acm research.

Build Night 6.

Pattern-Matching Strategies

(and More Code Work)

Team Meeting Agenda.

- Discuss homework & meeting with Dr. Gupta (5min)
- Get everyone on the same page in terms of setup (10min)
- Discuss pattern-matching strategies (10min)
- Discuss code architecture and package structure (5min)
- Code review for outstanding PRs (10min)
- Split up work and implement strategies (30min)
- Review, reflect, and prepare for build night presentation

Project Schedule: Vibing.

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
6. Pattern-Matching Strategies (you are here!)
7. Framework I
8. Framework II
9. Backtesting, Experimentation, and Comparison
10. Poster Work & Presentation Practice

Meeting Discussion.

Setup Time.

Pattern-Matching Strategies.

About These Strategies

- It’s in the name. Find a period in the market where the stock has behaved similarly and use it
- i.i.d = independent and individually distributed. In this context, it refers to each asset in our portfolio if it were a random variable (stats)

Pattern-Matching Framework

1. Sample Selection: Select an index set C of historical PRVs that will be used to predict the next one.
    a. Translation: Get a set of indices of historical price relatives which we'll use
2. Assign each PRV a probability using some criteria.
    a. Could be just straight up 1/|C| or could be something more complex
3. Learn an optimal portfolio by maximizing a utility function $U(\mathbf{b}; C)$ that takes in the current PRV and the index set

$$\mathbf{b}_{t+1} = \arg\max_{\mathbf{b} \in \Delta_m} U(\mathbf{b}; C)$$

ALGORITHM 2: Sample selection framework ($C(\mathbf{x}_1^t, w)$).

Input: $\mathbf{x}_1^t$: Historical market sequence; w: window size;
Output: C: Index set of similar price relatives.

```
Initialize C = Ø;
if t ≤ w + 1 then
    return;
end
for i = w + 1, w + 2, ..., t do
    if x_{i-w}^{i-1} is similar to x_{t-w+1}^{t} then
        C = C ∪ {i};
    end
end
```

Sample Selection Techniques

- Histogram-based approach: discretize each historical PRV into d_L buckets and pick the bucket that the current PRV x_t falls into
- Unclear as to what types of discretization functions may be acceptable

Nonparametric *histogram-based* sample selection [Györfi and Schäfer 2003] predefines a set of discretized partitions, partitions both the latest market window $\mathbf{x}_{t-w+1}^{t}$ and the historical market window $\mathbf{x}_{i-w}^{i-1}$, $i = w+1, \ldots, t$, and finally chooses price relative vectors whose $\mathbf{x}_{i-w}^{i-1}$ is in the same partition as $\mathbf{x}_{t-w+1}^{t}$. In particular, given a partition $P_\ell = A_{j,\ell}, j = 1, 2, \ldots, d_\ell$ of $\mathbb{R}_+^m$ into d_ℓ disjoint sets and a corresponding discretization function $G_\ell(\mathbf{x}) = j$, if $\mathbf{x} \in A_{\ell,j}$, we can define the similarity set as

$$C_H(\mathbf{x}_1^t, w) = \left\{ w < i < t+1 : G_\ell(\mathbf{x}_{t-w+1}^{t}) = G_\ell(\mathbf{x}_{i-w}^{i-1}) \right\}.$$

Note that ℓ is adopted to aggregate multiple experts.

Sample Selection Techniques

- Kernel-based approach: use Euclidean distance to compare two market windows
- If the market window is similar enough (based on some threshold), then we add it to the similarity set

Nonparametric *kernel-based* sample selection [Györfi et al. 2006] identifies the similarity set by comparing two market windows via Euclidean distance:

$$C_K(\mathbf{x}_1^t, w) = \left\{w < i < t+1 : \|\mathbf{x}_{t-w+1}^t - \mathbf{x}_{i-w}^{i-1}\| \leq \frac{c}{\ell}\right\},$$

where c and ℓ are the thresholds used to control the number of similar samples. Note that the authors adopted two threshold parameters for theoretical analysis.

Sample Selection Techniques

- Nearest-neighbor approach: pick *L* nearest neighbors via Euclidean distance

Nonparametric *nearest neighbor–based* sample selection [Györfi et al. 2008] searches the price relatives whose preceding market windows are within the ℓ nearest neighbor of latest market window in terms of Euclidean distance:

$$C_N(\mathbf{x}_1^t, w) = \left\{ w < i < t+1 : \mathbf{x}_{i-w}^{i-1} \text{ is among the } \ell \text{ NNs of } \mathbf{x}_{t-w+1}^t \right\},$$

where ℓ is a threshold parameter.

Sample Selection Techniques

- Correlation-driven approach: find the similarity between two market windows using correlation coefficient

Correlation-driven nonparametric sample selection [Li et al. 2011a] identifies the linear similarity among two market windows via correlation coefficient:

$$C_C(\mathbf{x}_1^t, w) = \left\{ w < i < t+1 : \frac{cov(\mathbf{x}_{i-w}^{i-1}, \mathbf{x}_{t-w+1}^t)}{std(\mathbf{x}_{i-w}^{i-1})std(\mathbf{x}_{t-w+1}^t)} \geq \rho \right\},$$

where ρ is a predefined correlation coefficient threshold.

Utility Functions

- See paper.

Code Architecture.

Code Review.

Homework.

This Week

- On Github Issues!