Homework 4 Write-up

Contributors: Ziheng Chen (zc2068), Cong Wang (cw3140), Chengyang Gu (cg3765)

Summary

We implement a pipeline to compute the co-integration coefficient γ on a rolling basis. We apply a smart online algorithm so it can update γ with least computation possible with new observations, making it a good candidate for real-time trading. We also design a batch-based version which provides a more accurate estimation and is 300 times faster than the rolling-based implementation. We also include in this document a proof of the multi-period trading problem.

Problem 1

(a)

We apply the Engle-Granger two-step co-integration test on the return series of each possible pair of stocks. We first take the log of prices and differentiate them to yield the returns. We then use `statsmodels` to perform the test.

The signature of the co-integration testing subroutine is

```
statsmodels.tsa.stattools.coint(y0, y1, trend, *kwargs)
```

where `y0` and `y1` are the two time series, and `trend` specifies the time term t in the first step of the E-G test. We use `trend='nc'` to exclude the constant, assuming that the expected return of each stock should be zero. The rest parameters are either useless or irrelevant to our problem and are thus omitted.

We iterate all the pairs (i, j) of stocks and find that all of them have co-integration relationship at $p = 0.05$.

(b)

The formula for γ is

$$\hat{\gamma}_t = \frac{\sum_{i=t-N+2}^{t}(y_i - mx_i - b)(y_{i-1} - mx_{i-1} - b)}{\sum_{i=t-N+2}^{t}(y_{i-1} - mx_{i-1} - b)^2}$$

where N is a pre-defined window size, and

$$
\begin{aligned}
m &= \frac{\overline{xy} - \overline{x}\,\overline{y}}{\overline{x^2} - \overline{x}^2} \\
b &= \overline{y} - m\overline{x} \\
\overline{x} &= \frac{1}{N}\sum_{i=t-N+1}^{t} x_i
\end{aligned}
$$

and similarly for $\overline{y}$, $\overline{x^2}$ and $\overline{xy}$. In real-time applications, we can use a circular queue which caches the summands to query and update their sum in $O(1)$ time. The computation of $\hat{\gamma}$ will take $O(N)$ time nevertheless.

We implement two different approaches to estimate the γ. The first reads all the data and computes all the $\hat{\gamma}$s in a batch. While this sounds slower, in practice this allows us to leverage numpy and reduce communication time. In fact, this method is 300 times faster than the iterative approach. We use this as an benchmark.

The second implementation mimics the real-time scenario by reading the data line by line and update the estimation of γ dynamically.

Problem 2

(a)

We consider the following min-variance portfolio optimization problem with risky securities and transaction costs

$$
min_x\{\frac{1}{2}x^T\Sigma x + \frac{1}{2}(x - x_0)^T\Lambda(x - x_0)\}
$$

Show that the solution to this problem is

$$
x^* = (\Sigma + \Lambda)^{-1}\Lambda x_0
$$

Let $f(x) = \frac{1}{2}x^T\Sigma x + \frac{1}{2}(x - x_0)^T\Lambda(x - x_0)$ and then take derivative w.r.t x and set it to 0. We have

$$
\begin{aligned}
f(x)^{'} &= \Sigma x + \Lambda(x - x_0) = 0 \\
\Sigma x + \Lambda x &= \Lambda x_0
\end{aligned}
$$

where Λ is symmetric positive definite matrix and thus it's invertible. If the covariance matrix of security return is also invertible, we have

$$
x^* = (\Sigma + \Lambda)^{-1}\Lambda x_0
$$

(b)

i) Note that at time T, the multi-period MVO problem reduces to one-period MVO problem. Using the result from (a) we get

$$
x_T^* = (\Sigma + \Lambda)^{-1}\Lambda x_{T-1}
$$

Let $A := (\Sigma + \Lambda)^{-1}\Lambda$

$$x^* = Ax_{T-1}$$

Thus,

$$\begin{aligned} J_T(x_{T-1}) &= \frac{1}{2}(x_T^*)^T \Sigma x_T^* + \frac{1}{2}(x_T^* - x_{T-1})^T \Lambda (x_T^* - x_{T-1}) \\ &= \frac{1}{2}(x_{T-1}^T A^T \Sigma A x_{T-1}) + \frac{1}{2} x_{T-1}^T (A-I)^T \Lambda (A-I) x_{T-1} \\ &= \frac{1}{2} x_{T-1}^T [A^T \Sigma A + (A-I)^T \Lambda (A-I)] x_{T-1} \end{aligned}$$

and

$$\begin{aligned} A^T \Sigma A + (A-I)^T \Lambda (A-I) &= A^T(\Sigma + \Lambda)A - A^T\Lambda - \Lambda A + \Lambda \\ &= \Lambda^T(\Sigma+\Lambda)^{-1}(\Sigma+\Lambda)(\Sigma+\Lambda)^{-1}\Lambda - A^T\Lambda - \Lambda A + \Lambda \\ &= \Lambda(\Sigma+\Lambda)^{-1}\Lambda - \Lambda(\Sigma+\Lambda)^{-1}\Lambda - \Lambda(\Sigma+\Lambda)^{-1}\Lambda + \Lambda \\ &= \Lambda - \Lambda(\Sigma+\Lambda)^{-1}\Lambda \\ &= K_T \end{aligned}$$

ii) Prove by induction.

First at $t = T-1$, the cost-to-go function is

$$J_{T-1}(x_{T-1}) = min_{x_{T-1}}[\frac{1}{2}x_{T-1}^* \Sigma x_{T-1} + \frac{1}{2}(x_{T-1} - x_{T-2})^T \Lambda (x_{T-1} - x_{T-2}) + J_T(x_{T-1})]$$
$$where\ J_T(x_{T-1}) = \frac{1}{2}x_{T-1}^T K_T x_{T-1}$$

$\implies$

$$J_{T-1}(x_{T-1}) = min_{x_{T-1}}[\frac{1}{2}x_{T-1}^T \Sigma_{T-1} X_{T-1} + \frac{1}{2}(x_{T-1} - x_{T-2})^T \Lambda (x_{T-1} - x_{T-2})]$$
$$where\ \Sigma_{T-1} := \Sigma + K_T$$

Thus

$$\begin{aligned} x_{T-1}^* &= (\Sigma_{T-1} + \Lambda)^{-1}\Lambda x_{T-2}\ from\ 2(a) \\ &= (\Sigma + \Lambda + K_T)^{-1}\Lambda x_{T-2} \end{aligned}$$

and

$$J_{T-1}(x_{T-2}) = J_{T-1}(x_{T-1}^*) = \frac{1}{2}x_{T-2}^* K_{T-1} x_{T-2}$$
$$where\ K_{T-1} = \Lambda - \Lambda(\Sigma_{T-1} + \Lambda)^{-1}\Lambda = \Lambda - \Lambda(\Sigma + \Lambda + K_T)^{-1}\Lambda$$

So we can guess the following form

$$x_t^* = L_t x_{t-1}$$
$$J_t(x_{t-1}) = \frac{1}{2}x_{t-1}^T K_t x_{t-1}$$

where

$$\begin{cases} L_t &= (\Sigma + \Lambda)^{-1}\Lambda \\ K_t &= \Lambda - \Lambda(\Sigma_t + \Lambda)^{-1}\Lambda,\ 1 \le t \le T \\ \Sigma_t &= \Sigma + K_{t+1}, 1 \le t \le T-1 \\ \Sigma_T &= \Sigma \end{cases}$$

Assume (10) holds true when $t = t_0$, then at $t = t_0 - 1$, we have

$$J_{t_0-1}(x_{t_0-1}) = min_{x_{t_0-1}}[\frac{1}{2}x_{t_0-1}^T\Sigma x_{t_0-1} + \frac{1}{2}(x_{t_0-1} - x_{t_0-2})^T\Lambda(x_{t_0-1} - x_{t_0-2}) + J_{t_0}(x_{t_0-1})]$$

where

$$J_{t_0}(x_{t_0-1}) = \frac{1}{2}x_{t_0-1}^T K_{t_0} x_{t_0-1} \implies \Sigma_{t_0-1} = \Sigma + K_{t_0}$$
$$x_{t_0-1}^* = (\Sigma_{t_0-1} + \Lambda)^{-1}x_{t_0-2} \implies L_{t_0-1} = (\Sigma_{t_0-1} + \Lambda)^{-1}\Lambda$$

and

$$\begin{aligned} J_{t_0-1}(x_{t_0-2}) &= J_{t_0-1}(x_{t_0-1}^*) \\ &= \frac{1}{2}x_{t_0-2}^T K_{t_0-1} x_{t_0-2} \end{aligned}$$

where $K_{t_0-1} = \Lambda - \Lambda(\Sigma_{t_0-1} + \Lambda)^{-1}\Lambda$

$\implies$ (10) is true at $t = t_0 - 1$

Complete.

(c)

We solve the infinite horizon liquidation problem. Those familiar with linear-quadratic regulators (LQR) should already notice that, compared with traditional LQR problems, this problem has a discount factor. While we cannot find any papers describing LQRs with discount factors (it might be a problem well studied in other fields, for example, reinforcement learning), we still manage to discover some interesting properties and sketch a solution to the problem.

Denote the state value function as $J(x_t)$. For infinite horizon problems, it's unnecessary to specify when this value function is observed. The Bellman equation is

$$J(x_{t-1}) = \min_{\Delta x_t} \frac{1}{2}x_t^T\Sigma x_t + \frac{1}{2}\Delta x_t^T\Lambda\Delta x_t + \theta J(x_t)$$

where $\Delta x_t = x_t - x_{t-1}$. The the first order condition w.r.t. x_t is

$$0 = \Sigma x_t + \Lambda\Delta x_t + \theta J'(x_t)$$

We take the *ansatz* $J(x_t) = x_t^T A x_t$ from here, where A is a deterministic matrix. The equation above reduces to

$$
\begin{aligned}
&\Sigma x_t + \theta A x_t + \Lambda \Delta_t = 0 \\
\implies &\Delta x_t = -(\Sigma + \theta A + \Lambda)^{-1}(\Sigma + \theta A) x_{t-1}
\end{aligned}
$$

As a sanity check, notice that with $\Lambda = 0$ (no transaction cost), the equation above degenerates to $\Delta x_t = -x_{t-1}$, implying that the agent should liquidate all the inventory immediately. This surely minimizes (zeros out) all future penalty on volatility and is the effective optimal strately.

With substitution and a little algebra, the equation can also be written as

$$
\begin{aligned}
x_t &= \Delta x_t + x_{t-1} \\
&= \Big[I - (\Sigma + \theta A + \Lambda)^{-1}(\Sigma + \theta A) \Big] x_{t-1} \\
&= (\Sigma + \theta A + \Lambda)^{-1} \Lambda x_{t-1}
\end{aligned}
$$

which corresponds to an **exponential liquidation or VWAP policy**. In other words, the agent sells a fixed proportion of the remaining inventory at each step. As $|det((\Sigma + \theta A + \Lambda)^{-1}\Lambda)| < 1$, one can expect that

$$
x_t = \Big((\Sigma + \theta A + \Lambda)^{-1} \Lambda \Big)^t x_0 \to 0 \ (t \to \infty)
$$

> Recall that, in (b), we have found the iteration of L_t as
>
> $$L_t = (\Sigma + K_{t+1} + \Lambda)^{-1}\Lambda$$

The remaining problem is to find A. For notational conciseness we let

$$
B = (\Sigma + \theta A + \Lambda)^{-1}(\Sigma + \theta A)
$$

We substitute Δx_t into the Bellman equation to get

$$
\begin{aligned}
\frac{1}{2} x_{t-1}^T A x_{t-1} &= \frac{1}{2}(x_{t-1} + \Delta x_t)^T \Sigma (x_{t-1} + \Delta x_t) + \frac{1}{2} \Delta x_t^T \Lambda \Delta x_t + \frac{\theta}{2}(x_{t-1} + \Delta x_t)^T A (x_{t-1} + \Delta x_t) \\
&= \frac{1}{2} x_{t-1}^T (I - B)^T (\Sigma + \theta A)(I - B) x_{t-1} + \frac{1}{2} x_{t-1}^T B^T \Lambda B x_{t-1} \\
\implies 0 &= x_{t-1}^T \Big(A - (I - B)^T (\Sigma + \theta A)(I - B) - B^T \Lambda B \Big) x_{t-1}
\end{aligned}
$$

holds for all x_{t-1}. Thus, we have

$$
\begin{aligned}
0 &= A - (I - B)^T (\Sigma + \theta A)(I - B) - B^T \Lambda B \\
A &= \Lambda (\Sigma + \theta A + \Lambda)^{-1}(\Sigma + \theta A)(\Sigma + \theta A + \Lambda)^{-1}\Lambda \\
&\quad + (\Sigma + \theta A)(\Sigma + \theta A + \Lambda)^{-1}\Lambda(\Sigma + \theta A + \Lambda)^{-1}(\Sigma + \theta A)
\end{aligned}
$$

The trick is to, for example, replace $\Sigma + \theta A$ with $(\Sigma + \theta A + \Lambda) - \Lambda$ and try to eliminate as many terms as possible. After some messy algebra, we finally get

$$
A = \Lambda - \Lambda(\Sigma + \theta A + \Lambda)^{-1}\Lambda
$$

> Recall that, in (b), we have found the iteration of K_t as
>
> $$K_t = \Lambda - \Lambda(\Sigma + K_{t+1} + \Lambda)^{-1}\Lambda$$

which can be further reduced to

$$
\theta A \Lambda^{-1} A + \Big((1 - \theta) I + \Sigma \Lambda^{-1} \Big) A - \Sigma = 0
$$

which is reminiscent of algebraic Riccati equations. This equation can be solved iteratively (assuming the solution is existent and unique, which doesn't generally hold).

> In fact, If we consider a equation of the form
>
> $$XAX + BX + C = 0$$
>
> where X, A, B, C are all symmetric and positive semi-definite, it's possible to parameterize all solutions.

The difference between (b) and (c) is subminimal in that we virtually recover a nearly identical recursion for the value function. However, with infinite horizon, the value function is deterministic but cannot be obtained analytically or decided deterministically.