Term Structure of Firm Characteristics and Multi-Horizon Investment[*]

Svetlana Bryzgalova
London Business School

Serhiy Kozak
University of Maryland

Markus Pelger
Stanford University

Ye Ye
Stanford University

May 16, 2022

Abstract

...

[*] Bryzgalova: xxx@london.edu; Kozak: sekozak@umd.edu; Pelger: xxx@stanford.edu; Ye: xxx@stanford.edu.
We appreciate helpful comments from ...

1 Introduction

1. Empirical asset pricing routinely works with portfolios formed on most recent observations of firm's characteristics. The history of characteristics and how past information in characteristics gets reflected in discount rates is thus ignored. In this paper we study the term structure of characteristic-sorted portfolios. Specifically, look at monthly expected excess return of portfolio i at time t sorted on characteristics $C_{i,t-l}$: $\mathbb{E}\left[r_{t,i}|C_{t-l}\right]$ for some lags $l = 1..L$.

2. We propose a novel "tensor factor model" which provides a parsimonious framework for the term structure of stock returns for multiple stocks and lags. At the heart of the model is the idea of collecting contemporaneous returns on stocks at various lags, $r_{t,i,m} = r_{t,i}|C_{t-m}$, into a three-dimensional *tensor* $\mathcal{R}$ of size $T \times N \times M$ and find its low-rank approximation by extracting factors and loadings which succinctly the dynamics of all returns within the tensor. This approximation, known as a "tensor decomposition" is a conceptual generalization of PCA to the three-dimensional settings.

3. The idea of looking for a factor structure in monthly contemporaneous returns, rather than returns at mixed frequencies and/or at different points in time has multiple benefits. First, because returns are weakly correlated in the time-series, focusing on all returns measured at the same time allows us to find strong factor structure and thus explain most of the variation in returns. Second, this approach allows us to convert a fundamentally time-series multi-horizon predictability problem into a cross-sectional single-horizon problem. Third, because we only look at high-frequency returns (monthly), we can use time-series observations relatively more efficiently compared to approaches that focus on long-horizon returns. Yet, since we model monthly returns at all lags, we can aggregate them up across arbitrary number of time-periods to arrive at a respective long-horizon return. We, therefore, effectively decompose a long-run return into a sum of "forward" returns, and focus on modeling these components of long-run returns.

4. Tensor factor model is given by

$$r_{t,i,l} = \sum_{k=1}^{K} f_{t,k} \cdot b_{i,k} \cdot g_{l,k},$$

where $f_{t,k}$ are time-series of returns on a factor k, $b_{i,k} \cdot g_{l,k}$ is a loading of a portfolio i based on lag-l characteristic on this factor. These loadings are further decompose into cross-sectional components (across stocks), $b_{i,k}$, and lag components, $g_{l,k}$. PCA is a special case of this decomposition when there is no lag dimension.

5. Note that fe can define the "lag factors" as

$$f_{l,t,k} = f_{t,k} \cdot g_{l,k}.$$

These "lag factors" factors themselves possess a term-structure which varies across different characteristic lags l as given by a simple scaling of the time-series factors f by a lag- and factor-specific constant $g_{l,k}$. Each asset loads on K such factors through its own collection of factor loadings $b_{i,k}$. Therefore, a term structure of every assets simply reflects a linear combination of term structures of the underlying "lag factors".

6. We use the model to study term structure of expected returns and MVE portfolio weights / SDF loadings across multiple horizons.

7. Why is the tensor model useful? First, It provides a low-dimensional representation of the data, which could be helpful both for economic interpretation and as a regularization for better OOS performance. Second, the model exploits an APT type logic to focus on explaining variation in returns (second moments), which we can estimate relatively well at higher frequencies. Then, by no-arbitrage restrictions in the model, this decomposition allows us to make predictions about first moments at lower frequencies without the need of measuring them directly (assuming stationarity and constancy of ER).

8. We explore the term structures of individual portfolios as well as factors. We check how well a low-rank tensor model can approximate these patterns in the data.

9. We compare the model to a naive approach which measures returns on characteristic-sorted buy-and-hold portfolio returns over longer horizons. We focus both on expected returns, across many portfolios, as well as the performance of the MVE portfolio which combines these returns.

1. Empirical asset pricing uses characteristics as pricing signals. How fast are these signals dying out? Equivalent question, what are the asset pricing results for different holding periods.

2. Most empirical asset pricing select the signal lag rather arbitrarily. Each lag of characteristics can be interpreted as new characteristic. This paper provides a comprehensive study the effect of characteristic lags.

3. Fundamentally, we study the term structure of expected returns. This term structure provides the risk loadings, risk premia and SDF weights for different horizons.

4. Large dimensional problem: many characteristics and large number of lags. Challenging. We solve it with a novel three dimensional tensor approach.

5. Three dimensional factor models generalize the conventional two-dimensional factor models by adding the horizon dimension. Our factor is latent, i.e. generalization of PCA. Intuitively, start with very large cross-section of characteristic and their lagged sorted portfolios. A simple PCA, treats the combination of lags and characteristics as new additional characteristics. Our tensor approach imposes structure on the loadings of latent factors and decomposes them into a cross-sectional characteristics and horizon component.

6. Tensor model provides parsimonieous

2 Methodology

2.1 Tensors

This paper studies the term structure of firm characteristics and provides a parsimonious tensor factor model solution. It provides answers to the following research questions:

1. What is the term structure of risk premia of firm characteristics? More specifically how does the risk premium of characteristics depend on their dynamics. We provide evidence that the risk premium is not Markovian in firm characteristics and depends on the history of lagged characteristics.

2. What is a parsimonious model for the term structure? We show that the term structure does not require more factors, but is captured by conditional risk exposure. We introduce a novel tensor factor model to capture this third dynamic dimension. Importantly, the loadings on risk factors for lagged characteristic portfolios follow a low dimensional structure which is captured by the tensor.

The goal of this draft is to establish a unified notation, research question and an outline of the results for the paper. Our main analysis will be based on the five characteristics size, value, profitability, investment and momentum.

We study excess returns, i.e. returns minus risk-free rate, for a three-dimensional array of returns:

$$R_{t,i,l} \quad t = 1, ..., T, i = 1, ..., N \text{ and } l = 1, ..., L$$

where T is the time, N the cross-sectional and L the lag dimension. In our empirics we consider $L = 60$ lags. We denote by

$$\underbrace{R_l}_{T\times N} \qquad \text{for } l = 1, ..., L$$

the panels of returns using the lag l. This means that R_1 is the panel which is conventionally used. We can treat the lagged characteristics as characteristics on their own and simply increase the cross-sectional dimension. This results in a particular stacking of the tensor:

$$\underbrace{\vec{R}}_{T\times NL} = \begin{pmatrix} R_1 & R_2 & \cdots & R_L \end{pmatrix}.$$

We will study the risk premia of the tensor R with various factor models.

1. **T-PCA**: Our novel benchmark model is a tensor factor model. It imposes the low rank tensor structure

$$R_{t,i,l} = \sum_{k=1}^{K} \lambda_k F_{t,k} B_{i,k} W_{l,k}$$

or in tensor notation

$$R = \sum_{k=1}^{K} \lambda_k \cdot F \circ B_k \circ W_k \qquad \text{for } F \in \mathbb{R}^{T\times K},\ B \in \mathbb{R}^{N\times K} \text{ and } W \in \mathbb{R}^{L\times K}$$

The normalizing scalar λ_k is similar to singular values. In the following I drop this scalar from the notation, i.e. it is implicitly subsumed by the factors. Note that this model can also be expressed as

$$\overrightarrow{R} = F \underbrace{(W \odot B)^\top}_{\Lambda^\top}$$

where $\odot$ is the Katri-Rao, or "matching columnwise" Kronecker product. This notation shows that this is a more restricted version of the PCA model presented next, but that the lower dimensional structure of Λ cannot be inferred by simple eigenvalue decompositions. An equivalent representation is

$$R_l = FD^{(l)}B^\top \qquad \text{with } D^{(l)} = \text{diag}(W_{l,:})$$

or equivalently

$$\overrightarrow{R} = F \begin{pmatrix} D^{(1)}B^\top & D^{(2)}B^\top & \cdots & D^{(L)}B^\top \end{pmatrix} = F\Lambda^\top.$$

We estimate the tensor factor model with the PARAFAC algorithm which is based on iterative regressions in each of the three dimensions.

Importantly, we normalize the first lag tensor to be constant, i.e. we set $W_1 = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^\top$. The corresponding factors F_1 and loadings B_1 are estimated similar to the PARAFAC algorithm with iterated regressions.

2. **PCA**: The PCA estimator interprets the lagged characteristics as new cross-sectional observations and applies conventional PCA to $\overrightarrow{R}$ to obtain

$$\overrightarrow{R} = F^{\text{PCA}} \underbrace{\Lambda^{\text{PCA}\top}}_{K\times NL}$$

Hence, the tensor model is a restricted version of PCA.

3. **T-FF6**: Similar to the latent tensor factor model, we estimate a tensor version for the Fama-French 5 factors + momentum. In this case we take the factor time-series as given, but estimate $B^{\text{T-FF6}}$ and $W^{\text{T-FF6}}$ with iterated regressions similar to the PARAFAC algorithm.

4. **FF6**: The FF6 (Fama-French 5 factors + momentum) is the counterpart of the PCA estimator. It is simply based on a regression of the six factors on the panel $\overrightarrow{R}$. Hence, T-FF6 is a special case of FF6.

Our main analysis is on the excess returns R, but it will be insightful to consider lag-excess returns R^{lex} and 1-lag returns $R^{\text{1-lag}}$ for evaluation. The lag-excess returns are returns in excess of the 1-lag returns:

$$R^{\text{lex}}_{t,i,l} = R_{t,i,l} - R_{t,i,1}.$$

Similarly, we can define the matrices

$$R^{\text{lex}}_{l} = R_l - R_1 \qquad \overrightarrow{R}^{\text{lex}} = \overrightarrow{R} - \begin{pmatrix} R_1 & \cdots R_1 \end{pmatrix}.$$

The 1-lag returns are simply R_1.

Our main metrics are the Sharpe ratio, cross-sectional pricing errors and unexplained variation. More specifically, we denote by SR the annualized Sharpe ratio of the unconditional mean-variance efficient portfolio based on different factors. Each factor model implies the return based on the factors. The residuals are the difference between the returns and the factor model implied return. For the tensor model this is

$$\epsilon_{t,i,l} = R_{t,i,l} - \sum_{k=1}^{K} \lambda_k F_{t,k} B_{i,k} W_{l,k}$$

and similarly for the other factor models. We also decompose the residuals into their lag-excess return and 1-lag component:

$$\begin{aligned}\epsilon^{\text{lex}}_{t,i,l} &= R_{t,i,l} - R_{t,i,1} - \left(\sum_{k=1}^{K} \lambda_k F_{t,k} B_{i,k} W_{l,k} - \sum_{k=1}^{K} \lambda_k F_{t,k} B_{i,k} W_{1,k} \right) \\ &= R^{\text{lex}}_{t,i,l} - \sum_{k=1}^{K} \lambda_k F_{t,k} B_{i,k} \left(W_{l,k} - W_{1,k} \right)\end{aligned}$$

and similarly for the other models. The 1-lag residuals are simply based on

$$\epsilon_1 = R_1 - \sum_{k=1}^{K} \lambda_k F_{t,k} B_{i,k} W_{1,k}$$

Pricing errors are obtained from time-series means

$$\alpha_{i,l} = \frac{1}{T} \sum_{t=1}^{T} \epsilon_{t,i,l} \qquad \alpha^{\text{lex}}_{i,l} = \frac{1}{T} \sum_{t=1}^{T} \epsilon^{\text{lex}}_{t,i,l}.$$

The unexplained variance is the time-series variance of the residuals:

$$\sigma^2_{\epsilon,i,l} = \frac{1}{T} \sum_{t=1}^{T} \epsilon^2_{t,i,l} - \alpha^2_{i,l} \qquad \sigma^2_{\epsilon^{\text{lex}},i,l} = \frac{1}{T} \sum_{t=1}^{T} {\epsilon^{\text{lex}}_{t,i,l}}^2 - {\alpha^{\text{lex}}_{i,l}}^2.$$

We will report various normalized averages:

$$
\begin{aligned}
\text{XS-}\alpha &= \sqrt{\frac{1}{NL}\sum_{i=1}^{N}\sum_{l=1}^{L}\alpha_{i,l}^{2}} \\
\text{XS-}\alpha_i^{\text{lex}} &= \sqrt{\frac{1}{L}\sum_{l=1}^{L}\alpha_{i,l}^{\text{lex}^2}} \\
\text{XS-}\alpha_i^{\text{lex}} &= \sqrt{\frac{1}{L}\sum_{l=1}^{L}\alpha_{i,l}^{\text{lex}^2}} \\
\sigma_\epsilon &= \sqrt{\frac{1}{NL}\sum_{i=1}^{N}\sum_{l=1}^{L}\sigma_{\epsilon,i,l}^{2}} \Bigg/ \sqrt{\frac{1}{NL}\sum_{i=1}^{N}\sum_{l=1}^{L}\text{Var}(R_{t,i,l})} \\
\sigma_\epsilon^{\text{lex}} &= \sqrt{\frac{1}{NL}\sum_{i=1}^{N}\sum_{l=1}^{L}\sigma_{\epsilon,i,l}^{\text{lex}^2}} \Bigg/ \sqrt{\frac{1}{NL}\sum_{i=1}^{N}\sum_{l=1}^{L}\text{Var}(R_{t,i,l}^{\text{lex}})}
\end{aligned}
$$

Note that all our models allow us to calculate the above quantities.

We will estimate those quantities in-sample and out-of-sample. In the case of the out-of-sample analysis we use a rolling window to estimate Λ respectively B and W and use those for an out-of-sample projection to obtain the factor time-series F. A rolling window of 120 months seems to be appropriate, but I think Ye is in the best position to make this decision. If the computation becomes too time-consuming, we could also update the loadings every 12 months using the past 120 months (or a similar coarser frequency).

2.2 Multi-Horizon Perspective

Recall, that we study excess returns, i.e. returns minus risk-free rate, for a three-dimensional array of returns:

$$R_{t,i,l} \quad t = 1, ..., T, i = 1, ..., N \text{ and } l = 1, ..., L$$

where T is the time, N the cross-sectional and L the lag dimension.

We are interested in expected returns and the SDF for multiple horizons. Let us denote by $\mu_{i,l} = \mathbb{E}[R_{t,i}|C_{i,t-l}]$ the monthly expected return for asset i conditional on information available at $t-l$. Importantly, under standard stationarity assumptions it holds that

$$\mu_{i,l} = \mathbb{E}[R_{t,i}|C_{i,t-l}] = \mathbb{E}[R_{t+l,i}|C_{i,t}]$$

The expected return of an S period investment is equal to

$$\mu_i^S = \sum_{l=1}^{S}\mathbb{E}[R_{t+l,i}|C_{i,t}] = \sum_{l=1}^{S}\mu_{i,l}$$

Hence, by obtaining the monthly expected returns for different lags, we also obtain a model for multi-horizon investments.

Importantly, the sensor model implies that we only require knowledge of K factors to construct any optimal multi-horizon investment. Hence, we do not estimate different models for different horizons, but have one parsimonious model. The SDF weights on these factors can depend on the investment horizon.

Recall that our tensor factor model imposes the low rank tensor structure

$$R_{t,i,l} = \sum_{k=1}^{K} \lambda_k F_{t,k} B_{i,k} W_{l,k}$$

or in tensor notation

$$R = \sum_{k=1}^{K} \lambda_k \cdot F \circ B_k \circ W_k \qquad \text{for } F \in \mathbb{R}^{T\times K},\ B \in \mathbb{R}^{N\times K} \text{ and } W \in \mathbb{R}^{L\times K}$$

We include the normalizing scalar λ_k in the loadings B_k and thus drop it from the notation.

Assume that the factors are stationary in the sense that their expected return is not horizon dependent. The multi-horizon expected return for asset i directly follows from the tensor model

$$\mu_i^S = \sum_{k=1}^{K} (\mathbb{E}[F_{t,k}]) \left(\sum_{l=1}^{S} W_{l,k} \right) B_{i,k}$$

Under standard APT assumptions, the SDF is spanned by the factors F. We can study the SDF weights for multiple investment horizons. Similar to the individual stock returns, let me denote the multi-horizon mean return for the characteristic based factors as

$$\mu_k^{F,S} = \left(\sum_{l=1}^{S} W_{l,k} \right) \mathbb{E}[F_{t,k}]$$

If returns for different months are uncorrelated (which is an empirically reasonable assumption), then we can express the multi-horizon variance as

$$\text{var}_k^{F,S} = \left(\sum_{l=1}^{S} W_l W_l^\top \right) \text{var}(F_t)$$

where $\text{var}(F_t)$ denotes the covariance matrix of the factors estimated with monthly data.

Hence, the SDF weights w^S for an investment horizon of S months are equal to

$$w^S = \left(\text{var}_k^{F,S}\right)^{-1} \mu_k^{F,S} = \left(\left(\sum_{l=1}^{S} W_l W_l^\top \right) \text{var}(F_t) \right)^{-1} \text{diag}\left(\left(\sum_{l=1}^{S} W_l \right) \right) \mathbb{E}[F_t]$$

The multi-horizon mean of the SDF is equal to $\mu^{F,S} w^S$

We would like to study the following objects:

1. SDF weights as a function of horizon, i.e. for different number of factors, we show w^S as function

of S in a lineplot. We can normalize the l_1 norm of the weights to 1.

2. The Sharpe ratio of the SDF as a function of the investment horizon:. This corresponds to

$$SR^S = \sqrt{(\mu^{F,S})^\top w^S / S}$$

   We normalize it by S to express it in monthly terms.

3. The mean of the factors as a function of the horizon, i.e. $\mu_k^{F,S}$. This would be the most meaningful, if we normalize the unconditional variance of the factors to 1.

3 Data

4 Empirics

TO DO:

1. Use standard specification with simple returns in excess of the risk-free rate
2. Use log returns
3. Estimate means and portfolio weights using the formulas with and without the uncorrelatedness assumption
4. Compare OOS to weights/MVE constructed using sample means and covariances computed at 1-month, 1-year, 2-year, 5-year, 10-year horizons.

4.1 The Term Structure of Characteristics

In this section we document the term structure effects in the mean returns of our three-dimensional array R. This section does not yet use any factor models to describe the patterns. In total we have six heatmaps. These characteristic-lag heatmap are the type of heatmap like Figure 4 in the last draft. The Y-axis displays the different characteristics and the X-axis the different lags. The metric is calculated over time e.g. means.

1. Characteristic-lag heatmap for mean returns.
2. Characteristic-lag heatmap for mean of lag-excess returns.
3. Characteristic-lag heatmap of in-sample time-series alphas of the CAPM model
4. Characteristic-lag heatmap of in-sample time-series alpha t-statistics of the CAPM model
5. Characteristic-lag heatmap of in-sample time-series alphas of the CAPM model for lag-excess returns
6. Characteristic-lag heatmap of in-sample time-series alpha t-statistics of the CAPM model for lag-excess returns

4.2 A Tensor Solution

In this section we compare the benchmark metrics for the four models T-PCA, PCA, T-FF6 and FF6. The point is not necessarily to "beat" PCA, but to show that a parsimonious model achieves a similar performance, i.e. that the underlying model has a very specific structure.

We start with two summary tables.

1. The first table includes the results for T-PCA and PCA for 1 to 10 factors for in-sample and out-of-sample results. More specifially the left side of the table has the in-sample results and the right side the out-of-sample results. Within each side we have a multicolumn for T-PCA and a multicolumn for PCA. Then we have five columns for SR, XS-α, XS-α^{lex}, σ_ϵ and $\sigma_\epsilon^{\text{lex}}$. The different rows are for factor 1 to 10. The first factor has W_1 normalized to be constant for T-PCA.

2. The second table shows the same resuls for 1 to 6 factors for T-FF6 and FF6 factors.

Next we show three line plots that have subset of the numbers of the tables.

1. The first line-plot shows the out-of-sample Sharpe ratio for T-PCA and PCA and PCA based only on R_1 for 1 to 10 factors.

2. The second line-plot shows the out-of-sample XS-α for T-PCA and PCA as a function of the number of factors for 1 to 10 factors.

3. The third line-plot shows the out-of-sample σ_ϵ for T-PCA and PCA as a function of the number of factors for 1 to 10 factors.

The following bar plots dissect where the differences in pricing errors and variation come from. In total we have eight set of bar plots.

1. The first bar plot shows the XS-α_i for different characteristics and different number of factors. Deviating from the notation, we average the errors over all quantile portfolios for each characteristic. Hence, this first set of barplots has five (one for each characteristic) sets of 10 bars (one for each factor). We first show the in-sample results for T-PCA.

2. The second bar plot shows XS-α_i for T-PCA out-of-sample.

3. Barplot for XS-α_i for PCA in-sample.

4. Barplot for XS-α_i for PCA out-of-sample.

5. Number 5- 8 like the first 4 but for $\sigma_{\epsilon,i}$.

The next set of line-plots helps with the interpretation of the factors. We have two set of line plots and two set of heatmaps:

1. Line plots of W for the first 10 factors for T-PCA (similar to Figure 15 in the latest draft)

2. Line plots of W for T-FF6 (similar to Figure 15 in the latest draft)

3. Heatmaps for loadings B for T-PCA. This is a 3 time 5 matrix with the five characteristics on the x-axis and the three terciles on the y-axis. We have one plot for each factor.

4. Heatmaps for the PCA loadings Λ^{PCA}. This is the same as Figure 1 in the latest draft.

Last but not least, we consider heatmaps for the pricing errors. In total we consider four heatmaps at the moment (after we have the complete tables and lineplots from above, we might want to consider the factor models that we use here).

1. Characteristic-lag heatmap for the pricing errors $\alpha_{i,l}$ for T-PCA with 6 factors.

2. Characteristic-lag heatmap for the pricing errors $\alpha_{i,l}$ for PCA with 6 factors.

3. Characteristic-lag heatmap for the pricing errors $\alpha_{i,l}$ for T-FF6.

4. Characteristic-lag heatmap for the pricing errors $\alpha_{i,l}$ for FF6.

4.3 More Characteristics

Depending on the results, we could potentially include one summary table with the results for a larger number of characteristics.

5 Conclusions

References

Appendix

A Overview

The tensor PCA approach can provide a parsimonious model for the term-structure of stock returns. Let me please summarize how I think about this problem and introduce some unifying notation. The main object of interest are returns of portfolios sorted on lagged characteristics. Let's define the monthly excess return of portfolio i at time t sorted on $C_{i,t-m}$ characteristics by

$$r_{t,i}|C_{t-m} =: r_{t,i,m}.$$

This means our data set is a three-dimensional array of dimension $T \times N \times M$. The motivation is to study long-horizon returns. Let's assume that we use log-returns as this simplifies the aggregation, but the results should be robust to this choice. The return over M time periods sorted on $C_{i,t-1}$ equals

$$\sum_{m=1}^{M} r_{t-1+m,i}|C_{t-1}.$$

The M period expected return is

$$\sum_{m=1}^{M} E\left[r_{t-1+m,i}|C_{t-1}\right] = \sum_{m=1}^{M} E\left[r_{t-1+m,i,1}\right].$$

In order to study the term-structure of risk premia we need to make the following crucial stationarity assumption, where expectations are taken over time:

$$E[r_{t-1+m,i}|C_{t-1}] = E[r_{t,i}|C_{t-m}]$$
$$E[r_{t-1+m,i,1}] = E[r_{t,i,m}].$$

This means that the multi-period risk premia can be decomposed into its individual parts:

$$E\left[\sum_{m=1}^{M} r_{t-1+m,i,1}\right] = \sum_{m=1}^{M} E[r_{t,i,m}].$$

We introduce a tensor factor model on the excess returns:

$$r_{t,i,m} = \sum_{k=1}^{K} f_{t,k} \cdot b_{i,k} \cdot g_{m,k}$$

This implies that the term-structure risk premia should equal

$$E[r_{t,i,m}] = \sum_{k=1}^{K} E[f_{t,k}] \cdot b_{i,k} \cdot g_{m,k}.$$

Each factor itself has a risk-premia term structure which is given by

$$g_{m,k} \cdot E[f_{t,k}].$$

This means the term structure of the factor is given by $g_{:,k}$ except for a scaling factor. The factor model is not identified. Given a specific estimation, I propose to use the following normalizations:

$$Var(f_{t,k}) = 1,$$
$$\sum_{m=1}^{M} g_{m,k}^2/M = 1.$$

In order to study the term structure risk premia of assets, we would like to have a table or plot of $E[r_{t,i,m}]$. So far, we have this result for the Sharpe ratios of $r_{t,i,m}$, but now we should also add it for the mean. We can then study the goodness of fit of the term structure risk premia by calculating the term structure alpha

$$\alpha_{i,m} = E[r_{t,i,m}] - \sum_{k=1}^{K} E[f_{t,k}] \cdot b_{i,k} \cdot g_{m,k}.$$

B Tensor Decomposition

B.1 Idea

Fundamentally we are interested in modeling asset pricing returns, not characteristics per se. I will, therefore, work directly with returns on portfolios of stocks sorted on characteristics. Under some assumptions (e.g., risk prices are linear in characteristics) this approach is optimal, meaning that an SDF constructed from portfolios of returns achieves MVE efficiency and thus the corresponding MVE portfolio has the same Sharpe ratio as the MVE portfolio constructed from individual stock returns.

We use the following indexing notation: $t = 1..T$ for time, $i = 1..I$ for stocks, $j = 1..J$ for characteristics, and $l = 1..L$ for characteristic lags.

Let $r_{j,l,t}$ denote a time t excess return on a portfolio of stocks sorted linearly on a characteristic j measured at lag l:

$$r_{j,l,t} = \sum_{i=1}^{I} c_{j,i,t-l} r_{i,t}, \tag{A1}$$

where $c_{j,i,t-l}$ denotes a jth characteristic for the stock i measured at $t - l$, and $r_{i,t}$ denotes excess return of a stock i at t. Let $\boldsymbol{R}$ denote a three-dimensional $T \times J \times L$ tensor which collects all $r_{j,l,t}$ across the three indexes.

The goal is to find a low-rank approximation of the tensor $\boldsymbol{R}$ and to extract factors which succinctly summarize the dynamics of all $J \times L$ return series, together with loadings across stocks and lags. This question is related to the idea of alpha decay explored in the literature. This literature looks at the evolution of returns sorted on contemporaneous characteristics across multiple investment horizons. This is equivalent to looking at contemporaneous returns on portfolios sorted on characteristics which are measured at different lags.

Note that the usual case considered in the literature is $L = 1$, so we focus only on the most recently observed characteristics and work with matrices rather than tensors. We will instead explicitly use information from portfolios which are sorted on older measurements of characteristics. One way to approach this problem is by flattening the tensor $\boldsymbol{R}$ into a matrix of the size $T \times JL$, so that all portfolios based on different lags are treated in the same fashion, with a subsequent application of SVD or PCA. This approach is fine, but it does not impose enough structure, i.e., it does not exploit the fact that portfolios with returns $r_{j,1,t}$ and $r_{j,2,t}$ are effectively the same portfolio (for persistent characteristics).

B.2 The Method

In what follows we will explicitly work with the three-dimensional tensors and decompose them directly. The approach is known under various names, such as tensor rank decomposition, tensor component analysis, (canonical) polyadic decomposition (PD), and is a generalization of the idea behind the matrix singular value decomposition (SVD) to tensors. It was introduced by Hitchcock in 1927 and later rediscovered several times in various fields. It is sometimes referred to by the names of proposed algorithms, such as CANDECOMP, PARAFAC, CANDECOMP/PARAFAC (CP).

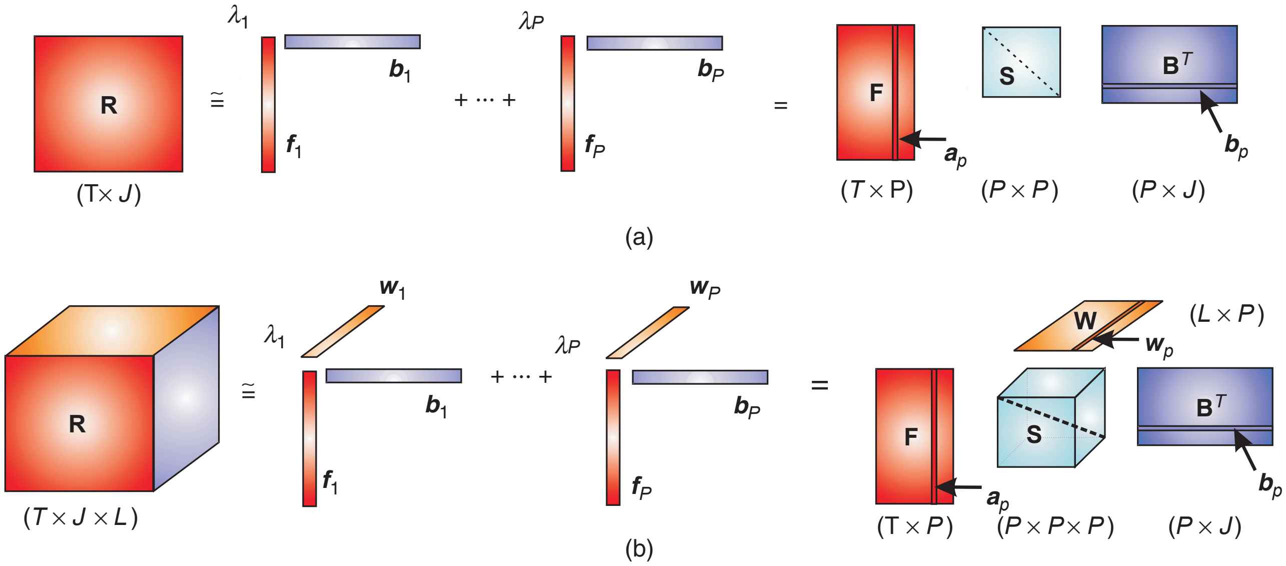

Figure A.1: SVD (a) and tensor (b) decompositions.

B.2.1 A two-dimensional case: PCA/SVD

First, consider the case of $L = 1$, that is we work with standard characteristics portfolios widely used in asset pricing, which are simply sorted on the latest available observation of any given characteristic. SVD/PCA amount to a matrix factorization. In particular SVD looks for a factorization in the form

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{R} - \boldsymbol{F}\boldsymbol{S}\boldsymbol{B}'\|_F^2 \quad \text{s.t.} \quad \boldsymbol{F}'\boldsymbol{F} = \boldsymbol{B}'\boldsymbol{B} = \boldsymbol{I}, \tag{A2}$$

that is, a factorization of the $T \times J$ matrix $\boldsymbol{R}$ into a $T \times P$ matrix of time-series factors $\boldsymbol{F}$, a $J \times P$ matrix of cross-sectional factor loadings $\boldsymbol{B}$, and a diagonal $P \times P$ matrix $\boldsymbol{S} = \operatorname{diag}(\lambda_1, ...\lambda_P)$. $\|\cdot\|_F$ denotes Frobenius norm.

Eckhart-Young thereom states that the optimal rank-p approximation of $\boldsymbol{R}$ in the least-squared sense is given by the rank-p truncation of SVD. Because $\boldsymbol{S}$ is diagonal, the rank-p approximation can be written as the sum of p products of distinct rank-1 vectors:

$$\boldsymbol{R} = \sum_{p=1}^{P} \lambda_p \boldsymbol{f}_p \circ \boldsymbol{b}_p, \tag{A3}$$

where the $\circ$ symbol denotes the outer product of two vectors, λ_p are scalars, $\boldsymbol{f}_p, \boldsymbol{b}_p$ are the time-series and cross-sectional (portfolio) factors, respectively.

The well-known indeterminacies intrinsic to this model are: 1) arbitrary scaling of components and 2) permutation of the rank-1 terms. Another indeterminacy is related to the physical meaning of the factors: if the model in (A3) is unconstrained, it admits infinitely many combinations of $\boldsymbol{f}$ and $\boldsymbol{b}$. Standard matrix factorizations in linear algebra, such as QR-factorization, eigenvalue decomposition (EVD), and SVD, are only special of (A3), and owe their uniqueness to hard and restrictive constraints such as triangularity and orthogonality. On the other hand, certain properties of the factors in (A3) can be represented by appropriate constraints, making possible the unique estimation or extraction of such factors. These constraints include statistical independence, sparsity, nonnegativity, exponential structure, uncorrelatedness, constant modulus, finite alphabet, smoothness, and unimodality.

Figure A.1 (a) depicts this decomposition graphically.

B.2.2 A three-dimensional case: tensor decomposition

Now let's consider the case when we measure a history of characteristics, $L > 1$.

A polyadic decomposition (PD) of a tensor $\boldsymbol{R}$ is a linear combination of rank-1 tensors in the form

$$\hat{\boldsymbol{R}} = \sum_{p=1}^{P} \lambda_p \boldsymbol{f}_p \circ \boldsymbol{b}_p \circ \boldsymbol{w}_p, \tag{A4}$$

where λ_p are scalars, $\boldsymbol{f}_p, \boldsymbol{b}_p, \boldsymbol{w}_p$ are the time-series $(T \times 1)$, cross-sectional $(J \times 1)$, and lag factors $(L \times 1)$, respectively. Figure A.1 (b) depicts this decomposition graphically.

PD is an intuitive generalization of SVD to three or more dimensions. There are important differences, however. Unlike SVD/PCA, it does not have rotational indeterminacy and can be shown to be unique under certain conditions, up to scaling constants.[1] Due to this, there is no need to impose orthogonality across factors. Imposing orthogonality generally leads to the loss of fit. Because orthogonality is usually not imposed, factors are not orthogonal. Moreover, factors of a $p+1$ rank model do not generally contain factors of a rank-p model. Lastly, because the problem is typically solved via optimization, it can easily handle missing values by omitting them from the objective of minimizing the Frobenius norm of the difference between the given data tensor and its CP approximation,

$$\min_{\lambda, \boldsymbol{f}, \boldsymbol{b}, \boldsymbol{w}} \left\| \boldsymbol{R} - \sum_{p=1}^{P} \lambda_p \boldsymbol{f}_p \circ \boldsymbol{b}_p \circ \boldsymbol{w}_p \right\|_F^2 . \tag{A5}$$

Instead of imposing orthogonality, the literature often imposes different constraints, such as non-negativity, sparsity etc. The latter mirrors the ideas from Sparse PCA.

B.2.3 Solving the model

The problem can be solved by iterated least squares. Fixing any two vectors leads to a convex OLS problem for the third one. Alternating the vectors and iterating until convergence gives the solution. There are publicly available implementations for Python, Matlab, R etc.

B.3 Asset pricing interpretation

Equation (A4) has a natural asset pricing interpretation. To see this, first note that when $L = 1$ we obtain a standard factor model. Factors are given by $\boldsymbol{f}$ and loadings (betas) are given by $\boldsymbol{b}$, up to a constant. When $L > 1$, the product of factors and loading is additionally multiplied by the vector $\boldsymbol{w}$. This vector can be thought of as a decay factor which tells us how quickly factor structure (and risk premia) of characteristic-sorted portfolios decays with lag, that is, how different are factor loadings of portfolios sorted on characteristics at, e.g., lag 1 vs. lag 12.

For instance, if factor loadings b_i are mean zero in the cross-section and if the effect of characteristics in sorting stocks into portfolios is strongest for the most up-to-date measurements, we would expect w_1 to be a relatively large number, while w_L would be a relative small one. At the same time, the loading on the market (e.g., b_1), might be increasing in lag, to offset the effect of reduction in variance due to decaying long-short factor loadings and consistent when the idea that portfolios sorted on noise or very stale information behave like the aggregate market. For this interpretation it might make sense to restrict $\boldsymbol{w}$ to be non-negative.

Note that the weights $\boldsymbol{w}$ are automatically learned from the data. Because of the 3-way decomposition, the weights are shared across all portfolios and across all time periods. This imposes a lot of structure in the model. It doesn't mean that all portfolios have the same profile of weight decay, as long as $P > 1$, though, but there is commonality in weight decays across portfolios in the cross section.

Decay weights $\boldsymbol{w}$ are of direct interest to us and need to be carefully analyzed. They will shed light on the anomaly decay for different characteristic-sorted portfolios.

Time-series factors $\boldsymbol{f}$ are of primary interest. They encode all time-series information necessary to model the dynamics of all portfolios at all lags. As such, they encode cross-sectional predictability of the entire history of characteristics on returns. As in PCA, by construction, factors are just portfolios of underlying stock returns (are linear in returns). One can see this by fixing $\boldsymbol{b}$ and $\boldsymbol{w}$: factors $\boldsymbol{f}$ in (A4) are then just OLS coefficients and

[1] A classical uniqueness condition is due to Jennrich (1970) and Kruskal (1977).

hence are linear in $\boldsymbol{R}$. We can use these factors directly in any asset pricing application, such as a construction of an SDF or an MVE portfolio.

B.3.1 Constraints and shrinkage

As mentioned earlier, under quite mild conditions, the PD is unique by itself, without requiring additional constraints. However, to enhance the accuracy and robustness with respect to noise, prior knowledge of data properties (e.g., statistical independence, sparsity) may be incorporated into the constraints on factors so as to facilitate their physical interpretation, relax the uniqueness conditions, and even simplify computation.

One such constraint could be a non-negativity constraint on $\boldsymbol{w}$ to facilitate the decaying weights interpretation.

We can potentially impose penalties on $\boldsymbol{b}$, e.g., to impose sparsity on the loadings on factors, to facilitate interpretation. This is similar to Sparse PCA approaches.

Another possibility is to impose some other statistical or economic constraints. For instance, in terms of objective, we could focus on RP-PCA. We could also blend in an SDF objective into the problem and impose sparsity on SDF risk prices, similar to SCS.

Lastly, Bayesian interpretation could be helpful, both in terms of motivating the objective (could be an L^1 objective for robustness, for instance), as well as penalties and sparsity.

B.4 Implementation

It should be rather straightforward:

1. Using characteristics data, form portfolios at all lags (say, up to 10 years) using linear cross-sectional sorts as in (A1). Use rank-transformed centered characteristics.
2. Form the tensor $\boldsymbol{R}$. Center the tensor along the time dimension.
3. Implement the algorithm using the alternating least squares or use one of the available packages (e.g., *parafac* in Matlab, *tensorly* package in Python) to compute the decomposition.
4. Explore and interpret factors $\boldsymbol{f}$, decay weights $\boldsymbol{w}$.
5. Use factors in asset pricing applications, e.g., in estimating an SDF. Potentially both objective can be blended to estimate everything in one shot, while imposing economically-motivated regularization.

B.5 Tensor decomposition for characteristics

In principle we could apply PD to characteristics themselves instead of portfolios by stacking all characteristics into a three-dimensional tensor $\boldsymbol{C}$. Note that the method can easily deal with missing data, due to the very rigid structure it imposes. We could perhaps even get away with including stocks which appear only in a part of sample. Still, this approach might be not ideal, since it will estimate fixed loadings for each stock and, therefore, assume that companies do not evolve through time in terms of their characteristics. It might be okay to implement this using a short moving window across the time-series dimension.

The approach of flattening $\boldsymbol{C}$ along the cross-sectional and lag dimensions could be more useful here. Basically we assume that measurements at different lags give us new observations for the joint distribution of J characteristics, and we are learning about this distribution from both the cross-section of stocks and across all lags. This make a lot of sense when characteristics are rank-transformed, since in this case all time-series information is already removed. We then work with two-dimensional matrices and can just use SVD/PCA's objectives, but still focus on minimizing Frobenius norm for observed datapoints. This approach can then be used to fill in any missing observations (if some characteristics for that stock are observable at that time).

This is a separate idea and is somewhat related to the paper Markus and Svetlana are working on. We don't have to do this now.