MS&E 349 FINAL REPORT

RANK REGULARIZED ESTIMATION OF APPROXIMATE FACTOR MODELS

Xiaoye Yuan
Institute for Computational and Mathematical Engineering
Stanford University
chamus@stanford.edu

Maoguo Shi
Institute for Computational and Mathematical Engineering
Stanford University
smgyl@stanford.edu

Carolyn Kao
Institute for Computational and Mathematical Engineering
Stanford University
chkao831@stanford.edu

March 17, 2021

1 Introduction

In economic analysis, utilizing factor models with a small group of reference variables to explain a larger number of variables is an appealing idea. For example, Capital Asset Pricing Model(CAPM), Fama-French 3-factor Model and other factor investing models are all well-known implementations in this area. It's natural to perform a Principle Component Analysis(PCA) on the data set and choose r largest eigenvalues and their corresponding eigenvectors to specify r factors for the model, since it had been shown that the space spanned by the factors can be consistently estimated by those eigenvectors.

However, this classical method developed for multi-factor model may perform poorly when both N and T $\to \infty$. One problem is that consistent estimation of Σ is not a well-defined problem. When $N > T$, the rank of $\hat{\Sigma}$ is no more than T, whereas the rank of Σ can always be N. In addition, when determining the number of factors, traditional information criteria like AIC and BIC, which ignore the effect from both the increasing cross-section dimension(N) and the increasing time-dimension(T), may result in an over-fitting problem. To solve the problem, Bai and Ng demonstrate in their paper[1] that a deterministic penalty, which is a function of both N and T, must be incorporated into the original objective function.

Although above large-dimensional factor modeling lays the foundation for the rank-regularized large-dimensional factor modeling discussed in this report, the deterministic penalties may still inflate the nominal number of factors due to the presence of weak factors or large measurement noise. To be more specific, it could be challenging to decisively separate the small from the large eigenvalues from the data, given that weak factors may only explain a small set of variables. Furthermore, one should be aware that eigenspace is sensitive to outliers even though they don't occur frequently. And this sparse spike phenomenon is usual in economic and financial data. In general, to estimate an approximate factor model, such variations should be recognized to avoid the over-estimation of number of factors.

In 2019, Bai and Ng present a new estimation for large-dimensional factor models in *Rank regularized estimation of approximate factor models*[2]. One can understand the improvements from two aspects. Firstly, the new methodology constructs the common component of the data, which is separated from the noise, with minimum rank constraint. When the data has extremely large dimension, it will make use of the iterative ridge regression technique, shrinking the model parameters to zero to estimate the regularized low rank component. Although ridge regression is biased, the reduction for variation still outweighs the extra bias introduced. Supported by [3], the iterative ridge algorithm implements Singular Value Thresholding(SVT), which shrinks and truncates the singular values of the common component. Hence it's able to deliver Robust Principal Component(RPC) results. Secondly, the new methodology modifies the objective function by implicitly adding a data dependent penalty to the original deterministic penalty. Suggested by simulations, the new information criteria now can produce more conservative estimation of the number of factors given the existence of the weak factors and outliers in the data set.

In this report, we will summary the major findings in[2], and apply the rank-regularized approximate factor models to a real-world dataset. For the theoretical part, we will first describe the problem notations and settings. Then we will present two classical estimations of approximate factor models without regularization. Next we will discuss the rank

minimization in broader machine learning literature. Finally, we will combine the arguments from the previous two parts and estimate the approximate factor model with the rank regularization.

2 Notations and problem setting

- T is the time dimension and N is the cross-section dimension. Both T and N are assumed increasing to infinite. Bai and Ng use (T, N) to denote number of rows and columns of the data matrix X in the theoretical statistical factor analysis and use (m, n) instead to denote number of rows and columns of the arbitrary matrix Z in the algorithm session.
- The Singular Value Decomposition(SVD) of Z is $Z = UDV^T$. $U = [u_1, u_2, ...u_m]$ is a $m \times m$ matrix, $V = [v_1, v_2, ..., v_n]$ is a $n \times n$ matrix, D is a $m \times n$ matrix of zeros except the first $min(m, n)$ diagonal entries.
- The nuclear norm is the sum of all singular values of Z: $||Z||_* = \sum_{i=1}^{n} d_i(Z)$. The singular values are ordered and $d_1(Z)$ is the largest.
- The component-wise 1-norm: $||Z||_1 = \sum_{i=1}^{n} |Z_{ij}|$. The F-norm: $||Z||_F = \sum_{i=1}^{n} \sum_{j=1}^{n} |Z_{ij}|^2$.
- Decompose U as the first r columns and the left: $U = [U_r; U_{m-r}]$, decompose V as the first r columns and the left: $V = [V_r; V_{n-r}]$. Then $Z = U_r D_r V_r^T + U_{m-r} D_{n-r} V_{n-r}^T$.
- Let $X_i = (X_{i1}, X_{i2}, ..., X_{iT})^T$ be a $T \times 1$ vector of random variables. The $T \times N$ data matrix is denoted as $X = (X_1, X_2, ..., X_N)$. The factor representation of the data is:

$$X = F^0 {\Lambda^0}^T + e$$

- F is a $T \times r$ matrix of common factors, Λ is a $N \times r$ matrix of factor loadings. The true values of F and Λ is F^0 and Λ^0. Let $C = F\Lambda^T$ be the common component of our interest. Notice that X is observable, but F, Λ, e are not. The population covariance of X_t is

$$\Sigma_X = \Sigma_C + \Sigma_e$$

3 Estimation of approximate factor models without rank regularization

3.1 Asymptotic Principal Components (APC)

We begin with summarizing the classical estimation of approximate factor model without rank regularization proposed by Bai and Ng in 2002[1]. The assumptions for this model are:

1. Factors and Loadings

$$E\left\|F_t^0\right\|^4 \leq M, \quad \|\Lambda_i\| \leq \bar{\Lambda}, \quad \frac{{F^0}^T F^0}{T} \xrightarrow{p} \Sigma_F > 0, \quad and \quad \frac{{\Lambda^0}^T \Lambda^0}{N} \xrightarrow{p} \Sigma_\Lambda > 0$$

   This allows the factors to be dynamic, while the loadings can be fixed or random, imposing a strong factor structure via positive definiteness of Σ_F and Σ_Λ.

2. Time and Cross-section Dependence

   (i) $E(e_{it}) = 0$, $E|e_{it}|^8 \leq M$

   (ii) $E\left(\frac{1}{N}\sum_{i=1}^{N} e_{it}e_{is}\right) = \gamma_N(s,t)$, $|\gamma_N(s,s)| \leq M$ for all s and $\frac{1}{T}\sum_{s=1}^{T}\sum_{t=1}^{T}|\gamma_N(s,t)| \leq M$

   (iii) $E(e_{it}e_{jt}) = \tau_{ij,t}$, $|\tau_{ij,t}| \leq |\tau_{ij,t}|$ for some $\tau_{ij,t}$ and for all t, and $\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}|\tau_{ij,t}| \leq M$

   (iv) $E(e_{it}e_{js}) = \tau_{ij,st}$ and $\frac{1}{NT}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{s=1}^{T}|\tau_{ij,ts}| < M$

   (v) $E\left(N^{-1/2}\sum_{i=1}^{N}\sum_{i=1}^{N}[e_{is}e_{it} - E(e_{is}e_{it})]^4\right) \leq M$ for every (t, s)

   This indicates that the errors can be serially and cross-sectionally dependent and heteroskedasticity may exist. It relaxes the constraint from the strict factor model that Σ_e should be diagonal. Instead, it requires Σ_e to be sufficiently sparse and can only be correlated weakly.

In addition, (1) and (2) collectively imply weak dependence between the factors and the errors:

$$E\left(\frac{1}{N}\sum_{i=1}^{N}\left\|\frac{1}{\sqrt{T}}\sum_{t=1}^{T}F_t^0 e_{it}\right\|^2\right)\leq M$$

3. Central Limit Theorems
   For each i and t, $\frac{1}{\sqrt{N}}\sum_{i=1}^{N}\Lambda_i^0 e_{it}\xrightarrow{d}N\left(0,\Gamma_t\right)$ as $N\to\infty$, where

$$\Gamma_t=\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}E\left(\Lambda_i\Lambda_j^T e_{it}e_{jt}\right)$$

   and $\frac{1}{\sqrt{T}}\sum_{t=1}^{T}F_t^0 e_{it}\xrightarrow{d}N\left(0,\Phi_i\right)$ as $T\to\infty$, where

$$\Phi_i=\lim_{T\to\infty}\frac{1}{T}\sum_{s=1}^{T}\sum_{t=1}^{T}E\left(F_t^0{F_s^0}^T e_{is}e_{it}\right)$$

For choosing the right number of factors r, Bai and Ng[1] show that it can be consistently estimated. The reason behind this is that determining r is a classical model selection problem given the assupmtion that we observe all potentially informative factors. The factor loadings then can be estimated by applying ordinary least squares. For the information criteria used in selecting the number of factors, we will discuss it with more details in the later two section.

Now treat r as known, the method of APC will solve the following problem:

$$\min_{F,\Lambda}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(X_{it}-\Lambda_i^T F_t\right)^2=\min_{F,\Lambda}\frac{1}{NT}\left\|X-F\Lambda^T\right\|_F^2$$

Estimated by APC, the matrix of factors is $\sqrt{T}$ times the eigenvectors corresponding to the r largest eigenvalues of $\frac{XX^T}{NT}$. Consequently, the factor loadings should satisfy $\frac{\Lambda\Lambda^T}{N}=D_r^2$, which is the diagonal matrix of the r largest eigenvalues of $\frac{XX^T}{NT}$. The results are defined as:

$$\widetilde{F}=\sqrt{T}U_r,\quad \widetilde{\Lambda}=X^T\widetilde{F}/T$$

To develop the statistical inference, Bai and Ng show that[1] as $N,T\to\infty$, $\widetilde{F}_t$ consistently estimates F_t^0 up to a rotation by the matrix $\widetilde{H}_{NT}$,

$$\min(N,T)\frac{1}{T}\sum_{t=1}^{T}\left\|\widetilde{F}_t-\widetilde{H}_{NT}F_t^0\right\|=O_p(1)$$

the newly introduced rotation matrix $\widetilde{H}_{NT}$ is defined as,

$$\widetilde{H}_{NT}=\left(\frac{{\Lambda^0}^T\Lambda^0}{N}\right)\left(\frac{{F^0}^T\widetilde{F}}{T}\right)D_r^{-2}$$

Bai and Ng further argue in[4] that $\widetilde{H}_{NT}$ wouldn't be an identity matrix mapping the j th factor estimate $\widetilde{F}_j$ to F_j^0 even asymptotically, given a violation of the assumption that $\frac{{F^0}^T F^0}{T}=I_r$ and ${\Lambda^0}^T\Lambda^0$ is diagonal. However, based on the argument that $\widetilde{F}$ consistently estimates the space spanned by F^0, if we don't require a rigorous interpretation from F factors, $\widetilde{F}$ is still a good estimation for F^0.

To summarize, the classical APC model argues that if $\sqrt{N}/T\to 0$ as $N,T\to\infty$, then plim $_{N,T\to\infty}\frac{\widetilde{F}^T F^0}{T}=\mathbb{Q}_r$, plim $_{N,T\to\infty}D_r^2=\mathbb{D}_r^2$, and

$$\sqrt{N}\left(\widetilde{F}_t-\widetilde{H}_{NT}^T F_t^0\right)\xrightarrow{d}\mathcal{N}\left(0,\mathbb{D}_r^{-2}\mathbb{Q}_r\Gamma_t\mathbb{Q}_r^T\mathbb{D}_r^{-2}\right)\equiv\mathcal{N}\left(0,\text{AVAR}\left(\widetilde{F}_t\right)\right)$$

$$\sqrt{T}\left(\widetilde{\Lambda}_i - \widetilde{H}_{NT}^{-1}\Lambda_i^0\right) \xrightarrow{d} \mathcal{N}\left(0, \left(\mathbb{Q}_r^T\right)^{-1}\Phi_i\mathbb{Q}_r^{-1}\right) \equiv \mathcal{N}\left(0, \text{AVAR}\left(\widetilde{\Lambda}_i\right)\right)$$

where AVAR stands for asymptotic variance, $\mathbb{Q}_r = \mathbb{D}_r\mathbb{V}_r\Sigma_\Lambda^{-1/2}$, and $\mathbb{D}_r^2$ and $\mathbb{V}_r$ are the eigenvalues and eigenvectors of the $r \times r$ matrix $\Sigma_\Lambda^{1/2}\Sigma_F\Sigma_\Lambda^{1/2}$.

3.2 Principal Components (PC)

Although the factor loadings estimated by APC are $\sqrt{T}$ consistent, the r largest eigenvalues of Σ_X still diverge as N increases, while the $r+1$ and the smaller eigenvalues are bounded. To regularize the rank of the common component, Bai and Ng introduce a new normalization. And the following section will discuss the estimation of the unregularized approximate factor model with the new normalization, denoted as Principle Components(PC) method.

The main difference between APC and PC is that APC works on the engienvectors of XX^T, while PC works on the singular values of X, which follows a similar logic when one compares spiked-covariance analysis and spiked-mean analysis in statistical modeling. Bai and Ng conclude in their paper that both definitions are valid even though with different normalization.

To start, PC performs a SVD on X and get $X = UDV^T$. And the singular values in D_r are of magnitude $O_p(\sqrt{NT})$. If we scale the data by $Z = \frac{X}{\sqrt{NT}}$, then the SVD of Z will be $Z = UDV^T$, the new D matrix is simply the original D matrix divided by $\sqrt{NT}$. Therefore, singular values in the normalized D_r are of magnitude $O_p(1)$. Hence we can further conclude that the largest r singular values of Z are bounded and the remaining $N-r$ singular values converge to zero as $N, T \to \infty$.

Mathematically, with the new normalization, the approximate factor model is,

$$Z = \frac{F^0}{\sqrt{T}}\frac{\Lambda^{0^T}}{\sqrt{N}} + \frac{e}{\sqrt{NT}} = F^*\Lambda^{*^T} + e^*$$

The factors and loadings estimated by PC is,

$$\widehat{F}_z = U_r D_r^{1/2}, \quad \widehat{\Lambda}_z = V_r D_r^{1/2}$$

Notice that the PC and APC estimations are mathematically equivalent up to a scale transformation. In particular, $\widehat{F}_z = \widetilde{F}\frac{D_r^{1/2}}{\sqrt{T}}$ and $\widehat{\Lambda}_z = \widetilde{\Lambda}\frac{D_r^{-1/2}}{\sqrt{N}}$. To obtain the large sample properties of the PC estimations, we still need to construct a rotation matrix. Given above relationship, it's natural to define the new rotation as $\widehat{H}_{NT} = \widetilde{H}_{NT}D_r^{1/2}$. Then we can establish the following inference (as N, T go to infinite),

$$\sqrt{N}\left(\widehat{F}_t - \widehat{H}_{NT}^T F_t^0\right) = \sqrt{N}D_r^{1/2}\left(\widetilde{F}_t - \widetilde{H}_{NT}^T F_t^0\right) \xrightarrow{d} N\left(0, \mathbb{D}_r^{1/2}\,\text{AVAR}\left(\widetilde{F}_t\right)\mathbb{D}_r^{1/2}\right) \equiv N\left(0, \text{AVAR}\left(\widehat{F}_t\right)\right)$$

$$\sqrt{T}\left(\widehat{\Lambda}_i - \widehat{H}_{NT}^{-1}\Lambda_i^0\right) = \sqrt{T}D_r^{-1/2}\left(\widetilde{\Lambda}_i - \widetilde{H}_{NT}^{-1}\Lambda_i^0\right) \xrightarrow{d} N\left(0, \mathbb{D}_r^{-1/2}\,\text{AVAR}\left(\widetilde{\Lambda}_i\right)\mathbb{D}_r^{-1/2}\right) \equiv N\left(0, \text{AVAR}\left(\widehat{\Lambda}_i\right)\right)$$

$$\left(W_{NT}\left(\widehat{C}_{it}\right)\right)^{-1/2}\left(\widehat{C}_{it} - C_{it}^0\right) \xrightarrow{d} N(0,1), \quad \text{where} \quad W_{NT}\left(\widehat{C}_{it}\right) = \frac{1}{T}\widehat{\Lambda}_i^T\,\text{AvAR}\left(\widehat{F}_t\right)\widehat{\Lambda}_i + \frac{1}{N}\widehat{F}_t^T\,\text{AvAR}\left(\widehat{\Lambda}_i\right)\widehat{F}_t$$

To conclude, the PC estimates of F_t and Λ_i are $\sqrt{N}$ and $\sqrt{T}$ consistent respectively, and the estimated common component $\widehat{C}_{it}$ is $\min(\sqrt{N}, \sqrt{T})$ consistent.

4 Rank and nuclear-norm minimization

4.1 Motivation of minimum rank

In fact, the accurate estimation of approximate factor model without rank regularization largely depends on the validity of above assumptions. When they are violated, there is high chance that the number of factors will be over-estimated. Bai and Ng explain the reasons behind from two aspects. The first is the existence of weak factors. By definition, weak factors affect only a subset of assets, thus the variation explained by weak factors could be much smaller compared

to regular factors. Failure to detect weak factors may result in lower sharpe-ratio. Back to the problem, as N goes to infinity, we're worried that the smaller eigenvalues of weak factors do not increase sufficiently fast. The second issue comes with the outliers. The outliers can enlarge the variation in the corresponding uninformative direction, which may cause the model unable to separate the r-th and (r+1)-th eignevalues.

Before introducing the notion of minimum rank, we'd like to figure out the difference between matrix spark and matrix rank. Remember that the rank of an arbitrary $m \times n$ matrix Z is the largest number of columns of Z that are linearly independent. On the contrary, the spark is the smallest k such that k columns of Z are linearly dependent. If $n \leq m$, the spark equals $n + 1$ if and only if Z has full rank n. One can compute the rank of a matrix using the number of non-zero singular values. However, the computation of spark is combinatorially hard. It's shown that if spark(Z) is not equal to $n + 1$, then it must be the case that $spark(Z) \leq rank(Z) + 1$. This issue is problematic because one can hence argue that the rank of Z may not be the size of the smallest set of factors that span Z. In this section, we're supposed to recover the actual smallest number of factors which span the target matrix Z.

What's the definition of minimum rank? Back to the original factor analysis problem, the goal is to decompose the target matrix into a communality matrix and a error matrix. Ideally, both matrices should be positive definite and the error matrix should be diagonal. The smallest rank that solves the problem will be defined as the minimum rank. However, rank minimization is a NP hard problem because the cardinality function that defines the rank is non-convex and non-differentiable. As a consequnt, below session will discuss an alternative method which solves a surrogate convex optimization problem.

4.2 Singular-Value Thresholding (SVT)

For this section, we will work on a matrix completion problem under a broader machine learning literature. Suppose we're dealing with large amounts of data, due to incomplete observations, the full matrix Z is not observable. However, the data are perfectly measured whenever they are observed. One should also take consideration of the uninformative variation in the data, also referred to as noise corruption[2].

Let Z be any $m \times n$ matrix, decomposed as $Z = L + S$. L is a matrix of minimum rank r, S is a noise matrix. The problem is to recover L with minimum rank possible from the data. Also, to minimize the effect from noise corruption, S should be sparse in nature. Bai and Ng argue that even a small number of extreme values can account for a significant amount of variation in the data. Since PCA is unable to detect whether the large variation is informative or noisy, the large noise contamination can corrupt the low rank component being identified.

Combining the rank regularization and the noise penalization, we have the following optimization problem. Notice that the sparsity is supposed to be measured by the cardinality $||S||_0$, also known as the number of non-zero entries in S. The regularization on the sparsity of the noise matrix decreases the possible occurrence of influential outlier.

$$\min_{L,S} \operatorname{rank}(L) + \bar{\gamma}\|S\|_0 \quad \text{subject to} \quad Z = L + S$$

where $\bar{\gamma} = (\max(m, n))^{-1/2}$ is a regularization parameter.

This problem is challenging because of the non convexity of the rank function and the cardinality function. [5] proved that under an incoherence on L and a cardinality condition on S, it is possible to recover both the low-rank and the sparse components exactly by solving a very convenient convex program called Principal Component Pursuit(PCP). Compared to the original problem, the non-convex constraints $\operatorname{rank}(L)$ and $\|S\|_0$ have been replaced by convex functions $\|L\|_*$, which is a sum of singular values, and $\|S\|_1$.

$$\min_{L,S} \|L\|_* + \bar{\gamma}\|S\|_1 \quad \text{subject to} \quad Z = L + S$$

The estimation of L is also referred to as Robust Principal Components(RPC).

Moreover, Zhou et al.[6] allowed the presence of small noise W in addition to the Z matrix $Z = L + S + W$. It is shown that L and S can still be recovered with high probability upon solving below convex problem. Bai and Ng argue that if $\|W\| \leq \delta$ holds for some known δ., then the RPC result is stable even if there exists small entry-wise noise, which make the problem more suitable for the factor analysis setting.

$$\min_{L,S} \|L\|_* + \bar{\gamma}\|S\|_1 \quad \text{s.t.} \quad \|Z - L - S\|_F \leq \delta$$

In terms of implementation, Singular Value Thresholding algorithm(SVT) can approximate the matrix with minimum nuclear norm among all matrices obeying a set of convex constraints. SVT is of simple first-order and easy-to-implement. The algorithm is iterative and produces a sequence of matrices at each step, mainly performs a thresholding operation

on the singular values of the target matrix. In a word, the algorithm can make use of minimal storage space and keep the computational cost of each iteration low.

Decompose a rank r matrix $Z = U_r D_r V_r^T + U_{n-r} D_{n-r} V_{n-r}^T$, define $D_r^\gamma = [D_r - \gamma I_r]_+ \equiv \max(D_r - \gamma I_r, 0)$, where γ is the target threshold. Furthermore, Cai et al.[7] shows that,

$$U_r D_r^\gamma V_r^T = \underset{L}{\operatorname{argmin}} \gamma \|L\|_* + \frac{1}{2}\|Z - L\|_F^2$$

To summarize, under the minimum risk constraint, the algorithm can produce an optimal approximation of L, $U_r D_r^\gamma V_r^T$ where D_r^γ is a matrix of thresholded singular values. As a consequence of thresholding, the rank of the regulated estimate of the low rank component can be smaller than the unregulated version.

4.3 Relation to factor models

Utilizing SVT algorithm, we can already recover the low rank component L. The problem then becomes factorizing L as a product of two rank r matrices, A and B, that is, $L = AB^T$, which will correspond to the factors and the loadings of the minimum-rank factor model in the following section.

That is to say, to combine the low-rank matrix completion problem and the factor model analysis, we should prove that,

$$\min_{A,B} \gamma \left\|AB^T\right\|_* + \frac{1}{2}\left\|Z - AB^T\right\|_F^2$$

has solution

$$\bar{A} = U_r (D_r^\gamma)^{1/2}, \quad \bar{B} = V_r (D_r^\gamma)^{1/2}$$

where $\bar{L} = \bar{A}\bar{B}^T$ should also be the solution of the previous problem.

The proof is: considering performing a SVD on the product of $AB^T = U_r D_r V_r^T$. Then by orthonormality of U_r and V_r and Cauchy-Schwarz inequality,

$$\begin{aligned}
\|L\|_* = \operatorname{trace}(D_r) &= \operatorname{trace}\left(U_r^T A B^T V_r\right) \\
&\leq \|U_r^T A\|_F \|B^T V_r\|_F \\
&\leq \|A\|_F \|B\|_F \\
&\leq \frac{1}{2}\left(\|A\|_F^2 + \|B\|_F^2\right)
\end{aligned}$$

The equality holds only when $A = U_r D_r^{1/2}$ and $B = V_r D_r^{1/2}$. Hence $\bar{L} = \bar{A}\bar{B}^T$ is the solution produced by SVT algorithm, and it can further be factorized as A matrix and B matrix. The reformulated optimization problem is,

$$\min_{A,B} \frac{1}{2}\left(\gamma \|A\|_F^2 + \gamma \|B\|_F^2 + \left\|Z - AB^T\right\|_F^2\right)$$

5 Rank constrained approximate factor models

5.1 The Rank Constrained Factor Estimates (RPCA)

In section 3, we discussed a consistent estimation of principal components of large dimensional factor modeling when cross-section dimension N and time dimension T is increasing to infinity from an econometric perspective. In section 4, we discussed a minimum rank matrix completion methodology which gives robust principal components results in a machine learning literature. In this section, we'd like to adapt the matrix completion technique to high-dimensional factor modeling.

Recall that normalizing the data matrix X by $\sqrt{NT}$ will make the largest r singular values of the normalized matrix bounded and the remaining $N-r$ singular values converge to zero. Therefore, for the rank-constrained factor estimation, we will first work on the normalized Z matrix. In addition, we will use the same assumptions specified in section 3, which put restrictions on the population singular values of the common component through moment conditions, therefore enable precise parametric convergence rate $min(\sqrt{N}, \sqrt{T})$ to be obtained.

Following the guidance of minimum rank matrix completion, after normalization, Z is consisting of three parts:

$$Z = C + e = C^* + C^- + e$$

The common component C is decomposed into 2 parts: $C = C^* + C^-$, and thresholding will be applied to it. This component will be estimated by the minimum rank matrix C^*, which is the product of the estimated factors F^* and the loadings Λ^*. After re-scaling, we can get back the true factors and loadings of our interest.

Finally, the rank-regularized factor model approximation will be,

$$\left(\bar{F}_z, \bar{\Lambda}_z\right) = \underset{F,\Lambda}{\operatorname{argmin}} \frac{1}{2}\left(\left\|Z - F\Lambda^T\right\|_F^2 + \gamma\|F\|_F^2 + \gamma\|\Lambda\|_F^2\right)$$

Define $D_r^\gamma = [D_r - \gamma I_r]_+$, the optimal solutions are,

$$\bar{F}_z = U_r \left(D_r^\gamma\right)^{1/2}, \quad \bar{\Lambda}_z = V_r \left(D_r^\gamma\right)^{1/2}$$

Multiply the normalization scalars back we can obtain,

$$\bar{F} = \sqrt{T} U_r \left(D_r^\gamma\right)^{1/2} = \sqrt{T}\bar{F}_z, \quad \bar{\Lambda} = \sqrt{N} V_r \left(D_r^\gamma\right)^{1/2} = \sqrt{N}\bar{\Lambda}_z$$

To build the rotation matrix we should first figure out the relation between the rank-restricted estimation and the un-restricted estimation,

$$\bar{F} = \widehat{F}\Delta_{NT}, \quad \bar{\Lambda} = \widehat{\Lambda}\Delta_{NT}, \quad \text{where} \quad \Delta_{NT}^2 = D_r^\gamma D_r^{-1} = \operatorname{diag}\left(\frac{(d_1 - \gamma)_+}{d_1}, \ldots, \frac{(d_r - \gamma)_+}{d_r}\right)$$

Consequently, we can introduce the rotation matrix for $\bar{F}$ by $\bar{H}_{NT} = \widehat{H}_{NT}\Delta_{NT}$, it follows that,

$$\bar{F}_t - \bar{H}_{NT}^T F_t^0 = \Delta_{NT}\left(\widehat{F}_t - \widehat{H}_{NT}^T F_t^0\right)$$

In addition, we should be cautious that the inverse of $\bar{H}_{NT}$ is not the rotation matrix for $\bar{\Lambda}$. It is otherwise defnied as,

$$\bar{G}_{NT} = \Delta_{NT}^2 \left(\bar{H}_{NT}\right)^{-1} = \Delta_{NT}\widehat{H}_{NT}^{-1}$$

Hence we have,

$$\bar{\Lambda}_i - \bar{G}_{NT}\Lambda_i^0 = \Delta_{NT}\left[\widehat{\Lambda}_i - \widehat{H}_{NT}^{-1}\Lambda_i^0\right]$$

Denote the probability limit of Δ_{NT} as $\Delta_\infty = \left(\mathbb{D}_r^\gamma \mathbb{D}_r^{-1}\right)^{1/2}$. In summary, as $N, T \to \infty$, we can obtain the following results,

$$\sqrt{N}\left(\bar{F}_t - \bar{H}_{NT}^T F_t^0\right) \xrightarrow{d} N\left(0, \Delta_\infty \operatorname{AVAR}\left(\widehat{F}_t\right)\Delta_\infty\right) \equiv N\left(0, \operatorname{AVAR}\left(\bar{F}_t\right)\right)$$

$$\sqrt{T}\left(\bar{\Lambda}_i - \bar{G}_{NT}\Lambda_i^0\right) \xrightarrow{d} N\left(0, \Delta_\infty \operatorname{AVAR}\left(\widehat{\Lambda}_i\right)\Delta_\infty\right) \equiv N\left(0, \operatorname{AVAR}\left(\bar{\Lambda}_i\right)\right)$$

implying that $\operatorname{AVAR}(\bar{F}_t) \leq \operatorname{AVAR}(\widehat{F}_t)$, and $\operatorname{AVAR}(\bar{\Lambda}_i) \leq \operatorname{AVAR}(\widehat{\Lambda}_i)$, because of the fact that the diagonal elements of Δ_∞ are less than 1. This observation is reasonable, since the shrinkage applied to the singular value matrix can be regarded as a form of regularization, which trades the variance with the bias within a specific tolerance of the deviation from the unbiased estimator.

Bai and Ng emphasize that RPC is different from Sparse Principal Component(SPC) given a specific low rank. While the thresholding operation in SPC analysis doesn't change the rank of the factor model, SVT constrains the rank of the low rank component to be at most r. As a consequence, the asymptotic variance of $\bar{F}_{jt}$ cannot exceed that of the unrestricted estimates $\widehat{F}_{jt}$ for all $j = 1, \ldots r$.

5.2 Bias and variance analysis of the estimated common component

From RPC, the estimate common component is $\bar{C} = \bar{F}\bar{\Lambda}^T$. From PC, the estimation is $\widehat{C} = \widehat{F}\widehat{\Lambda}^T$. Given above connection, we can see that

$$\bar{C} = \bar{F}\bar{\Lambda}^T = \widehat{F}\Delta_{NT}^2\widehat{\Lambda}^T \neq \widehat{F}\widehat{\Lambda}^T = \widehat{C}$$

In section 3, Bai and Ng claim that $\widehat{C}$ is an asymptotically unbiased estimate for the corresponding element of $C^0 = F^0\Lambda^{0^T}$. Hence $\bar{C}$ is a biased one. It follows that,

$$\frac{\|\bar{C}\|_F^2}{\|X\|_F^2} = \frac{\text{trace}\left(\bar{C}\bar{C}^T\right)}{\text{trace}\left(XX^T\right)} = \frac{\text{trace}\left(\left(D_r^\gamma\right)^2\right)}{\text{trace}\left(D^2\right)} < \frac{\text{trace}\left(D_r^2\right)}{\text{trace}\left(D^2\right)} = \frac{\text{trace}\left(\widehat{C}\widehat{C}^T\right)}{\text{trace}\left(XX^T\right)} = \frac{\|\widehat{C}\|_F^2}{\|X\|_F^2}$$

That is, compared to $\widehat{C}$, $\bar{C}$ can only account for a smaller fraction of the variation in X. The bias term can be quantified,

$$\gamma F_t^0 \bar{H}_{NT} D_r^{-1} \bar{H}_{NT}^{-1} \Lambda_i^0$$

Similar to the unconstrained version, with a convergence rate $m_{NT} = \min\{\sqrt{N}, \sqrt{T}\}$, the limiting distribution of the estimated common component $\bar{C}_{it}$ is,

$$\left(W_{NT}\left(\bar{C}_{it}\right)\right)^{-1/2}\left(\bar{C}_{it} - C_{it}^0 - \gamma F_t^0 \bar{H}_{NT} D_r^{-1} \bar{H}_{NT}^{-1} \Lambda_i^0\right) \xrightarrow{d} N(0,1)$$

where $W_{NT}\left(\bar{C}_{it}\right) = \frac{1}{T}\bar{\Lambda}_i^T \text{ AVAR}\left(\bar{F}_t\right)\bar{\Lambda}_i + \frac{1}{N}\bar{F}_t^T \text{ AVAR}\left(\bar{\Lambda}_i\right)\bar{F}_t$

Choosing the right threshold γ is necessary. By setting $\gamma \to 0$, the bias term will also go to zero. But we may over-estimate the dimension of the common component and the number of factors. Bai and Ng argue that we need to gurantee $\gamma \geq O\left(m_{NT}^{-1}\right)$ so that the estimated number of factors not be contaminated by the idiosyncratic errors. This is important because the largest singular value of $\frac{e}{\sqrt{NT}}$ is no less than $O_p\left(m_{NT}^{-1}\right)$.

As for the variance, since the objective function is aiming for finding a good fit subject to a minimum rank constraint, we shouldn't expect $\bar{C}$ to produce smaller mean-squared error than $\widehat{C}$ all the time. In fact, the asymptotic MSE(AMSE) may or may not be smaller than that of the unbiased estimator. Overall, the biased estimator can have smaller AMSE when the signal of the common component is weak, which can be due to a small Σ_Λ and/or a small Σ_F, or when the idiosyncratic errors have a large deviation. As a consequence, in addition to algorithmic advantages, regularized principal components analysis can help reduce MSE when the data are noisy and when the weak factors exist.

5.3 Selection of factors

As the number of factors increasing, the model can explain larger variance of the data, in the expense of model complexity, leading to a optimal choice of r which can minimize the residual of variance under a model complexity constraint. In general, we will stop including factors when its contribution to the low-rank component is small. The optimization problem of our interest is,

$$\widehat{r} = \min_{k=0,\ldots,\text{ rmax}} \widehat{IC}(k), \quad \widehat{IC}(k) = \log\left(\text{SSR}_k\right) + kg(N,T)$$

The SSR term in the information criterion function is the sum of squared residuals from fitting a model with k factors. Standardizing the input data will give $\|Z\|_F^2 = 1$, together with $\left\|\widehat{F}\Lambda^T\right\| = \|D_r\|$, then SSR_k is,

$$\text{SSR}_k = 1 - \sum_{j=1}^{k} d_j^2 = \left\|Z - \widehat{F}_k\widehat{\Lambda}_k^T\right\|_F^2$$

The second component of the information criteria is a function of both N and T. Why this form? The first advantage of this function is that it doesn't depend on the choice of rmax through the asymptotic variance of the error terms, which could be desirable in practice. The second reason is that traditional information criteria like AIC, BIC only penalize single dimension variable, which may violate the problem assumptions, hence not appropriate for all N and Ts.

$$g(N,T) = \frac{(N+T)}{NT}\log\left(\frac{NT}{N+T}\right)$$

Customized to the rank-constraint estimation, we update

$$\text{SSR}_k(\gamma) = 1 - \sum_{j=1}^{k}\left(d_j - \gamma\right)_+^2 = \left\|Z - \bar{F}_k\bar{\Lambda}_k^T\right\|_F^2$$

In summary,

$$\bar{r} = \min_{k=0,\ldots,\text{rmax}} \log\left(1 - \sum_{j=1}^{k}\left(d_j - \gamma\right)_+^2\right) + kg(N,T)$$

Comparing the objective function of the un-regularized estimates to that of the rank-regularized estimates, Bai and Ng find that the later one will have a heavier penalty. Without regularization, the contribution from the jth factor, if selected, will be d_j^2, which is larger than $(d_j - \gamma)^2$, the regularized one. In other words, the rank constraint adds a data dependent term to each factor to deliver a more conservative estimate of r. Bai and Ng conclude that this result is attractive, since the data-dependent adjustment does not require the researcher to be precise about the source of the small singular values.

5.4 Large-dimension consideration

When the Z matrix is small in dimension, $\left(\bar{F}_z, \bar{\Lambda}_z\right)$ can be directly computed from SVD of Z. But for data with high dimension, we need to deal with the tremendous memory requirement in computation. To compute the first singular vector efficiently, Bai and Ng introduce power iteration. The algorithm starts with an initial vector that has a non-zero component in the direction of the target singular vector, then recursively updates and re-normalizes the vector till convergence. The resulting vector is the largest singular vector of our interest.

Iterative Ridge Algorithm Given a $m \times n$ matrix Z, initialize a $m \times r$ matrix $F = \mathbb{U}\mathbb{D}$ where $\mathbb{U}$ is orthonormal and $\mathbb{D} = I_r$. Repeat till convergence:

(solve Λ given F): $\widetilde{\Lambda} = Z^T F \left(F^T F + \gamma I_r\right)^{-1}$

(orthogonalize): Do SVD$(\widetilde{\Lambda}) = \widetilde{\mathbb{U}}_\Lambda \widetilde{\mathbb{D}}_\Lambda \widetilde{\mathbb{V}}_\Lambda^T$, and let $\Lambda = \widetilde{\mathbb{U}}_\Lambda \widetilde{\mathbb{D}}_\Lambda$ and $\mathbb{D} = \widetilde{\mathbb{D}}_\Lambda$

(solve F given Λ): $\widetilde{F} = Z\Lambda(\Lambda^T\Lambda + \gamma I_r)^{-1}$

(orthogonalize): Do SVD$(\tilde{F}) = \widetilde{\mathbb{U}}_F \widetilde{\mathbb{D}}_T \widetilde{\mathbb{V}}_F^T$, and let $F = \widetilde{\mathbb{U}}_F \widetilde{\mathbb{D}}_F$ and $\mathbb{D} = \widetilde{\mathbb{D}}_F$

The converged result will give $\left(\bar{F}_z, \bar{\Lambda}_z\right)$, which is also the solution of the convex optimization in Section 5.1. To compute the true factors and loadings, we need to re-construct $\bar{F} = \sqrt{T}F_z$ and $\bar{\Lambda} = \sqrt{N}\Lambda_z$. The main takeway of this algorithm is that thresholding of the singular values can overcome the numerical difficulty of iterating till the eigenvalues are very small.

6 Simulation and Empirical Application

6.1 Simulation

6.1.1 Data

Before walking into the empirical studies, a quick simulation exercise is performed to illustrate the benefit of estimating the approximate factor models with minimum rank constraint. The factor model is,

$$X_{it} = {F_t^0}^T \Lambda_i^0 + e_{it} + s_{it}$$

Data are simulated following below assumptions,

$$F_t^0 \sim \mathcal{N}(0,\, I_r), \quad \Lambda_i^0 \sim \mathcal{N}(0,\, I_r), \quad e_{it} \sim \mathcal{N}(0,\, 1)$$

To construct the sparse noise matrix S, let $s_{it} \sim N\left(0, \sigma^2\right)$ if $(i,t) \in \Omega$. Ω is an index set containing (i,t) positions with non-zero values of s_{it}. Otherwise, $s_{it} = 0$. We further assume that $P(i \in \Omega) = 0.1$ and $P(t \in \Phi) = 0.03$. To get the results of different situations, we have $N \in \{50, 100\}$, $T \in \{100, 200, 400\}$, $\sigma \in \{5, 10, 15\}$.

As for the number of factors, we choose $r = 5$, which specifies the dimensions of the real factors. Then we construct factor models using asymptotic principle components and robust principal components respectively, and figure out the corresponding estimate of the number of factors. When solving these two problems, we set $rmax = 8$ and let the threshold of singular values $\gamma = 0.05$.

6.1.2 Approach

We utilize `produce_X` to generate the data matrix X.

```
def produce_X(N, T, w, r):
  X = np.zeros((N, T))
  F = np.zeros((T, r))
  Gamma = np.zeros((N, r))
  a = np.random.choice([0, 1], size = N,  p=[0.9, 0.1])
  b = np.random.choice([0, 1], size = T, p=[0.97, 0.03])
  for t in range(T):
    F[t] = np.random.multivariate_normal(np.zeros(r), np.diag(np.ones(r)))
  for i in range(N):
    Gamma[i] = np.random.multivariate_normal(np.zeros(r), np.diag(np.ones(r)))
  for i in range(N):
    for t in range(T):
      X[i][t] = np.dot(F[t], Gamma[i])+ np.random.normal(0, 1)+ a[i]*b[t]* np.random.normal(0, w)
  return X
```

In the function `get_r`, we solve the convex optimization problem without rank minimization constraint.

```
def get_r(X, N, T, rmax):
  Z = X / np.linalg.norm(X)
  u, d, v = np.linalg.svd(Z, full_matrices=True)
  IC = np.zeros(rmax + 1)
  for i in range(rmax + 1):
    D_r = np.diag(d[:i])
    u_r = u[ :, : i]
    v_r = v[ : i, :]
    ssr = 1- np.dot(d[:i], d[:i])
    IC[i] = np.log(ssr ) + i * (N + T) * np.log(N * T / (N + T)) / (N * T)
  return np.argmin(IC)
```

Then, in the function `get_r_gam`, we complete singular value thresholding and solve the surrogate convex rank-constraint minimization problem.

One thing we should notice that in the theoretical part, we let $Z = X/\sqrt{NT}$ and $\|Z\|_F = 1$ because we suppose that each series in standardized. But in real dataset, if $Z = X/\sqrt{NT}$, then $\|Z\|_F = 1$ may not be true and this may affect the implementation of RPCA method. Hence, we let $Z = X/\|X\|_F$ in our implementation.

```
def get_r_gam(X, N, T, rmax):
  Z = X / np.linalg.norm(X)
  u, d, v = np.linalg.svd(Z, full_matrices=True)
  IC = np.zeros(rmax + 1)
  for i in range(rmax + 1):
    d_gam = np.zeros(i)
    for m in range(i):
      d_gam[m] = np.maximum(d[m] - gam, 0)
    D_r = np.diag(d_gam)
    u_r = u[ :, : i]
    v_r = v[ : i, :]
    ssr = 1 - np.dot(d_gam, d_gam)
    IC[i] = np.log(ssr) + i * (N + T) * np.log(N * T / (N + T)) / (N * T)
  return np.argmin(IC)
```

Finally, we apply different values of N, T, σ and calculate the result of APC estimate r_1 and RPCA estimate r_2.

```
for N in [50, 100]:
  for T in [100, 200, 400]:
    for w in [5, 10, 15]:
      lst1 = np.zeros(100)
      lst2 = np.zeros(100)
      for p in range(100):
        X = produce_X(N, T, w, 5)
        my_r1 = get_r(X, N, T, 8)
        my_r2 = get_r_gam(X, N, T, 8)
        lst1[p] = my_r1
        lst2[p] = my_r2
      print("N = ", N, "T = ", T, "w = ", w, "r1 is", np.mean(lst1), "r2 is", np.mean(lst2))
```

6.1.3 Result

N	T	σ	APC	RPC
50	100	5	5.17	5.0
50	100	10	6.49	4.99
50	100	15	6.83	5.12
50	200	5	5.02	5.0
50	200	10	6.48	5.0
50	200	15	7.34	5.19
50	400	5	5.0	4.99
50	400	10	6.69	4.99
50	400	15	7.57	5.07
100	100	5	5.17	5.0
100	100	10	7.06	5.01
100	100	15	7.18	5.53
100	200	5	5.06	5.0
100	200	10	7.26	5.0
100	200	15	7.78	5.22
100	400	5	5.0	5.0
100	400	10	7.24	5.0
100	400	15	7.91	5.06

Table 1: Simulation results of APC and RPC

Since dimensions are not too large, we performed direct SVD instead of iterative ridge algorithm in the simulation part. Our fast simulation results are presented in above table. With different choices of N, T and the standard deviation, we can see that the estimated robust PC produces smaller number of factors, compared to the estimated asymptotic PC, which is also closer to the true number of factors predefined in the simulation setup step.

6.2 Empirical Application

6.2.1 Dataset Description

FRED-MD and FRED-QD are large macroeconomic databases designed for the empirical analysis of "big data." The datasets of monthly and quarterly observations mimic the coverage of datasets already used in the literature, but they add three appealing features. They are updated in real-time through the FRED database. They are publicly accessible, facilitating the replication of empirical work. And they relieve the researcher of the task of incorporating data changes and revisions.

In the first empirical application, we use **FRED-MD** as our dataset. It is a large, monthly frequency, macroeconomic database containing a panel of 134 monthly US macroeconomic series spanning the period 1960-2016. Amongst all 134 factor series, it is illustrated in [8] that by organizing those factors into groups of 8 that illustrate the explanatory power the top eight factors, accordingly *output and income; labor market; consumption and housing; orders and inventories;*

money and credit; interest rate and exchange rates; price and *stock market*, it sufficiently explains 0.476 of the total variation in all series as a whole.

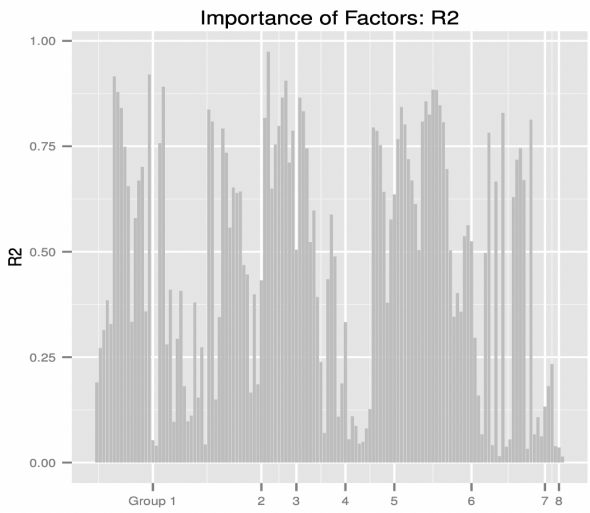

Figure 1: Importance of Factors in Dataset FRED-MD [8]

In the second empirical application, we use **FRED-QD** as our dataset. Similarly to the previous dataset, is a database that consists of 248 macroeconomic time series, in a quarterly basis. To be specific, the series could be roughly classified into 14 groups: *NIPA; Industrial Production; Employment and Unemployment; Housing; Inventories, Orders, and Sales; Prices; Earnings and Productivity; Interest Rates; Money and Credit; Household Balance Sheets; Exchange Rates; Other; Stock Markets* and *Non-Household Balance Sheets.*

Our choice of the two databases results from the fact that both share comparable information content to various vintages of so-called Stock-Watson datasets [9]. James Stock of Harvard and Mark Watson of Princeton created a macroeconomic data set that has become the benchmark for a lot of what people do in economics when they are working with big data. Those extracted factor estimates take significant part in forecasting a wide range of macroeconomic series.

6.2.2 Data Preprocessing and Implementation

The raw panel is unbalanced, so we turn both datasets into a balanced panel by manually removing particular series. For example, in the `2015:04` vintage of FRED-MD, 9 series have observations missing at the beginning of the sample, and 12 series have observations missing at the end of the sample. Dropping series of 64, 66, 101 and 130, a balanced panel of 130 series dating from `1960:1` to `2014:12` can then be formed in such a way. Also, the raw data has a column of dates and a row of features of the data. We need to drop them while implementing data preprocessing.

```
data = pd.read_csv("2016-08.csv")
missing_data = data.columns[data.isna().any()].tolist()
a_data = data.drop(columns=missing_data)
b_data = a_data.drop(labels="sasdate", axis=1)
final_data = b_data.drop(list(range(1,13)))
```

Then we can calculate the RPCA estimate of the number of factors.

```
npdata = np.transpose(final_data.to_numpy())
rq = get_r_gam(npdata, 107, 677, 30)
print("the RPCA estimate of dimension of factors of FRED_MD is", rq)
```

The data preprocessing and implementation of FRED-QD is almost the same.

```
data2_raw = pd.read_csv("current.csv")
data2 = data2_raw.drop(746)
missing_data2 = data2.columns[data2.isna().any()].tolist()
a_data2 = data2.drop(columns=missing_data2)
b_data2 = a_data2.drop(labels="sasdate", axis=1)
final_data2 = b_data2.drop(list(range(1,13)))
npdata2 = np.transpose(final_data2.to_numpy())
rq2 = get_r_gam(npdata2, 105, 734, 15)
print("the RPCA estimate of dimension of factors of FRED_QD is", rq2)
```

6.2.3 Conclusion

On FRED-MD, we get an estimate of eight factors in the previous APC result. But after regularization, we find three factors.

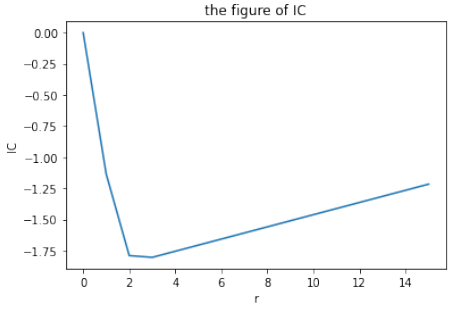

Figure 2: The figure of IC while implementing RPCA on FRED-MD

On FRED-QD, we get an estimate of four factors in the previous APC result. But after regularization, we find two factors.

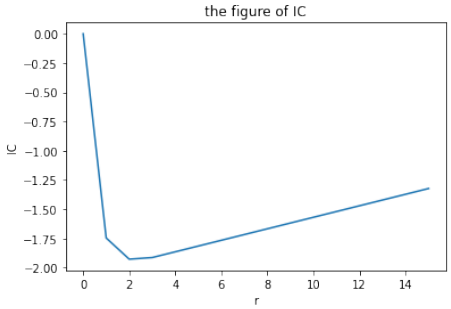

Figure 3: The figure of IC while implementing RPCA on FRED-QD

While the dimension of the dataset is large, we can still perform singular value decomposition efficiently, as you can tell that the dimensions of real economic dataset are similar to the simulation set up. For even larger dimension, iterative ridge algorithm will be necessary.

In summary, we find that while trying to estimate the number of factors with regard to macroeconomic data, the RPCA estimate is significantly lower than the APC result. On the one hand, this can prove the existence of the sparse noise as well as the large variation in the macroeconomic dataset, and a factor model with smaller set of features can be an alternative solution. On the other hand, it's infeasible to construct a precise factor model with strong interpret ability, since we rotate the factor matrix for large sample properties. Overall, RCPA is efficient in providing more conservative

estimates when the strong factor assumption is questionable. However, because that we mostly work with economic data in this project, the evaluation metric and explanation power of the estimate approximate model is limited. To improve in the future, when working with specific data set, one can develop more efficient and inclusive evaluation methodology customized for the nature of the data, e.g. the Sharpe ratio for any portfolio strategy.

References

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

[2] Jushan Bai and Serena Ng. Rank regularized estimation of approximate factor models. *Journal of Econometrics*, 212(1):78–96, 2019.

[3] T. Hastie, R. Mazumder, J.D. Lee, and R. Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *Journal of Machine Learning Research*, 16:3367–3402, 2015. cited By 107.

[4] Jushan Bai and Serena Ng. Principal components estimation and identification of static factors. *Journal of Econometrics*, 176(1):18–29, 2013.

[5] Emmanuel Candes, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58, 12 2009.

[6] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma. Stable principal component pursuit. In *2010 IEEE International Symposium on Information Theory*, pages 1518–1522, 2010.

[7] J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20:1956–1982, 2010.

[8] Michael W. McCracken and Serena Ng. Fred-md: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, 2016.

[9] James Stock and Mark Watson. Evidence on structural instability in macroeconomic time series relations. *Journal of Business Economic Statistics*, 14(1):11–30, 1996.