FINANCIAL MACHINE LEARNING PROJECTS (FIN-423)

PROJECT : RISK FACTORS EXTRACTION USING DATA DRIVEN APPROACHES - SWISSQUOTE

Group

Gergo Berta (359525)

Ali ElGuindy (28723)

Under the supervision of

Dr. Damien Ackerer

Dr. Urban Ulrych

December 23, 2023

1 Introduction

1.1 Context

Factor models have profound importance for the field of finance for multiple reasons. The core idea of factor models is to project the covariance matrix of the whole universe of stocks into a lower dimension that can explain the variance as much as possible. Furthermore, optimal portfolios can be constructed using a combination of these factors that prevent arbitrage. Factors models also play a pivotal role in corporate finance, where discount rates need to be estimated for the equity of a firm. Finally, having appropriate risk factors to explain the distribution of losses is also of key importance for quantitative risk management.

1.2 Motivation

The Fama-French 3 and 5-factor models are among the most popular factor models in finance due to their easy interpretability and relatively good in sample performance. For both models factors correspond to pre-defined long-short strategies, which makes it easy to isolate the premium to these assuming one knows the exposure. On the other hand, one is not restricted to take these as their preferred choice of model, since estimating factors can be done purely statistically. This motivates the research to find the best-performing statistical model.

1.3 Goals

This project aims to use statistical models to estimate risk factors and their exposures that can outperform the Fama-French factor models on several criteria. Our goal is to compare the performance of a number of these models on our dataset to evaluate their performance and relative strengths compared to our benchmark models.

2 Modelling and Methods

2.1 Evaluation metrics

In this section, we will discuss the evaluation metrics and their exact definitions. The definition are taken from (Kelly et al., 2019) and (Gu et al., 2021). We will stick to these definitions throughout the report :

- **Total** R^2: This is defined as

$$1 - \frac{\sum_{i,t}(r_{i,t+1} - \hat{\beta}_t \hat{f}_{t+1})^2}{\sum_{i,t} r_{i,t+1}^2}$$

  where i,t denotes all assets for the total estimation period, $\hat{\beta}_t$ is the factor exposure (available at time t), and $\hat{f}_{t+1}$ is the factor premium (estimated using weights at t).

- **Out of Sample** R^2: It is the same as Total R^2 but $(i,t) \in OOS$ denotes only out of sample observations.

- **Predictive** R^2: This is defined as

$$1 - \frac{\sum_{(i,t)\in OOS}(r_{i,t+1} - \hat{\beta}_t\hat{\lambda})^2}{\sum_{i,t} r_{i,t+1}^2}$$

  where $\hat{\lambda}$ is the average of $\hat{f}$ in the training sample.

Note that the definitions of Total R^2 and Out of Sample R^2 are nuanced. Once Beta is estimated, it is held constant; however, the distinction lies in the use of observations at $t+1$. Thus, we cannot assert that the Out of Sample R^2 is truly out of sample in the sense that it utilizes information at $t+1$ to predict returns at $t+1$. For a genuine measure of out-of-sample performance, we refer to what is termed Predictive R^2.

3 Data

We use the University of Chicago dataset [1] which was provided to us by Urban. The dataset contains monthly returns data from 1926-2020 X unique stocks. The dataset also contains 94 unique firm-level characteristics. These become relevant for the IPCA and Autoencoder models. To avoid having to use the whole dataset for our training purposes, we restrict to use of only data from 1970 which also helps us control for information which may not be important anymore. Furthermore, we apply data preprocessing by dropping all observations which have more than 50 of 94 missing values. To focus on the relative, not absolute magnitude of a characteristic, we standardize each of the 94 features for each month by subtracting the mean and dividing by the standard deviation.

3.1 Fama-French

As discussed in the introduction we use the Fama-French 3 and 5 factor models as our benchmarks. These models are given respectively by the equations:

3-Factor Model

$$R_{i,t} - R_{f,t} = \alpha_i + \beta_i^{(1)}(R_{M,t} - R_{f,t}) + \beta_i^{(2)} SMB_t + \beta_i^{(3)} HML_t + \varepsilon_{i,t}$$

$R_{i,t}$ = total return of a stock or portfolio i at time t
$R_{f,t}$ = risk-free rate of return at time t
$R_{M,t}$ = total market portfolio return at time t
$R_{i,t} - R_{f,t}$ = expected excess return
$R_{M,t} - R_{f,t}$ = excess return on the market portfolio (index)
SMB_t = size premium (small minus big)
HML_t = value premium (high minus low)
$\beta_i, i \in [1,3]$ = factor coefficients

5-Factor Model

$$R_{it} - R_{ft} = \alpha_i + \beta_1(R_{Mt} - R_{ft}) + \beta_2 SMB_t + \beta_3 HML_t + \beta_4 RMW_t + \beta_5 CMA_t + \varepsilon_{it}$$

[1] Available From: (https://dachxiu.chicagobooth.edu/)

RMW_t = profitability premium (robust minus weak)
CMA_t = investment premium (conservative minus aggressive)
$\beta_i, i \in [1,5]$ = factor coefficients

Each of the factor exposures is therefore obtained by a Linear Regression on the estimation window. Betas and alpha are estimated in a static manner from 1990-2010. The testing set spans from 2010-2020. Note that we can only evaluate performance metrics on firms that are included within the training sample i.e. for the firms we've been able to compute their alphas and betas.

3.2 Principal Component Analysis

.

Principal Component Analysis is a dimensionality reduction technique which aims to choose the factors and betas such that it minimizes the unexplained variance. This corresponds to the following optimization problem:

$$\min_{\beta,F} \sum_{t=1}^{T} (r_t - \beta_t f_t)' (r_t - \beta_t f_t). \tag{1}$$

The solution is given by:

$$\beta = \hat{\Gamma}_{[:i]}$$
$$f_t = (\beta'\beta)^{-1} r_t$$

Where $\hat{\Gamma}$ is obtained by the eigendecomposition of the sample covariance matrix $\hat{\Sigma} = \hat{\Gamma}\hat{\Lambda}\hat{\Gamma}'$. Beta will therefore correspond to the first i eigenvectors that correspond to the i largest eigenvalues.

To construct the sample covariance matrix $\hat{\Sigma}$ we use 10 years of monthly return data closest to the out-of-sample period. We make this restriction for a number of reasons. First, to estimate the covariance matrix, returns should jointly exist between all stocks in the estimation sample and for all periods. In our case, we ease the last restriction by allowing covariance to exist between stocks as long as they have at least 2 joint observations.

While the equation above uses a static PCA model, one can also make it dynamic by rolling over one period at a time and re-estimating the covariance matrix to get time-varying betas. Hence, to make PCA comparable to IPCA and Autoencoder models we can use the following set of equations:

$$\hat{\Sigma}_t = \hat{\Gamma}_t \hat{\Lambda}_t \hat{\Gamma}_t'$$
$$\beta_t = \hat{\Gamma}_{t,[:i]}$$
$$f_{t+1} = (\beta_t'\beta_t)^{-1} r_{t+1}$$

PCA was trained on data spanning from 2000 to 2009 and it was tested from 2010 to 2020. Moreover, we had the same restriction as discussed in Fama-French, it can only be evaluated on in-sample stocks.

3.3 Instrumented PCA

Instrumented Principal Component Analysis leverages its name from the fact that its optimization problem closely resembles that of 1, but with the factor exposure instrumented and factors instrumented by their firm characteristics. In fact, it can be shown that IPCA has approximately the same solution as PCA when it is applied to characteristic managed portfolios instead of returns. IPCA leverages the variance in the instruments Z_t to pin down the exposures while reducing the noise in the estimation. In very simple words, the main idea is that we want to use observable data (firms' characteristics) to represent dynamic, un-observable factors in financial models to improve the accuracy of predicting how asset returns react to those factors over time, providing a more realistic depiction of the market's changing dynamics. Here is a conceptual overview of the IPCA estimation :

The model specification for excess return is given by this equation (1) :

$$r_{t+1} = Z_t \Gamma_\beta f_{t+1} + \varepsilon^*_{t+1} \tag{1}$$

Where Γ_β is a time-invariant constant.

Minimizing the sum of squared residuals yields the following optimization problem given by (2) :

$$\min_{\Gamma_\beta, F} \sum_{t=1}^{T-1} \left(r_{t+1} - Z_t \Gamma_\beta f_{t+1}\right)' \left(r_{t+1} - Z_t \Gamma_\beta f_{t+1}\right). \tag{2}$$

The values of f_{t+1} and Γ_β that minimize (2) satisfy the first-order conditions

$$\hat{f}_{t+1} = \left(\hat{\Gamma}'_\beta Z'_t Z_t \hat{\Gamma}_\beta\right)^{-1} \hat{\Gamma}'_\beta Z'_t r_{t+1}, \quad \forall t. \tag{3}$$

and

$$\text{vec}(\hat{\Gamma}'_\beta) = \left(\sum_{t=1}^{T-1} Z'_t Z_t \otimes \hat{f}_{t+1} \hat{f}'_{t+1}\right)^{-1} \left(\sum_{t=1}^{T-1} \left[Z_t \otimes \hat{f}_{t+1}\right] r_{t+1}\right) \tag{4}$$

We use the Python implementation of IPCA[2] by Brian Kelly, one of the authors of the original paper. Although, for a sanity check, the above algorithm was also implemented in Python yielding the same results. The model was trained on data spanning from 1970 to 2010. The subsequent testing phase utilized data from 1970 to 2020.

[2]https://github.com/bkelly-lab/ipca

3.4 Autoencoders

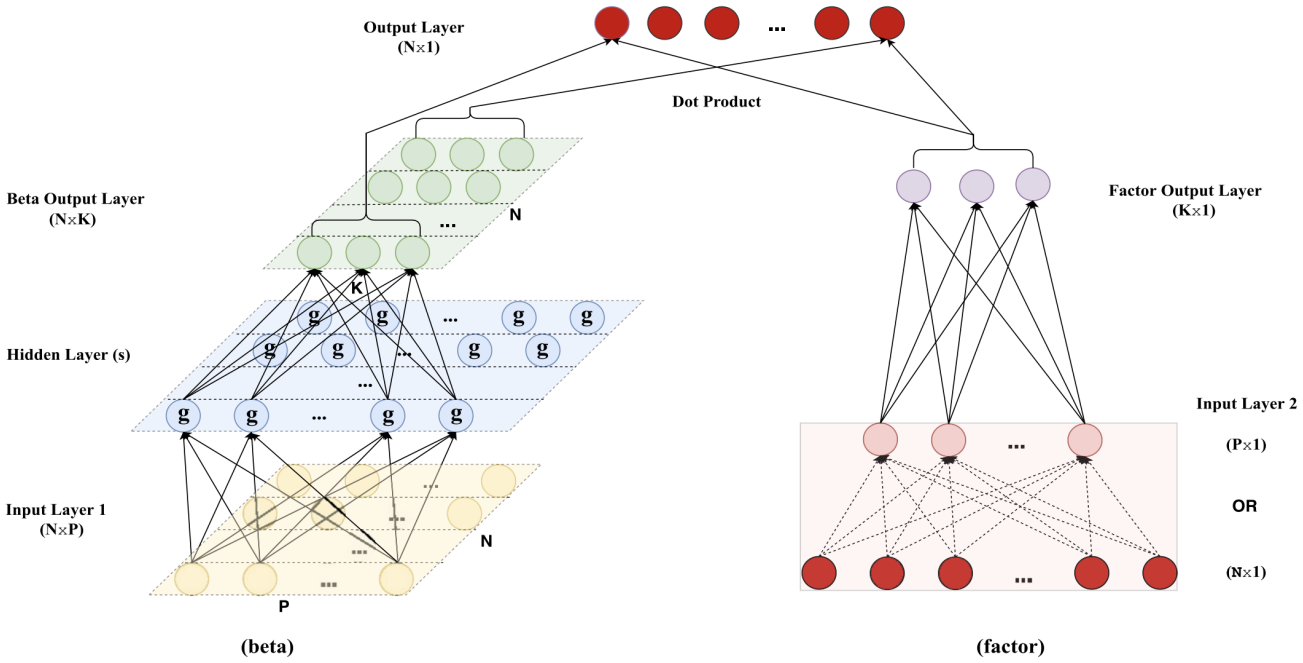

Figure 1: Autoencoder architechture taken from (Gu et al., 2021)

Autoencoder is an unsupervised machine learning model, that reduces the dimension of the input data and reconstructs it through Feedforward Neural Networks. Similarly to the IPCA model, we can use a set of characteristics Z_t to improve the estimation of factors and exposures.

For autoencoders, the left layer which determines the betas is as follows:

$$\begin{aligned} z^0_{i,t-1} &= z_{i,t-1} \\ z^l_{i,t-1} &= f(b^{l-1} + W^{l-1} z^{l-1}_{i,t-1}), \forall l \leq L \\ \beta_{i,t-1} &= b^L + W^L z^L_{i,t-1} \end{aligned}$$

This shows that at each time $t-1$ we construct the betas for each firm i by inputting its vector of characteristic that goes through $L-1$ hidden layers and a final output layer. We can note that each of the weight matrices W^j are constant which is similar to Γ_β in the IPCA model.

The layer determining the factors looks as follows:

$$\begin{aligned} x^0_t &= (Z'_{t-1} Z_{t-1})^{-1} Z_{t-1} r_t \\ f_t &= b^1 + W^1_f x^0_t \end{aligned}$$

Where the factors are linear functions in individual stock returns r_t. Hence they keep the desirable property of being investible portfolios.

The final output of the Autoencoder is given by the inner product of the two output layers: i.e.

$$r_{i,t+1} = \beta_{i,t} \cdot f_t$$

To build the Autoencoder we use Tensorflow's Keras library. Similar to (Kelly et al., 2019) we use autoencoders with 1, (CA1) 2 (CA2) and 3 (CA3) hidden layers. We optimize the hyper-parameters for each model using Scikitlearn's RandomizedSearchCV function for the validation set. We report the optimal parameters in 1. Furthermore, to control for overfitting we use the early stopping method which stops gradient descent if the model had its validation loss decrease three times in a row.

For autoencoders, we could only use 70 of 94 firm characteristics. The reason for this is that the input for factors x_t requires $Z'_{t-1}Z_{t-1}$ to be invertible which was not the the case due to the large number of missing values and/or highly correlated features.

	CA1	**CA2**	**CA3**
Activation	ReLU	ReLU	ReLU
Learning Rate	0.02	0.02	0.02
Number of Nodes (1st)	16	64	64
Number of Nodes (2nd)	-	32	128
Number of Nodes (3rd)	-	-	32
Optimizers	ADAM		
Regularization	0		
Early Stopping (time)	3		
Weight initializer	He Uniform		
Batch Size	1000		
Epcohs	50		

Table 1: Autoencoder Hyperparameters

4 Results

4.1 Overall Results

Table 2 above shows the performances of all our models across various evaluation metrics. It reports values for static PCA and Fama-French models trained on the (2000-2009) time period mentioned before and all evaluation scores correspond to the (2010-202) test period. Furthermore, we use the one-hidden Layer Autoencoder model (with 70 characteristics) and the IPCA models (with 94 characteristics) trained on the 1970-2009 window.

We notice that the best out-of-sample scores correspond to the 5-factor Autoencoder model although it does not work well when it comes to prediction the PCA model performs best. In general, both PCA and Autoencoder models beat the Fama-French benchmark in virtually all performance measures. On the other hand, we see a slack performance from IPCA contrary to Kelly et al. (2019). To try to address this issue we tried several approaches with IPCA, namely 1. We are using rank normalization instead of z-scores, 2. Implementing the algorithm on our own and 3. allowing for an interceptΓ_α. These results are not reported here but they yield no real improvement.

However, as we mentioned before these values are not "fair" in the sense that scores for both Fama-French and PCA were evaluated purely on stocks which were found in the sample whereas the IPCA and Autoencoder models could price any asset that had at least one previous

Model	Score	**3-Factor Model**	**5-Factor Model**
Fama-French	Total	14.2%	16.3%
	Out of Sample	2.4%	-2%
	Predictive	-1.4%	-1.4%
PCA	Total	21.2%	26.1%
	Out of Sample	9.6%	9.8%
	Predictive	0.4%	0.3%
IPCA	Total	5.5%	6.4%
	Out of Sample	3.5%	3.4%
	Predictive	-0.1%	-0.1%
CA1	Total	14.4%	15.4%
	Out of Sample	11.2%	11.8%
	Predictive	0.16%	-0.12%
CA2	Total	15.5%	16.4%
	Out of Sample	10.9%	11.6%
	Predictive	0.38%	0.48%
CA3	Total	15.8%	16.3%
	Out of Sample	10.2%	11.4%
	Predictive	0.37%	0.46%

Table 2: Model performances for 3 and 5 Factor

observation. We justify using this as the initial comparison since a successful factor model should also be able to price assets that have limited information available from the past. On the other hand, for the reported PCA and Fama-French models the scores were calculated based on a static Beta approach To address these issues in the next section we include adjustments such that both of these two models are trained solely on the same assets as the PCA and Fama-French models and we add a time dimension to both models as described in the Methods section.

4.2 Pure Comparisons For a limited Number of stocks

Due to the fact that our models specified in the previous section use vastly different amounts of stocks and estimation periods, we report the ‘pure‘ statistics across all models which train each model on the same set of 3,551 number of stocks. At first, we do not change the estimation period for our models as one of the key advantages of IPCA and Autoencoder models is to produce the static Γ and W metrics that help them create the dynamic betas. Hence this set of results is reported in Table 3. However, Table 4 also shows what happens if we restrict the estimation window for both models to the same (2000-2009) period as we did for Fama-French and PCA.

Model	Score	**3-Factor Model**	**5-Factor Model**
Fama-French	Total	17.1%	18.5%
	Out of Sample	10.5%	9.6%
	Predictive	0.05%	0.05%
PCA	Total	21.8%	25.16%
	Out of Sample	13.4%	13.55%
	Predictive	-0.7%	-0.08%
IPCA	Total	7.6%	7.6%
	Out of Sample	4.18%	5.34%
	Predictive	0.02%	-0.01%
CA1	Total	17.65%	18.26%
	Out of Sample	13.31%	13.35%
	Predictive	-0.47%	-1.29%
CA2	Total	18.58%	19.31%
	Out of Sample	10.12%	10.57%
	Predictive	0.29%	0.21%
CA3	Total	18.42%	19.13%
	Out of Sample	10.24%	10.70%
	Predictive	0.58%	0%

Table 3: Restricted sample model performances for 3 and 5 Factor
(With firms restricted to PCA ones)

Model	Score	**3-Factor Model**	**5-Factor Model**
Fama-French	Total	15.9%	16.9%
	Out of Sample	9.1%	8%
	Predictive	-0.3%	-0.3%
PCA	Total	23.6%	27%
	Out of Sample	11.5%	11.7%
	Predictive	0.06%	0.06%
IPCA	Total	8%	8%
	Out of Sample	5.7%	7.6%
	Predictive	0.03%	0.09%
CA1	Total	17.79%	18.57%
	Out of Sample	11.1 %	11.25%
	Predictive	-1.15%	-0.81%
CA2	Total	18.6%	19.93%
	Out of Sample	12.24%	13.41%
	Predictive	0.18%	0.49%
CA3	Total	20.5%	19.79%
	Out of Sample	12.55%	13%
	Predictive	0.41%	0.21%

Table 4: Restricted sample model performances for 3 and 5 Factor
(With firms and estimation period restricted to PCA cols and 10 years estimation window)

5 Discussion

Our analysis evaluated the performance of various models in relation to Fama-French benchmarks, which are factors extracted statistically. The findings indicate that Principal Component Analysis (PCA) is effective when applied to a subset of stocks. Additionally, it outperforms in-sample, out-of-sample and prediction and is a relatively quick technique. However, while Instrumented PCA (IPCA) offers more flexibility, it unfortunately delivers worse performance in comparison. On the other hand, autoencoders stand out as both powerful and flexible, representing a promising approach in this context. In our example, autoencoder models produced the best out-of-sample and predictive R^2 scores, although in different contexts. However, autoencoders are prone to overfitting and require longer training time and many hyperparameters to tune. Based on our results the more complex autoencoder models are the better they can produce predictive factors, however, they seem to overfit for the out-of-sample scores.

Fama-French 3 and 5-factor models to a variety of purely statistical approaches. Our results indicate that even a simple approach such as PCA can beat the Fama-French model when the objective is to maximize R^2 values. We note, however, that factors and exposures in PCA are not unique similar to the IPCA model.

We also see that Autoencoders provide a novel and good approach to estimating factors and their respective exposures. It does so by not relying on the set of restrictions that either PCA or Fama-French models do. Interestingly we see that when autoencoders are trained only on a few stocks it perform worse compared to the situation where it is trained on the whole universe and is even beaten by the PCA model in terms of out-of-sample scores. This is not the case for the IPCA model which performs better when both estimation periods and stocks are restricted.

6 Conclusion

In conclusion, factor models have profound importance in the world of finance. They have a large collection of use cases which estimate a good factor model of key importance. In this paper, we aimed to compare the performance of the Fama-French 3 and 5-factor models to a variety of purely statistical approaches. Our results indicate that even a simple approach such as PCA can beat the Fama-French model when the objective is to maximize R^2 values. We also see that Autoencoders provide a novel and good approach to estimating factors and their respective exposures. Contrary to one of our reference papers Kelly et al. (2019) we do not find the IPCA model adequate to beat autoencoders in any manner.

Suggestion for Future Research

One suggestion for future research could be to try incorporating reinforcement learning approaches, where the objective of the agent would be to predict returns accurately, by choosing actions for beta and the factors. Due to time constraints, we were unable to use the dataset of macro variables, that perhaps could have improved both IPCA and Autoencoder models. Hence, it is recommended to try using more characteristics. Finally, one could also try using models which combine discretionary and statistical methods similar to what Urban suggested us with Instrumented Fama-French, which uses Fama-French factors as prior in the IPCA construct.

References

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 222(1, Part B):429–450, 2021. ISSN 0304-4076. doi: https://doi.org/10.1016/j.jeconom.2020.07.009. URL https://www.sciencedirect.com/science/article/pii/S0304407620301998. Annals Issue: Financial Econometrics in the Age of the Digital Economy.

Bryan T. Kelly, Seth Pruitt, and Yinan Su. Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524, 2019. ISSN 0304-405X. doi: https://doi.org/10.1016/j.jfineco.2019.05.001. URL https://www.sciencedirect.com/science/article/pii/S0304405X19301151.