NOTE-2023-0624-1617

Manifold Optimisation for Linear Factor Models

Guangyu Xu

E-mail: guangyu.xu@cantab.ac.uk

Abstract: This note introduces the technique of manifold optimisation to build linear factor models for predicting stock returns.

July 20, 2023

Keywords: Optimisation, Linear Factor Model, Differential Geometry

1 Introduction

A linear factor model is the simplest probabilistic models with latent variables. In its generic form, a random vector $x \in \mathbb{R}^D$ follows a linear factor model [1] if it can be decomposed as

$$x = a + bf + \epsilon\,, \tag{1.1}$$

where $b \in \mathbb{R}^{D\times F}$ is a matrix, and $f \in \mathbb{F}$ contains the "factors". Despite its simple form, it is not only a powerful technique of dimension reduction, serving as the basis for probabilistic principal component analysis [2], but also extensively used as the workhorse in asset pricing [3]. It is also useful in building more powerful models, e.g., mixture models and deep probabilistic models.

In this note we focus on the application of linear factor models in predicting stock returns [4], with the novel approach of manifold optimisation. Specifically, given a set of the historical returns $\{R_t \in \mathbb{R}^N\}$ of N stocks, we aim to build a model predicting the return R_i at day i from the past information, i.e., $\{R_j\}_{j<i}$. One approach to solve this problem is to use a linear factor model, where the predictor S_t is a linear combination of F "factors"

$$S_t(\beta, A) = \sum_{l=1}^{F} \beta_l \sum_{k=1}^{D} A_{kl} R_{t-k} \tag{1.2}$$

parametrised by $\left(\beta \in \mathbb{R}^F, A \in \mathbb{R}^{D\times F}\right)$. The factors refer to the linear combinations $\sum_{k=1}^{D} A_{kl} R_{t-k}$ of returns in past D days, which are required to be linearly independent

$$\sum_{k=1}^{D} A_{kl} A_{kl'} = \delta_{ll'}\,. \tag{1.3}$$

This orthogonality constraint (1.3) is the only non-linear ingredient in the linear factor model. Note that the autoregressive is one example of such models with two factors $F = 2$.

Given the set $\{R_t\}_{t=0}^{M>D}$ of returns, the performance of this model can be measured with the mean overlap

$$f(\beta, A) = \frac{1}{D - M + 1} \sum_{t=D}^{M} f_t(\beta, A)\,, \tag{1.4}$$

where a single overlap is given by

$$f_t(\beta, A) := \frac{\langle S_t(\beta, A), R_t\rangle}{\|S_t(\beta, A)\|\,\|R_t\|}\,. \tag{1.5}$$

For computations later, the formulae for the partial derivatives are given by

$$\frac{\partial f_t(\beta, A)}{\partial A_{ij}} = \chi^i(t)\beta_j = \chi(t) \otimes \beta^\intercal\,, \tag{1.6a}$$

and

$$\frac{\partial f_t(\beta, A)}{\partial \beta_l} = \sum_{k=1}^{D} A_{kl}\chi^k(t) = \chi^\intercal(t) \cdot A\,, \tag{1.6b}$$

where $\chi(t) \in \mathbb{R}^D$ is a vector given by

$$\chi^k(t) = \frac{\langle R_{t-k}, R_t\rangle}{\|R_t\|\,\|S_t(\beta, A)\|} - f_t(\beta, A)\frac{\langle R_{t-k}, S_t(\beta, A)\rangle}{\|S_t(\beta, A)\|^2}\,. \tag{1.7}$$

2 Manifold Optimisation

This problem can be re-framed as a constrained optimisation problem where the minimising objective L is taken to be

$$L(A, \beta) := -f(A, \beta) \tag{2.1}$$

over the parameter space $(\beta, A) \in \mathbb{R}^F \times \mathbb{R}^{D\times F}$, with the orthogonality constraint (1.3).

However it is not trivial to preserve the orthogonality constraint while training the model. Instead, it can be converted to an unconstrained optimisation problem over the loci of the constraint (1.3), which contains the Stiefel manifold

$$\mathcal{V}_F(\mathbb{R}^D) = \{A \in \mathbb{R}^{D\times F} \mid A^\intercal A = \mathbb{1}\}. \tag{2.2}$$

Therefore the problem reduces to searching for the minimiser of (2.1) in the parameter space $\mathbb{R}^F \times \mathcal{V}_F(\mathbb{R}^D)$ without constraints.

2.1 Stiefel Manifold

We now have converted the problem to a geometric problem. We would like to find a curve $\mathbb{R} \to \mathbb{R}^F \times \mathcal{V}_F(\mathbb{R}^D)$ such that the objective $L : \mathbb{R}^F \times \mathcal{V}_F(\mathbb{R}^D) \to \mathbb{R}$ decreases the fastest. This curve decomposes into two components: $\Gamma : \mathbb{R} \to \mathbb{R}^F$ and $Y : \mathbb{R} \to \mathcal{V}_F(\mathbb{R}^D)$.

To find the descent curve Y in $\mathcal{V}_F(\mathbb{R}^D)$, we need to find the gradient $\nabla_A L$ at a point $A \in \mathcal{V}_F(\mathbb{R}^D)$ such that

$$\langle \nabla_A L, Z\rangle = \mathcal{D}_A L(Z), \tag{2.3}$$

where the right hand side is the directional derivative along the integral curve γ of the tangent vector $Z \in T_A\mathcal{V}_F(\mathbb{R}^D)$. The directional derivative along a curve $\gamma(\tau) = A + \tau Z$ is

$$dL = \sum_{k,l} \frac{\partial L}{\partial A_{kl}} \frac{\partial A_{kl}(\gamma(\tau))}{\partial \tau}\bigg|_{\tau=0} = \sum_{k,l} \frac{\partial L}{\partial A_{kl}} Z_{kl} = \operatorname{tr} G^\intercal Z, \tag{2.4}$$

where the matrix $G \in \mathbb{R}^{D\times F}$ is

$$G^{kl} := \frac{\partial L}{\partial A_{kl}}. \tag{2.5}$$

The choice of inner product in (2.3) is non-unique. Here we choose the canonical inner product on $T_A\mathcal{V}_F(\mathbb{R}^D)$ given by

$$\langle Z_1, Z_2\rangle = \operatorname{tr} Z_1\left(\mathbb{1} - \frac{1}{2}AA^\intercal\right)Z_2. \tag{2.6}$$

Therefore we would like to find the gradient $\nabla_A L$ satisfying

$$\langle \nabla_A L, Z\rangle = \operatorname{tr} G^\intercal Z \tag{2.7}$$

under the canonical inner product. It can be shown [5] that

$$\nabla_A L = WA, \tag{2.8}$$

where the matrix $W \in \mathbb{R}^{D\times D}$ is

$$W = GA^\intercal - AG^\intercal. \tag{2.9}$$

2.2 Descent Curves

Now we would like to find a descent curve $Y(\tau) \in \mathcal{V}_F(\mathbb{R}^D)$ starting at A such that

- It always stays in the Stiefel manifold, i.e.,

$$Y(\tau)^\intercal Y(\tau) = \mathbb{1}\,;$$

- It is tangent to $-\nabla_A L$ at $\tau = 0$, i.e.,

$$Y'(0) = -\nabla_A L = -WA\,.$$

Such a curve can be written [6] as

$$Y(\tau) = \left(\mathbb{1} + \frac{\tau}{2}W\right)^{-1}\left(\mathbb{1} - \frac{\tau}{2}W\right)A\,. \tag{2.10}$$

The descent curve $\Gamma(\tau) \in \mathbb{R}^F$ is trivial

$$\Gamma(\tau) = \beta - \tau\nabla_\beta L \quad \in \mathbb{R}^F\,, \tag{2.11}$$

where the gradient $\nabla_\beta L$ is simply the partial derivatives $\frac{\partial L}{\partial \beta_i}$.

The total directional derivative $L'(\tau)$ along the curve $(\Gamma, Y)(\tau)$ is then given by

$$L'(\tau) = \operatorname{tr} G^\intercal Y'(\tau) - \|\nabla_\beta L\|^2\,. \tag{2.12}$$

2.3 Woodbury Formula

The formula (2.10) involves the inversion of $D \times D$ matrices, which is computationally expensive. Assuming $D > 2F$, this computation can be accelerated [6] using the Sherman-Morrison-Woodbury formula, enabling us to write

$$\left(\mathbb{1} + \frac{\tau}{2}W\right)^{-1} = \mathbb{1} - \frac{\tau}{2}U\left(\mathbb{1} + \frac{\tau}{2}V^\intercal U\right)^{-1}V^\intercal\,, \tag{2.13}$$

where $U := (G,\ A)$ and$V := (A, -G)$ are $D \times 2F$ matrices such that

$$W = UV^\intercal = (G,\ A)\begin{pmatrix} A^\intercal \\ -G^\intercal \end{pmatrix}.$$

This reduces the inversion of $D \times D$ matrices to the inversion of $2F \times 2F$ matrices.

2.4 Gradient Descent

The minimiser $(\bar\beta, \bar A)$ of the objective $L(\beta, A)$ can be found using a gradient descent algorithm:

- Initialise with random parameters $X_{[1]} = (\beta_1, A_1) \in \mathbb{R}^F \times \mathcal{V}_F(\mathbb{R}^D)$.
- While the directional derivative $L' > \epsilon$ for some threshold ϵ:
  - Do curvilinear search to obtain $X_{[k+1]}$.
- Return $X_{[-1]}$ as $(\bar\beta, \bar A)$.

The curvilinear search can be implemented as a backtracking line search [7] satisfying the Armijo condition

$$L(\tau) \leqslant L(\tau = 0) + \rho\tau L'(\tau = 0)\,, \tag{2.14}$$

of sufficient descent for some parameter $0 < \rho < 1$, typically set to the order of 10^{-3}. The algorithm follows:

- Initialise $\tau \neq 0$ to a relative large value.
- While the Armijo condition (2.14) is not met:
  - Set $\tau \leftarrow \frac{\tau}{2}$.
- Return $(\Gamma(\tau), Y(\tau))$.

3 Regularisation

Solving the manifold optimisation problem outlined in Section 2 does not completely solve the training of the linear factor model. The gradient descent algorithm would over-fit the training dataset without further restrictions.

To reduce over-fitting, we deploy a shrinkage to the objective (2.1) to encourage less complexity in the model. Here we use the ridge regularisation [8] to add a penalty term

$$L_{\text{reg}}(\beta, A; \lambda) = \lambda \sum_{k=1}^{D} \sum_{l=1}^{F} \beta_l^2 A_{kl}^2 = (A\beta)^2 \,, \tag{3.1}$$

where λ is a parameter controlling the severity of this penalty. The partial derivatives can be derived as

$$\frac{\partial L_{\text{reg}}}{\partial A_{ij}} = 2\lambda \beta_j^2 A_{ij} \,, \tag{3.2a}$$

and

$$\frac{\partial L_{\text{reg}}}{\partial \beta_l} = 2\lambda \beta_l \,. \tag{3.2b}$$

Note that the last expression relies on the identity $\sum_{k=1}^{D} A_{kl}^2 = 1$.

Now we are fully equipped to train a linear factor model by tuning the parameter λ based on cross validation.

References

[1] A. F. McNeil, R. Frey and P. Embrechts, *Quantitative Risk Management*, Princeton University Press, Princeton, USA (2015).

[2] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, Cambridge, USA (2016).

[3] P. Zaffaroni, *Factor Models for Conditional Asset Pricing*, Imperial College London (2019).

[4] A. Hardy, *Learning Factors for Stock Market Returns Prediction*, Queb Research & Technologies, Paris (2022).

[5] H. D. Tagare, *Notes on Optimization on Stiefel Manifolds*, Yale University (2011).

[6] Z. Wen and W. Yin, *A Feasible Method for Optimization with Orthogonality Constraints*, Rich University Technical Report (2010).

[7] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, USA (2006).

[8] T. Hastie, R. Tibshirani and J. Friedman *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer (2017).