Traders@MIT

Competition
Sample Case Packet

traders@mit.edu
traders.mit.edu

October 22, 2019

Contents

1 Order Execution

1.1 Overview

For order execution, your task is to synthesize price predictions from various models and use them to trade algorithmically in Python. At a high level, you should profit in this case by:

1. Market making.
2. Assessing trustworthiness of individual models (sources).
3. Synthesizing price predictions to get a more accurate prediction.
4. Using these price predictions to profit.

Your goal is to obtain the highest possible PnL.

1.2 Case Information

You will start the case with an endowment of $100,000.

1.2.1 Price Predictions

At any point in time, for each security, you may receive some news that will give you price predictions for the next 10-50 timesteps. At any time t, you will receive predictions for the times $t + k$, where $k \in [10, 50]$. These predictions will be sent individually, and from different sources. You may receive this news from one source, multiple sources, or no sources at all.

There are no guarantees about the news that each individual news source provides– however, we guarantee that at least one news source will be approximately correct. We also guarantee that news sources will be consistent; that is, if a news source is known to be reliable for a certain time frame, it will continue to be reliable. The noise of any reasonable price estimate is normally distributed.

For any two distinct securities, they will have a fixed correlation coefficient $\hat{\rho}$ (note $\hat{\rho}$ is the same across all pairs of securities). In more mathematical terms, there exists a $\hat{\rho}$ such that given two any sequences of prices $\{a_i\}$ and $\{b_i\}$. Let A be the random variable where $P(a)$ is equal to the probability that given any i, $a_{i+1} - a_i$ is equal to a, and let the random variable B be defined likewise for the sequence $\{b_i\}$. Then, the correlation between A and B is equal to

$$\rho_{A,B} = \frac{\text{Cov}[A, B]}{\sigma_A \sigma_B}$$

where σ_A, σ_B are the standard deviations of A and B, respectively and

$$\text{Cov}[A, B] = \mathbb{E}[AB] - \mathbb{E}[\mathbb{A}]\mathbb{E}[\mathbb{B}]$$

is the covariance of A and B.Then, $\rho_{A,B} = \hat{\rho}$.

1.2.2 Limits

At any point, you can only have a maximum of 100 open orders, and a position limit of 500 shares of any security. Each order must contain between 10 to 1000 shares.

You will also be subject to message limits as listed in the Mangocore API.

1.2.3 Competition Day

Most of the work for this case will be done in advance of the competition. On the day of the competition, we will run the case for 4 rounds for 450 seconds each (7.5 minutes). In between the rounds, you will have time to edit your strategy and your code. In each round, you will be ranked based on your PnL. Your rank will be the average of the ranks for the four rounds.

1.3 Interaction with Mangocore

Here, we will provide some case-specific information about how Mangocore will send price predictions.

All price predictions will be considered a piece of news, to be received by the `onNews` method, with the title of the news being the name of the security followed by the time that the prediction is for. For example, if I'm receiving the price prediction at time 4 from Joe for time 12 for the ticker TRDRS1, the news will look like:

```
{
   "headline": "TRDRS1 12",
```

```
   "source": "Joe",
   "body": "106.01",
   "time": 4,
   "price": 0
}
```

Please make sure that you are not reading `"price"` as the prediction–that refers to the cost of the news itself (which is always 0 in this case), not the prediction. For reference, we will provide a template code that receives the news and stores it.

1.4 Suggested Strategies

We will suggest some strategies here to help improve your bot. You do not have to use these strategies.

1.4.1 Market Making

One possible way to make money is by *market making*, which is buying at a low price and selling at a higher one. However, this runs the risk of adverse selection. Someone who has a better grasp of the market, or better information, could take advantage of an overzealous market maker offering tiny spreads.

Successful competitors should be able to market make when spreads are high.

1.4.2 Using Price Predictions

The first step to using the provided price predictions is to determine which sources are reliable and which are not. Note that some sources are more reliable in the short term, and others are more reliable in the long term.

Luckily, it is easy to test if a price prediction is reliable–just wait a couple of timesteps and see how much the prediction deviates from the actual price.

Since the bot is receiving price paths from multiple sources, and multiple price predictions for a price at a single point in time, it would make sense to use all of this information, rather than blindly trusting a single news source.

1.4.3 Covariance

You can also get more accurate price predictions based off of covariance. Once you have sufficiently many data points, you will be able to calculate the covariance fairly accurately. Upon finding the covariance, you will be able to use price predictions from other securities to enhance your prediction for a given security.