Build Night 5

Follow-the-Loser Strategies

Team Meeting Agenda

- Discuss homework (5min)
- Go over Follow-the-Loser strategies (25min)
- Ensure every team member has Python installed (10min)
- Code architecture and module brainstorm session (20min)
- Split up code work and translate into Github Issues (10min)

Project Schedule: Zoomin’

1. Welcome & Problem Definition
2. Power Outage – Python and Reading Review
3. Baseline + Follow-the-Winner Strategies I
4. Follow-the-Winner Strategies II
5. Follow-the-Loser Strategies (you are here!)
6. Pattern-Matching Strategies
7. Framework I
8. Framework II
9. Backtesting, Experimentation, and Comparison
10. Poster Work & Presentation Practice

Homework Discussion

Follow-the-Loser

About These Strategies

- Work on the principle of mean reversion:
  - Stocks that fall will eventually revert to their mean
- A stock that performs poorly, it tends to do better in subsequent trading periods, and vice versa for good-performing stocks
- Strategies discussed:
  - Anti-Correlation (Anticor)
  - Passive-Aggressive Mean Reversion
  - Confidence-Weighted Mean Reversion
  - Online Moving Average Reversion
  - Robust Mean Reversion

Anti-Correlation (Anticor)

- Assumes market follows mean reversion
- Utilize two windows of logarithmic price relatives and calculate cross-correlation matrix between these two windows

price relatives [Hull 2008] in two specific market windows—that is, $\mathbf{y}_1 = \log(\mathbf{x}_{t-2w+1}^{t-w})$ and $\mathbf{y}_2 = \log(\mathbf{x}_{t-w+1}^{t})$. It then calculates the cross-correlation matrix between $\mathbf{y}_1$ and $\mathbf{y}_2$:

$$M_{cov}(i,j) = \frac{1}{w-1}\left(\mathbf{y}_{1,i} - \bar{y}_1\right)^{\top}\left(\mathbf{y}_{2,j} - \bar{y}_2\right)$$

$$M_{cor}(i,j) = \begin{cases} \frac{M_{cov}(i,j)}{\sigma_1(i) * \sigma_2(j)} & \sigma_1(i), \sigma_2(j) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

- Moves proportions from the stocks that increased more in these two windows to the ones that increased less

Anti-Correlation (Anticor)

- Empirically outperforms all other strategies
  - This is a thing we can test!
- It's very much heuristic-based, other methods use learning algorithms

Passive-Aggressive Mean Reversion (PAMR)

- Write a loss function that reflects mean reversion and optimize this loss function

$$\ell_\epsilon(\mathbf{b};\mathbf{x}_t) = \begin{cases} 0 & \mathbf{b}\cdot\mathbf{x}_t \leq \epsilon \\ \mathbf{b}\cdot\mathbf{x}_t - \epsilon & \text{otherwise} \end{cases},$$

- ε = sensitivity parameter in range [0, 1], controls the mean reversion threshold

Passive-Aggressive Mean Reversion (PAMR)

- Optimization problem:

$$\mathbf{b}_{t+1} = \arg\min_{\mathbf{b}\in\Delta_m} \frac{1}{2}\left\|\mathbf{b} - \mathbf{b}_t\right\|^2 \quad \text{s.t.} \quad \ell_\epsilon(\mathbf{b};\mathbf{x}_t) = 0.$$

- Closed-form update scheme:

$$\mathbf{b}_{t+1} = \mathbf{b}_t - \tau_t\left(\mathbf{x}_t - \bar{x}_t\mathbf{1}\right), \quad \tau_t = \max\left\{0, \frac{\mathbf{b}_t \cdot \mathbf{x}_t - \epsilon}{\left\|\mathbf{x}_t - \bar{x}_t\mathbf{1}\right\|^2}\right\}.$$

Passive-Aggressive Mean Reversion (PAMR)

- Also beats every single algorithm?
- $O(n)$ update time, very fast when backtesting

Confidence-Adjusted Mean Reversion (CWMR)

- Utilize the variance of portfolio weights
- Model portfolio *m* as a multivariate Gaussian distribution
  - Mean \mu \in \mathbb{R}^m
  - Covariance matrix \mathbf{\Sigma} \in \mathbb{R}^{m \times m}
    - Diagonal elements is variance and 0 elsewhere
- Define $b_t \in \mathcal{N}(\mu_t, \Sigma_t)$
- Optimization problem:

$$\begin{aligned} (\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1}) = \underset{\boldsymbol{\mu}\in\Delta_m, \boldsymbol{\Sigma}}{\arg\min} \quad & \mathrm{D}_{KL}\left(\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \| \mathcal{N}\left(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\right)\right) \\ \text{s.t.} \quad & \Pr\left[\boldsymbol{\mu} \cdot \mathbf{x}_t \leq \epsilon\right] \geq \theta. \end{aligned}$$

Confidence-Adjusted Mean Reversion (CWMR)

- Closed form solution!!

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \lambda_{t+1}\boldsymbol{\Sigma}_t\left(\mathbf{x}_t - \bar{x}_t\mathbf{1}\right), \quad \boldsymbol{\Sigma}_{t+1}^{-1} = \boldsymbol{\Sigma}_t^{-1} + 2\lambda_{t+1}\phi\mathbf{x}_t\mathbf{x}_t^\top,$$

$$\bar{x}_t = \frac{\mathbf{1}^\top\boldsymbol{\Sigma}_t\mathbf{x}_t}{\mathbf{1}^\top\boldsymbol{\Sigma}_t\mathbf{1}}$$

Online Moving Average Reversion (OLMAR)

- Predicts next price as a moving average in a window

$$\mathrm{MA}_t = \frac{1}{w} \sum_{i=t-w+1}^{t} \mathbf{p}_i .$$

- Predicted price relative:

$$\hat{\mathbf{x}}_{t+1}(w) = \frac{MA_t(w)}{\mathbf{p}_t} = \frac{1}{w}\left(1 + \frac{1}{\mathbf{x}_t} + \cdots + \frac{1}{\bigodot_{i=0}^{w-2} \mathbf{x}_{t-i}}\right),$$

Online Moving Average Reversion (OLMAR)

- Optimization problem:

$$\mathbf{b}_{t+1} = \arg\min_{\mathbf{b}\in\Delta_m} \quad \frac{1}{2}\left\|\mathbf{b}-\mathbf{b}_t\right\|^2 \text{ s.t. } \mathbf{b}\cdot\hat{\mathbf{x}}_{t+1} \geq \epsilon.$$

Robust Mean Reversion (RMR)

- Mean reversion algorithms don't take into account noise or outliers
- The median is a summary statistic that doesn't get as affected by outliers
- RMR estimates the next price estimator using a "median estimator" and a window size

The basic idea of RMR is to explicitly estimate next price vector via a robust L_1-median estimator at the end of t^{th} period—that is, $\hat{\mathbf{p}}_{t+1} = L_1 med_{t+1}(w) = \boldsymbol{\mu}_{t+1}$, where w is a window size and $\boldsymbol{\mu}$ is calculated by solving a optimization [Weber 1929, Fermat-Weber problem],

$$\boldsymbol{\mu}_{t+1} = \arg\min_{\boldsymbol{\mu}} \sum_{i=0}^{w-1} \left\| \mathbf{p}_{t-i} - \boldsymbol{\mu} \right\| .$$

Robust Mean Reversion (RMR)

In other words, L_1-median is the point with minimal sum of Euclidean distance to the k given price vectors. The solution to the optimization problem is unique [Weiszfeld 1937] if the data points are not collinear. Therefore, the expected price relative with the L_1-median estimator becomes

$$\hat{x}_{t+1}(w) = \frac{L_1 med_{t+1}(w)}{\mathbf{p}_t} = \frac{\boldsymbol{\mu}_{t+1}}{\mathbf{p}_t}. \tag{9}$$

- Optimization problem is the same as OLMAR

Homework

This Week

- On Github (will be modified soon)
- Finals are not fun. That’s why you have Spring Break to finish homework too.