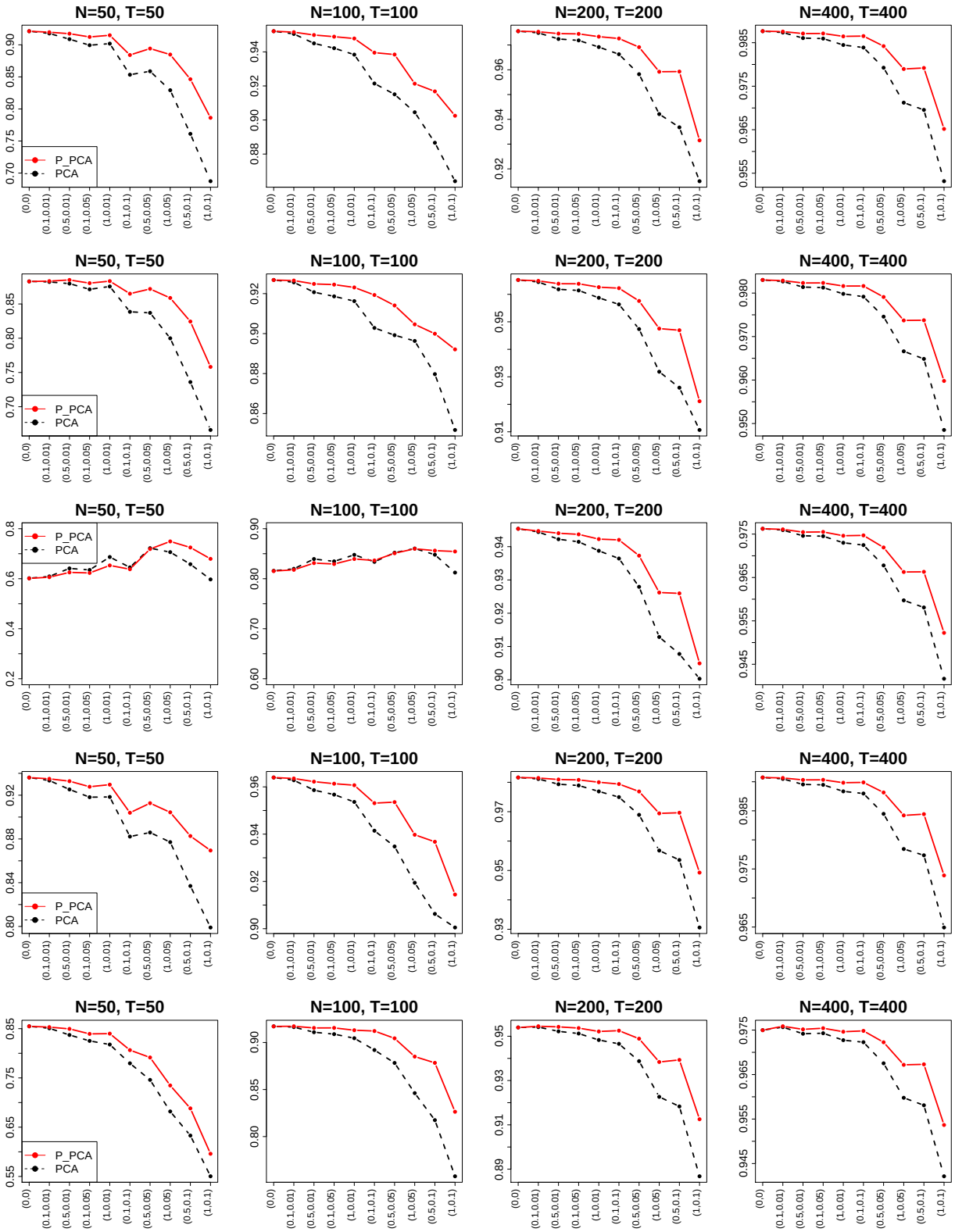

Figure 3: The Figure shows Trace R^2 between the true and estimated factors for each combination of (a, ν) from 2000 simulations. Plots in the first row to fifth row are corresponding to Model 1 to Model 5.

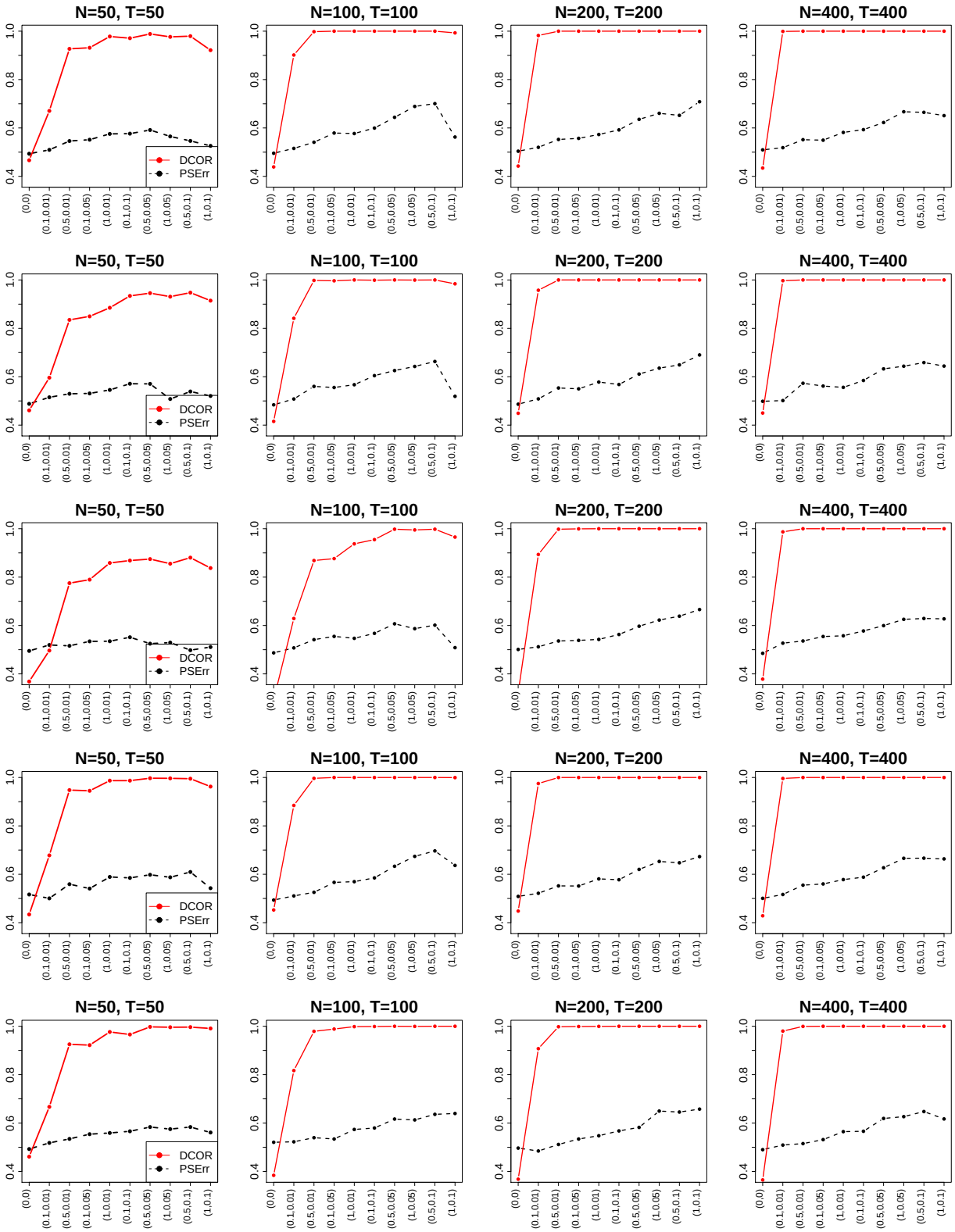

Figure 4: The Figure shows the proportion that P-PCA method has a higher Distance Correlation between the true and estimated factors (DCOR) or a lower squared predictive error than does PCA method for each combination of (a, ν) over 2000 simulations. Plots in the first row to fifth row are corresponding to Model 1 to Model 5.

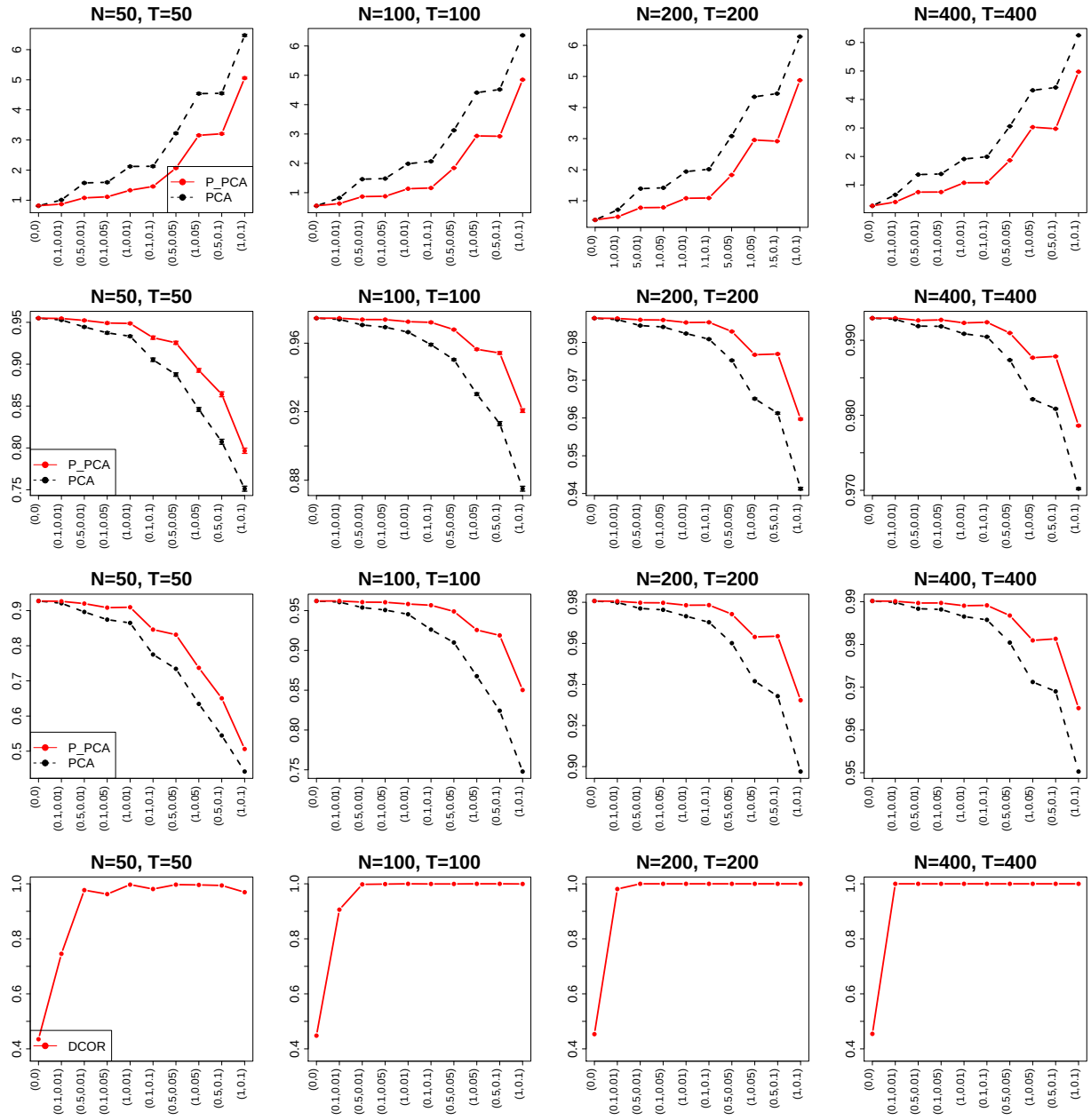

Figure 5: The Figure shows the l_2 distance between the estimated β and the oracle estimation of β (the first row), averages of Distance Correlations between the true and estimated factors (DCOR's) together with their 99% confidence intervals (the second row), Trace R^2 (the third row) and the proportion that P-PCA method has a higher DCOR than does PCA method (the fourth row) for each combination of (a, ν) over 2000 simulations. The data generating process used here is Model 6.

0.8
(0,0)
(0.1,0.01)
(0.5,0.01)
(1,0.05)
(0,0.01)
(0.1,0.1)
(0.5,0.05)
(1,0.05)
(0.5,0.1)
(1,0.1)
N=50, T=50
0.8
(0,0)
(0.1,0.01)
(0.5,0.01)
(1,0.05)
(0,0.01)
(0.1,0.1)
(0.5,0.05)
(1,0.05)
(0.5,0.1)
(1,0.1)
N=100, T=100
0.8
(0,0)
(0.1,0.01)
(0.5,0.01)
(1,0.05)
(0,0.01)
(0.1,0.1)
(0.5,0.05)
(1,0.05)
(0.5,0.1)
(1,0.1)
N=200, T=200
0.98
(0,0)
(0.1,0.01)
(0.5,0.01)
(1,0.05)
(0,0.01)
(0.1,0.1)
(0.5,0.05)
(1,0.05)
(0.5,0.1)
(1,0.1)
N=400, T=400
0.8
(0,0)
(0.1,0.01)
(0.5,0.01)
(1,0.05)
(0,0.01)
(0.1,0.1)
(0.5,0.05)
(1,0.05)
(0.5,0.1)
(1,0.1)
N=400, T=400

References

Ahn, S. C. and A. R. Horenstein (2013): "Eigenvalue Ratio Test for the Number of Factors," *Econometrica*, 81, 1203–1227.

Alessi, L., M. Barigozzi, and M. Capasso (2010): "Improved penalization for determining the number of factors in approximate factor models," *Statistics & Probability Letters*, 80, 1806–1813.

Amengual, D. and M. W. Watson (2007): "Consistent Estimation of the Number of Dynamic Factors in a Large N and T Panel," *Journal of Business & Economic Statistics*, 25, 91–96.

Ando, T. and J. Bai (2013): "Multifactor asset pricing with a large number of observable risk factors and unobservable common and group-specific factors," MPRA Paper 52785, University Library of Munich, Germany.

Ang, A., R. J. Hodrick, Y. Xing, and X. Zhang (2006): "The Cross-Section of Volatility and Expected Returns," *Journal of Finance*, 61, 259–299.

——— (2009): "High idiosyncratic volatility and low returns: International and further U.S. evidence," *Journal of Financial Economics*, 91, 1–23.

Bai, J. (2009): "Panel Data Models With Interactive Fixed Effects," *Econometrica*, 77, 1229–1279.

Bai, J. and S. Ng (2002): "Determining the Number of Factors in Approximate Factor Models," *Econometrica*, 70, 191–221.

Bali, T. G. and N. Cakici (2008): "Idiosyncratic Volatility and the Cross Section of Expected Returns," *Journal of Financial and Quantitative Analysis*, 43, 29–58.

Bates, B. J., M. Plagborg-Mller, J. H. Stock, and M. W. Watson (2013): "Consistent factor estimation in dynamic factor models with structural instability," *Journal of Econometrics*, 177, 289 – 304.

Bernanke, B., J. Boivin, and P. S. Eliasz (2005): "Measuring the Effects of Monetary Policy: A Factor-augmented Vector Autoregressive (FAVAR) Approach," *The Quarterly Journal of Economics*, 120, 387–422.

Campbell, J. Y., M. Lettau, B. G. Malkiel, and Y. Xu (2001): "Have Individual Stocks Become More Volatile? An Empirical Exploration of Idiosyncratic Risk," *The Journal of Finance*, 56, 1–43.

Candès, E. J., X. Li, Y. Ma, and J. Wright (2011): "Robust Principal Component Analysis?" *J. ACM*, 58, 11:1–11:37.

Cheng, X., Z. Liao, and F. Schorfheide (2014): "Shrinkage Estimation of High-Dimensional Factor Models with Structure Instability," NBER working paper 19792, NBER.

Corsi, F. (2009): "A Simple Approximate Long-Memory Model of Realized Volatility," *Journal of Financial Econometrics*, 7, 174–196.

Donoho, D. L. and I. M. Jonhstone (1994): "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, 81, 425–455.

Fama, E. F. and K. R. French (1992): "The Cross-Section of Expected Stock Returns," *The Journal of Finance*, 47, 427–465.

——— (1993): "Common risk factors in the returns on stocks and bonds," *Journal of Financial Economics*, 33, 3–56.

Fama, E. F. and J. D. MacBeth (1973): "Risk, Return, and Equilibrium: Empirical Tests," *Journal of Political Economy*, 81, 607–36.

Fu, F. (2009): "Idiosyncratic risk and the cross-section of expected stock returns," *Journal of Financial Economics*, 91, 24–37.

Jolliffe, I. T. (2002): *Principal Component Analysis*, Springer, second ed.

Jurado, K., S. C. Ludvigson, and S. Ng (2013): "Measuring Uncertainty," NBER Working Papers 19456, National Bureau of Economic Research, Inc.

Ludvigson, S. C. and S. Ng (2009): "Macro Factors in Bond Risk Premia," *Review of Financial Studies*, 22, p. 5027 – 5067.

Merton, R. C. (1987): "A Simple Model of Capital Market Equilibrium with Incomplete Information," *Journal of Finance*, 42, 483–510.

Moench, E., S. Ng, and S. Potter (2013): "Dynamic Hierarchical Factor Model," *The Review of Economics and Statistics*, 95, 1811–1817.

Onatski, A. (2009): "Testing Hypotheses About the Number of Factors in Large Factor Models," *Econometrica*, 77, 1447–1479.

——— (2010): "Determining the Number of Factors from Empirical Distribution of Eigenvalues," *The Review of Economics and Statistics*, 92, 1004–1016.

Stock, J. H. and M. W. Watson (2002): "Forecasting using Principal components from a Large Number of Predictors," *Journal of the American Statistical Association*, 97, p. 1167 – 1179.

——— (2012): "Generalized Shrinkage Methods for Forecasting Using Many Predictors," *Journal of Business & Economic Statistics*, 30, 481–493.

Szekely, G. J., M. L. Rizzo, and N. K. Bakirov (2007): "Measuring and testing dependence by correlation of distances," *The Annals of Statistics*, 35, 2769–2794.

Tibshirani, R. (1996): "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society, Series B*, 58, 267–288.

Wasserman, L. (2006): *All of Nonparametric Statistics*, Springer.