Portfolio construction - Project

Camilo A. García Trillos

Market risk and portfolio theory

Section 1

Portfolio from linear regression

Metatheorem (Pedersen 2015)

- A time-series regression corresponds to a market timing strategy
- A cross-sectional regression corresponds to a security selection strategy
- Univariate regression corresponds to sorting securities by one signal; multivariate is sorting with respect to several signals.

Moreover, expected returns are related to regression coefficients, and risk (std. dev.)-adjusted (Sharpe ratios) is related to the t-statistic of the test $\textit{coeff} > 0$.

Linear factor-based portfolio selection

The metatheorem can be applied to the Fama-French model and other types of linear regressions.

- Obtain coefficients of regression and their t-statistic
- The positions are taken as

$$x_t \propto k(F_t - \bar{F}) \quad \text{or} \quad x_t \propto k(F_t^i - \bar{F})$$

  where the former denotes a long-short strategy in time, while the latter denotes a long-short on asset selection.

Section 2

Functionally generated portfolios

Functionally generated portfolios

- Log-prices follow a (continuous-time) stochastic process.

$$dX_t = \log(S_t) = \gamma_t dt + \sigma_t dW_t.$$

We denote

$$\sigma_\pi = \pi^\top \sigma; \qquad \gamma_{\pi,t} = \pi^\top \gamma_t + \gamma^*_{\pi,t}$$

where $a := \sigma\sigma^\top$.

$$\gamma^*_{\pi_t} = \frac{1}{2}(\sum_{i=1}^{n} \pi^i_t a^{ii}_t - \pi_t^\top a_t \pi_t).$$

We use the notation

$$S^{\diamond\rho} := \frac{S}{S^\rho}; \qquad X^{\diamond\rho} := \log(S^{\diamond\rho})$$

i.e. price relative to the portfolio with (constant in time) weights ρ; and where $a^{\diamond\rho} = \frac{d}{dt}\langle X^{\diamond\rho}, X^{\diamond\rho}\rangle$.

- Let $H \in C^2$. One can show that the portfolio

$$\pi = \lambda\rho + \nabla H(X^{\diamond\rho}); \text{ where } \lambda = 1 - \mathbf{1}^\top \nabla H(X^{\diamond\rho})$$

satisfies the *Master Equation*

$$\log(S_T^\pi) - \log(S_T^\rho) = H(X_T^{\diamond\rho}) - H(X_0^{\diamond\rho}) + \int_0^T h_t dt$$

where

$$h = \gamma_\pi^* - \lambda\gamma_\rho^* - \frac{1}{2}\sum_{i,j=1}^n D_{i,j}^2 H(X^\rho) a_{i,j}^\rho.$$

- Functionally Generated Portfolios use this idea, to choose functions H and associated portfolios π that are guaranteed to produce an excess value over the market.

Functionally generated portfolios

Example: Diversity-weighted portfolio:

$$H(\boldsymbol{y}) := \frac{1}{p}\log\left(\sum_{i=1}^{n} e^{p y_i}\right)$$

implies the portfolio

$$\pi = \nabla H(X^{\diamond\rho}) = \frac{(S^{\diamond\rho})^p}{\sum_{i=1}^{n}(S^{i,\diamond\rho})^p}$$

If *enough volatility* is available in the market, it produces almost surely a relative arbitrage if we wait long enough.

Functionally generated portfolios

Some references:

[KF09] Ioannis Karatzas and Robert Fernholz. "Stochastic Portfolio Theory: an Overview". In: *Handbook of Numerical Analysis*. Vol. 15. Elsevier, 2009, pp. 89–167.

[RX19] Johannes Ruf and Kangjianan Xie. "Generalised Lyapunov Functions and Functionally Generated Trading Strategies". In: *Applied Mathematical Finance* 26.4 (July 4, 2019), pp. 293–327.

[Str14] Winslow Strong. "Generalizations of Functionally Generated Portfolios with Applications to Statistical Arbitrage". In: *SIAM Journal on Financial Mathematics* 5.1 (Jan. 2014), pp. 472–492.

Section 3

Reinforcement learning

Reinforcement learning

In this approach, the idea is to apply reinforcement learning techniques to learn how to choose optimally portfolios to attain a given goal.

Reinforcement learning to solve
A type of learning task to find a policy (control) to maximise the reward (objective) for an agent acting in a given environment.

Approach to solve them

- Value iteration
- Policy iteration

Reinforcement learning (cont.)

To render the approach practical:

- Choose variables
- Choose type of approach
- Choose implementation (eg. Deep Forward, Convolutional, etc.)
- Choose reward (eg. Sharpe ratio, cumulated return, etc.)

Reinforcement learning

Some references:

[JXL17] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. *A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem*. July 16, 2017. arXiv: `1706.10059[cs,q-fin]`.

[Lia+18] Zhipeng Liang et al. *Adversarial Deep Reinforcement Learning in Portfolio Management*. Nov. 17, 2018. arXiv: `1808.09940[cs,q-fin,stat]`.

Section 4

Deep learning of SDF

Deep learning SDF

- We can show that the projection of an SDF onto the space of risky assets is an affine transformation of the tangency portfolio.
- Since we know that $\mathbb{E}_t[M_{t+1}E_{t+1}] = 0$, it follows that

$$\mathbb{E}[M_{t+1}E_{t+1}g(I_t, I_{i,i})] = 0$$

where I_t, $I_{t,i}$ are $\mathcal{F}_t$ measurable variables, and g is measurable.

Deep learning SDF (cont.)

In Chen, Pelger, and Zhu 2019, the authors look for an approximation of the SDF by:

- Expressing it as an affine transform of a portfolio whose weights are functions of some extended space variables. A forward neural network is used for this representation
- Using the characterisation above to find the weights. Use a GAN to find the adversarial ’g’.
- Including a LSTM to add time dependency in terms of economic variables.

Some references:

[CPZ19] Luyang Chen, Markus Pelger, and Jason Zhu. “Deep Learning in Asset Pricing”. In: *arXiv:1904.00745 [q-fin, stat]* (June 12, 2019). arXiv: `1904.00745`.