QRT Challenge

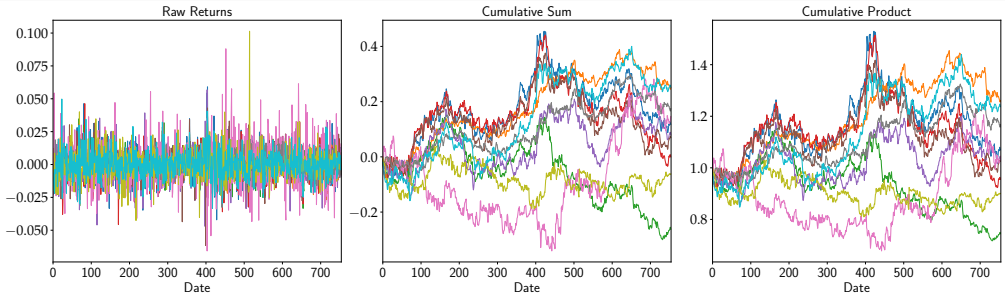
Dr. Arrykrishna Mootoovaloo

4 March 2025

QRT, London

Motivation

Motivation



- Devise techniques to predict stock market data.
- Leverage Bayesian and Deep Learning methods to find factors.

Notations

Symbol	Description	Size
m	number of stocks	$m \in \mathbb{R}^{50}$
τ	window size	$\tau \in \mathbb{R}^{250}$
$T - \tau$	number of future time, t	$T - \tau \in \mathbb{R}^{504}$
$\mathbf{X}$	time-lagged returns	$\mathbf{X} \in \mathbb{R}^{504 \times 50 \times 250}$
$\mathbf{Y}$	future returns	$\mathbf{Y} \in \mathbb{R}^{504 \times 50}$
β	unknown (latent) parameters	$\beta \in \mathbb{R}^{10}$
$\mathbf{A}$	projection matrix	$\mathbf{A} \in \mathbb{R}^{250 \times 10}$
$\tilde{\mathbf{X}} \equiv \mathrm{vec}(\mathbf{X})$	column-stack of $\mathbf{X}$	$\tilde{\mathbf{X}} \in \mathbb{R}^{25200 \times 250}$
$\mathbf{y} \equiv \mathrm{vec}(\mathbf{Y})$	column-stack of $\mathbf{Y}$	$\mathbf{y} \in \mathbb{R}^{25200}$

Table 1: Notations adopted in this work.

Exploring Different Methods

Method 1 - Brute Force (QRT Method)

Idea is to generate a random matrix $\mathbf{A}$ and orthonormalise the columns via QR decomposition.

$$\mathbf{y} = \tilde{\mathbf{X}}\mathbf{A}\beta \tag{1}$$

- $\mathbf{W} \leftarrow \mathcal{N}(0, 1, (250, 10))$
- $\mathbf{A} = \mathrm{QR}(\mathbf{W})$
- Projected matrix, $\mathbf{F} = \tilde{\mathbf{X}}\mathbf{A}$
- $\hat{\beta} = (\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{y}$
- Calculate the metric defined as:

$$\mathrm{Metric}(\mathbf{A}, \beta) := \frac{1}{504} \sum_{t=250}^{753} \frac{\langle \mathbf{Y}_t, \hat{\mathbf{Y}}_t \rangle}{\|\mathbf{Y}_t\| \|\hat{\mathbf{Y}}_t\|} \tag{2}$$

  where $\hat{\mathbf{Y}}_t \in \mathbb{R}^m$ is a vector of the predicted stocks at time t.
- Repeat until we beat the metric.

Method 2 - Exploit covariance of factor returns

$$\mathbf{y} = \tilde{\mathbf{X}}\mathbf{A}\beta \tag{3}$$

- $\mathbf{r} = (\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}$ where $\mathbf{r} \in \mathbb{R}^{250}$
- $\mathbf{r} = \mathbf{A}\beta$
- $\mathbf{r}\mathbf{r}^{\mathrm{T}} = \mathbf{A}\mathbf{C}\mathbf{A}^{\mathrm{T}}$, where $\mathbf{C} \equiv \beta\beta^{\mathrm{T}}$.
- Singular Value Decomposition:

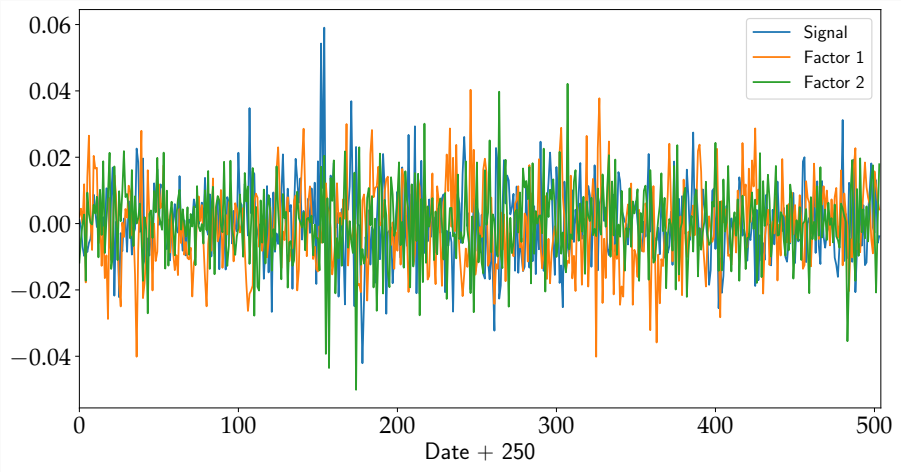
$$\mathbf{r}\mathbf{r}^{\mathrm{T}} = \mathbf{U}\Lambda\mathbf{U}^{\mathrm{T}} \tag{4}$$

  where $\mathbf{U} \in \mathbb{R}^{250\times250}$ and $\Lambda \in \mathbb{R}^{250\times250}$ is a diagonal matrix.
- We take only the first 10 columns of $\mathbf{U}$ as the matrix $\mathbf{A}$.
- Since the columns of $\mathbf{A}$ are orthonormal,

$$\hat{\beta} = \mathbf{A}^{\mathrm{T}}\mathbf{r} \tag{5}$$

Method 2 - Exploit covariance of factor returns (continued)

Example of the first two factors for the first stock returns.



Method 3 - Project $\tilde{\mathbf{X}}$

- Singular Value Decomposition:

$$\tilde{\mathbf{X}} = \mathbf{U}\Lambda\mathbf{V}^{\mathrm{T}} \tag{6}$$

- Take the first 10 columns of $\mathbf{V}$ as the matrix $\mathbf{A}$.
- These columns are orthornormal.
- Find β by least squares method (similar to Method 1).

Method 4 - Bayesian Approach using SVD of $\mathbf{r}\mathbf{r}^{\mathrm{T}}$

Write a proper likelihood for the data, that is,

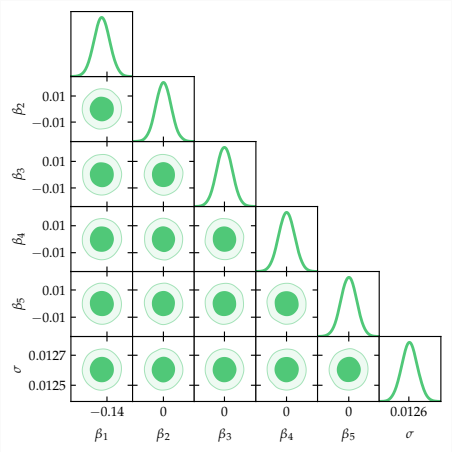
$$\mathbf{y} = \tilde{\mathbf{X}}\mathbf{A}\beta + \boldsymbol{\epsilon} \tag{7}$$

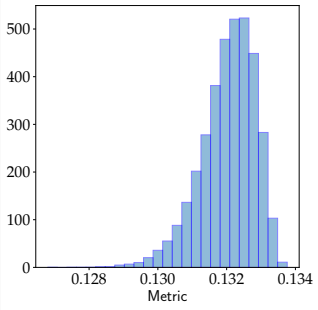
where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbb{I})$ and we fix $\mathbf{A}$ as obtained in Method 2. We then have a system of 11 unknown parameters. Using Bayes' theorem, we have:

$$p(\beta, \sigma|\tilde{\mathbf{X}}, \mathbf{Y}) \propto p(\mathbf{y}|\tilde{\mathbf{X}}, \beta, \sigma)\, \pi(\beta)\, \pi(\sigma) \tag{8}$$

and we adopt the following priors: $\beta \sim \mathcal{N}(0, 1)$ and $\sigma \sim \mathrm{Exp}(100)$.

Method 4 - Bayesian Approach using SVD of $\mathbf{r}\mathbf{r}^{\mathrm{T}}$ (continued)

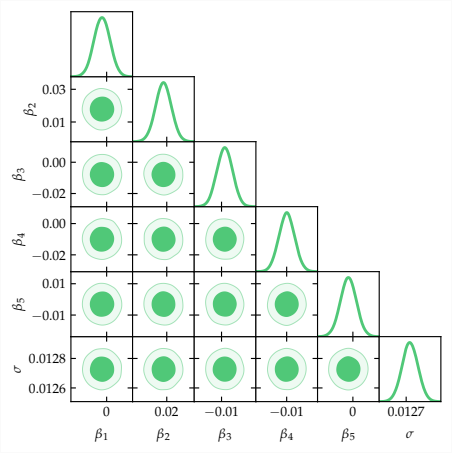


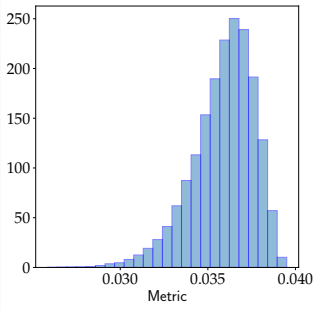


- Marginalised posterior first five parameters β.
- Important - marginalisation over σ.
- Many possible realisations of the metric.

Method 5: Bayesian Approach using SVD of $\tilde{X}$

Similar concept as in the previous slides, but we fix **A** as derived in Method 3.





- Lower metric compared to the previous method.
- But centred roughly on the QRT benchmark[a].
- Many possible realisations of the metric.

[a] 0.03535

Method 6: Deriving Factors

Instead, we can also pre-compute some factors based on the data and fit for β directly, that is,

$$\mathbf{y} = \mathbf{F}\beta \tag{9}$$

where $\mathbf{F} \in \mathbb{R}^{25200 \times 10}$.

1) **Rolling Mean**

$$R_t^{(w)} = \frac{1}{\sqrt{w}} \sum_{k=1}^{w} R_{t+1-k} \tag{10}$$

2) **Rolling Volatility**

$$\sigma_t^{(w)} = \sqrt{\frac{1}{w} \sum_{k=1}^{w} (R_{t+1-k} - \bar{R})^2} \tag{11}$$

3) **Rolling Downside Volatility**

$$\sigma_t^{(w)} = \sqrt{\frac{1}{w}\sum_{k=1}^{w} \min(R_{t+1-k}, 0)^2} \tag{12}$$

4) **Rolling Upside Volatility**

$$\sigma_t^{(w)} = \sqrt{\frac{1}{w}\sum_{k}^{w} (R_{t+1-k} - \bar{R})^2} \tag{13}$$

where $R_{t+1-k} > 0$.

5) **Rolling Skewness**

$$\text{Skewness} = \frac{\mathbb{E}[(X-\mu)^3]}{\sigma^3} \tag{14}$$

6) **Rolling Kurtosis**

$$\text{Kurtosis} = \frac{\mathbb{E}[(X-\mu)^4]}{\sigma^4} \tag{15}$$

7) **Rolling Sharpe Ratio**

$$\text{Sharpe Ratio} = \frac{\mathbb{E}[R_t - r_f]}{\sigma} \tag{16}$$

8) **Rolling Sortino Ratio**

$$\text{Sortino Ratio} = \frac{\mathbb{R}[R_t - r_f]}{\sigma_d} \tag{17}$$

9) **Omega Ratio**

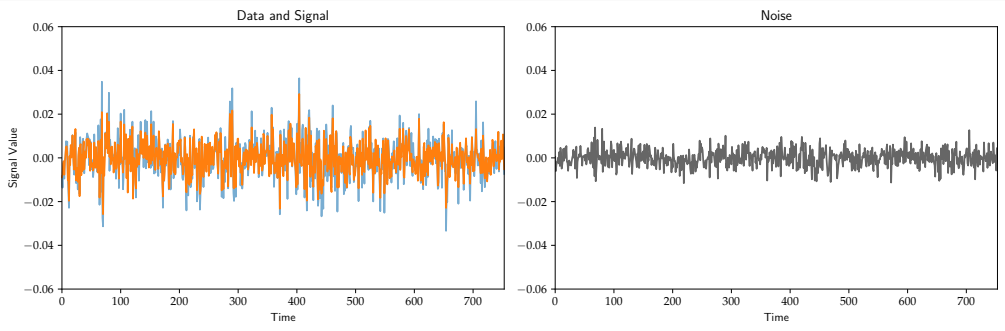
$$\Omega_t = \frac{Q^+}{Q^-} = \frac{\sum_1^w R_t^+}{\sum_1^w |R_t^-|} \tag{18}$$

One can build a series of rolling mean factors for different time lag and different window size. For example, the momentum factor is defined as:

$$M_t := R_{t-20}^{(230)} \tag{19}$$

Method 7: FFT Method

Idea to separate the signal from the data.



- Extract 250 Fourier components from each stock return.
- Compute the noise covariance matrix.
- Learn the matrix **A** via SVD.
- Fit for β via Monte Carlo sampling.

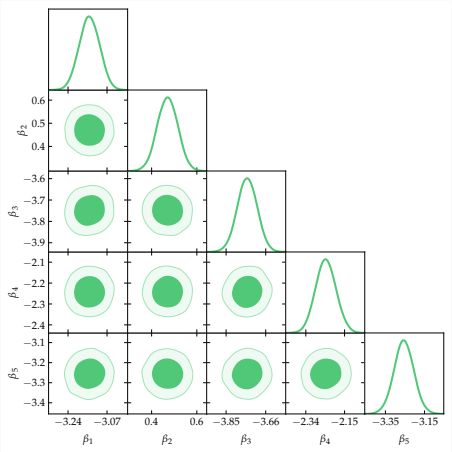
The likelihood is given by:

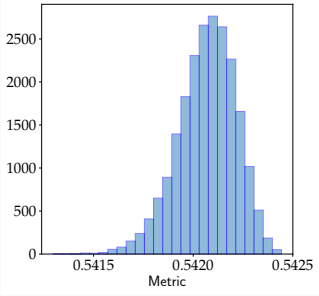
$$p(\mathbf{Y}|\beta) = \prod_{t=1}^{504} \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left[-\frac{1}{2}(\mathbf{Y}_t - \mathbf{F}_t\beta)^{\mathrm{T}}\Sigma^{-1}(\mathbf{Y}_t - \mathbf{F}_t\beta)\right] \tag{20}$$

where $\mathbf{Y}_t \in \mathbb{R}^{50}$, $\mathbf{F}_t \in \mathbb{R}^{(50\times 10)}$ and $\Sigma \in \mathbb{R}^{50\times 50}$. We are assuming that the noise covariance matrix is fixed across the time period. The posterior of β is:

$$p(\beta|\mathbf{Y}) \propto p(\mathbf{Y}|\beta)\,\pi(\beta) \tag{21}$$

Method 7: FFT Method (continued)

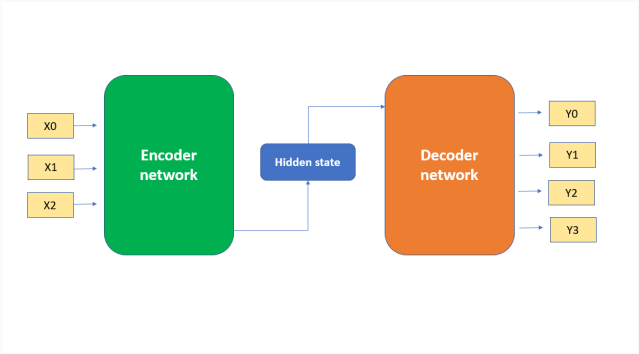




- Very high metric (expected).
- But, does not generalise to the test set.
- Score of -0.006 when submitted on the portal.
- Most likely overfitting the training set.

Method 8: Deep Learning - Autoencoder

- Encoder (CNN) and decoder architecture.



- Optimise for the metric (correlation loss) directly.
- The encoder outputs a matrix, $\mathbf{F} \in \mathbb{R}^{N \times 250}$, where $N = 504 \times 50$.
- We then proceed by finding the matrix $\mathbf{A}$ and β via least squares method.

Results

	Method	Score	Online
1	QRT Brute Force	0.0477	0.0306
2	SVD of rr^T	0.1332	0.0819
3	SVD of $\tilde{X}$	0.0396	0.0365
4	Bayesian - SVD of rr^T	0.1338	0.0810
5	Bayesian - SVD of $\tilde{X}$	0.0395	0.0365
6	Custom Factors	0.0295	0.0285
7	FFT Method	0.5420	-0.0063
8	CNN Autoencoder	0.9952	0.0049

Table 2: Calculated metrics with the different methods

The maximum score on the public and private leaderboards are 0.0918 and 0.1051 respectively. The QRT benchmark is 0.03535.

Slow Dynamics

Linear Time-Lagged Autoencoder

Suppose we have the returns, $\mathbf{R} \in \mathbb{R}^{m \times T}$, where $m = 50$ and $T = 754$. The goal is to find a d-dimensional embedding, where $d < m$, such that the error

$$\epsilon = \sum_t ||\mathbf{R}_{t+\tau} - \mathbf{K}\mathbf{R}_t||_2^2 \tag{22}$$

is minimum. $\mathbf{K}$ is the Koopman operator and can be rewritten as

$$\mathbf{K} = \mathbf{D}\mathbf{E} \tag{23}$$

where $\mathbf{D}$ is the decoder and $\mathbf{E}$ is the encoder.

Mean-free Coordinates

For a single return, r_t,

$$x_t = r_t - \frac{1}{T-\tau}\sum_{s=1}^{T-\tau} r_s \tag{24}$$

and

$$y_t = r_{t+\tau} - \frac{1}{T-\tau}\sum_{s=1}^{T-\tau} r_{s+\tau} \tag{25}$$

Pre-Whitening

$$\tilde{x}_t = \mathbf{L}_{xx}^{-1} x_t \tag{26}$$

$$\tilde{y}_t = \mathbf{L}_{yy}^{-1} y_t \tag{27}$$

where the covariances and Cholesky factors are:

$$\mathbf{C}_{xx} = \frac{1}{T-\tau}\mathbf{X}\mathbf{X}^T \tag{28}$$

$$\mathbf{C}_{yy} = \frac{1}{T-\tau}\mathbf{Y}\mathbf{Y}^T \tag{29}$$

$$\mathbf{C}_{xy} = \frac{1}{T-\tau}\mathbf{X}\mathbf{Y}^T \tag{30}$$

where $\mathbf{X} \in \mathbb{R}^{m \times (T-\tau)}$ and $\mathbf{Y} \in \mathbb{R}^{m \times (T-\tau)}$ and $\mathbf{C}_{xx} = \mathbf{L}_{xx}\mathbf{L}_{xx}^T$ and $\mathbf{C}_{yy} = \mathbf{L}_{yy}\mathbf{L}_{yy}^T$

Koopman Matrix in the Pre-whitened Space

If we denote the pre-whitened (time-lagged) returns as $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$, we want to find $\tilde{\mathbf{K}}$ such that the error

$$\tilde{\epsilon} = ||\tilde{\mathbf{Y}} - \tilde{\mathbf{K}}\tilde{\mathbf{X}}||_2^2 \tag{31}$$

is minimum. This has an analytical solution:

$$\tilde{\mathbf{K}} = \tilde{\mathbf{Y}}\tilde{\mathbf{X}}^T(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}. \tag{32}$$

Finding the encoder and decoder

We can use SVD to find the encoder and decoder of the Koopman operator in the transformed space.

$$\tilde{\mathbf{K}} = \mathbf{U}\Sigma\mathbf{V}^T \tag{33}$$

If we retain the top d components,

$$\tilde{\mathbf{E}} = \Sigma_d \mathbf{V}_d^T \tag{34}$$

$$\tilde{\mathbf{D}} = \mathbf{U}_d \tag{35}$$

where $\tilde{\mathbf{E}} \in \mathbb{R}^{d \times m}$ and $\tilde{\mathbf{D}} \in \mathbb{R}^{m \times d}$.

Koopman Matrix in the Original Space

We can show that the Koopman operator, the decoder and the encoder are:

$$\mathbf{K} = \mathbf{C}_{xy}^T \mathbf{C}_{xx}^{-1} \qquad \mathbf{D} = \mathbf{L}_{yy} \tilde{\mathbf{D}} \qquad \mathbf{E} = \tilde{\mathbf{E}} \mathbf{L}_{xx}^{-1} \tag{36}$$

Experiments

- $d = 10$ and $\tau = 250$ result in a correlation metric of 0.202.
- $d = 15$ and $\tau = 250$ result in a correlation metric of 0.240.

Time-Lagged Autoencoder (Non-Linear Embeddings)

Alternatively, we can use an autoencoder[1] to find the non-linear embeddings. As in the paper, we employ the following configurations:

- an input layer with m units,
- two hidden layers with sizes $H_1 = 200$ and $H_2 = 100$,
- the latent layer with size d,
- the decoder is simply the mirror image of the encoder and
- a learning rate of 0.001.

Experiments

We also tested two loss functions (MSE loss and correlation metric).

- For $d = 10$, the metrics are : 0.346 and 0.378.
- For $d = 15$, the metrics are : 0.359 and 0.377.

[1] Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics

Deeptime Library

Deeptime Library Functionalities

Explore the Deeptime library (https://deeptime-ml.github.io/).

Thank You