The Challenges of Applying Machine Learning to High-Frequency Trading

Chris Hammond

Susquehanna International Group (SIG)

October 11, 2019

Summary

With the application of machine learning techniques growing in popularity, we will discuss the challenges that arise when using these techniques in the financial industry. Some of these challenges, like choosing a good objective function and avoidance of overfitting are encountered in most applications of machine learning. Other challenges are unique to high-frequency trading. These include, accounting for correlated risk and deploying models in an environment where latency and throughput affect model performance. We will explore these challenges using some toy models.

Overview

- Problem Definition
- Choosing the Right Objective Function
- Training on Data with Correlated Risk Factors
- Performance Considerations
- Other Considerations

Susquehanna International Group (SIG)

Before we begin, who are we?

- Proprietary Trading
- Known for Options Trading
- Values Quantitative Research and Technology

What are we trying to accomplish?

The goal in trading is to make money by knowing the value of assets and trading them for *edge*. If we have the opportunity to buy an asset at a price y and our estimate of fair value is $\hat{y}$, if the cost of trading is c, then our edge is $\hat{y} - y - c$.
One of the goals in machine learning (and statistics) is to find *good* estimates of some target y given some features x. More precisely, there is some loss function L and some function space $\mathcal{H}$ where we are looking to find

$$\min_{h \in \mathcal{H}} \mathbb{E}[L(h(x), y)]$$

These problems seem to line up well . . .

Choosing the Right Objective Function

How do we choose L? It should correspond to what we want to accomplish, and generally also have some nice properties. For regression problems, the most common choice is

$$L(\hat{y}, y) = (y - \hat{y})^2$$

In our trading example, what is the true loss? Let's change notation a little bit so that y is the edge with no cost. If L is how much money we lose relative to the optimal decision, then:

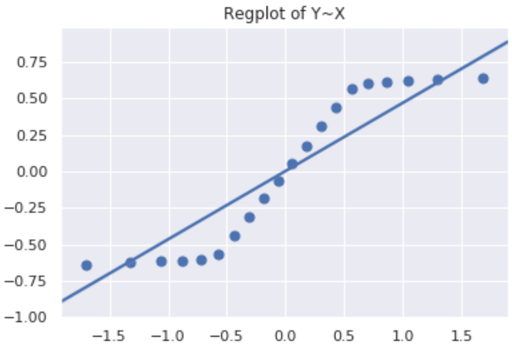
$$L(\hat{y}, y) = \begin{cases} 0, \text{ if } \operatorname{sign}((\hat{y} - c)(y - c)) > 0 \\ |y - c|, \text{ else} \end{cases}$$

or more succinctly

$$L(\hat{y}, y) = \max(0, -(y - c)\operatorname{sign}(\hat{y} - c))$$

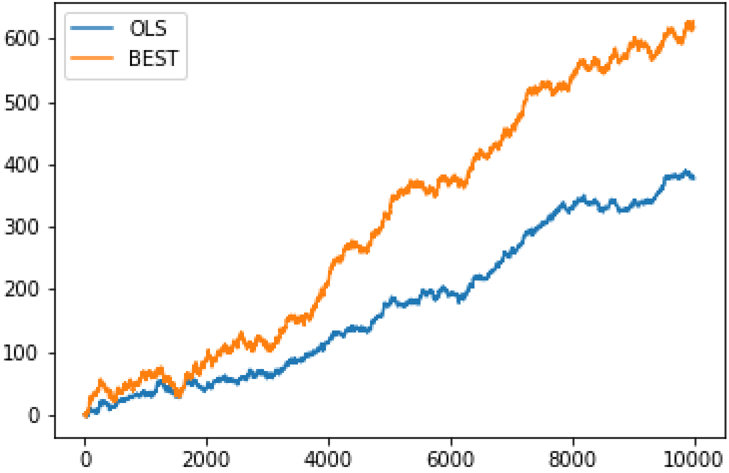
Choosing the Right Objective Function : Toy Model

```
X = np.random.randn(N,1)
def f(X):
return np.sign(X)*np.interp(np.abs(X),[0.0,0.6,2.0],[0.0,0.6,0.65])
Y = f(X)+noise*np.random.randn(N,1)
```



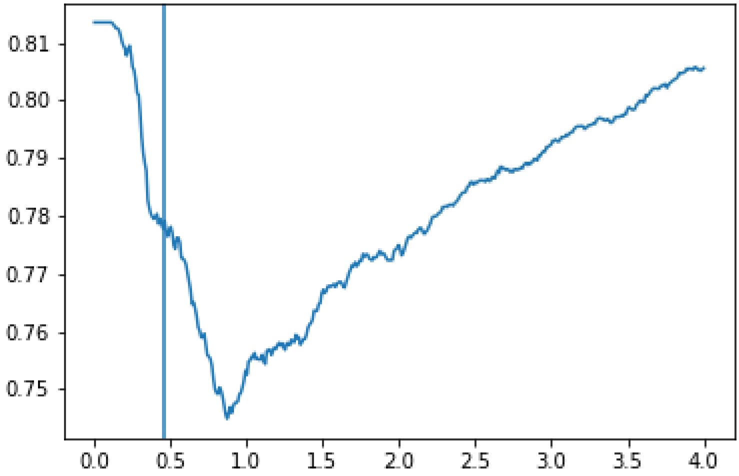
Squared Error Performance

If we fit an OLS regression and look at the PnL compared with the theoretical best



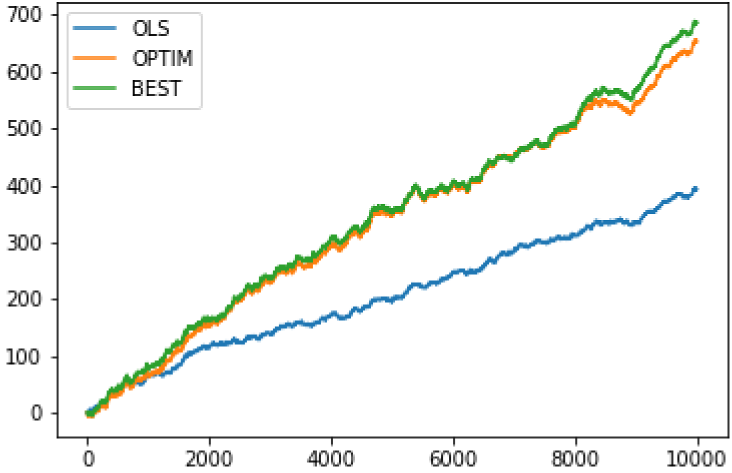
Brute Force Optimization

Note that the coefficient in the OLS regression is about 0.46.



Compare OLS and Brute Force Optimization

An (out-of-sample) comparison yields

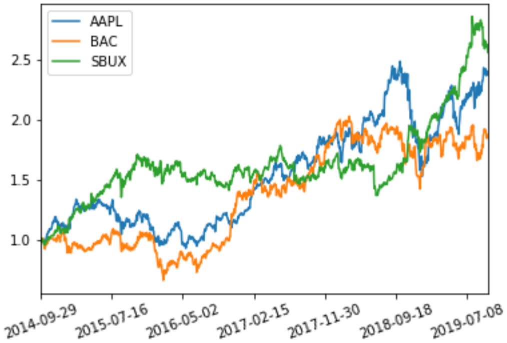


Some Comments on this Example

- In this example, if we train a more complex model, we achieve optimal results with OLS regression.
- Brute force optimization is not a reasonable solution in most situations, since one typically has many more parameters.
- The loss function in this example is not very nice.

Correlated Samples

In finance, the things we are trying to predict are often highly correlated, and may depend on other factors that have some nonzero mean or exhibit some degree of predictability. For instance, daily stock returns.



Modelling Correlated Samples

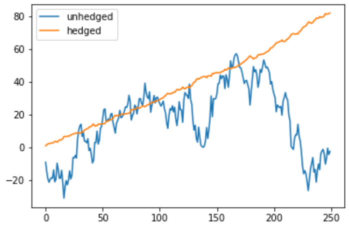
Consider the following scenario. Each day, there is one sample of $z \sim \mathcal{N}(\mu, \sigma_z)$ but there are K samples $y_i = z + \tilde{y}_i$ where $\tilde{y}_i = \beta x_i + \epsilon_i$ and $x_i \sim \mathcal{N}(0, \sigma_x)$, $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon)$ with all ϵ_i, x_i independent. Suppose that you are going to produce estimates $\hat{y}_i$ and trade with no cost.
You could estimate y_i or you could estimate $\tilde{y}_i$.

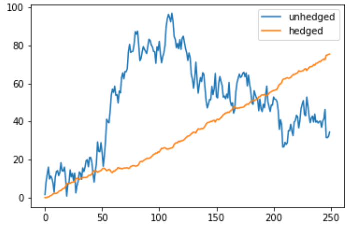
Risk

Consider the ratio of the expected return of a portfolio with the standard deviation, this is the information ratio with a trivial benchmark. Suppose one estimates $\tilde{y}_i$. If K is reasonably large, the average position in z will be small, and if the expected return and volatility of each investment is $\tilde{\mu}$ and $\tilde{\sigma}$, then the portfolio expected return is $K\tilde{\mu}$ and the portfolio volatility is sqrt(K)$\tilde{\sigma}$. so the information ratio scales as sqrt(K).

Comparing Hedged and Unhedged Models

Comparing models trained on hedged versus unhedged returns:





Performance Considerations : Some Easy Math

Consider an ensemble of decision trees. Suppose each tree has a depth of 7 and there are 500 trees. If each evaluation of a tree split takes 2 nanosecond, then each full model evaluation will take 7 microseconds. Suppose you are trading options on 4000 stocks, and each stock has 50 options, so each full update must be run on 200,000 options. With 7 microseconds per evaluation, that is 1.4 seconds.

Why this Matters

- Latency
- Throuput
- Limited/Expensive Real Estate for Hardware

Other Considerations

- The signal to noise ratio is very low relative to other applications of machine learning.
- Because of the nature of the business, there may be siginificant interest in interpretable models.
- Distributions are not stationary.

Next Steps with SIG

- PhD Discovery Day
- Internship Opportunities
- Full-Time Openings:
  - Quantitative Researcher
  - Machine Learning Engineer (Bay Area)
  - Quantitative Sports Researcher
  - Quantitative Developer
- To apply, email PhD@sig.com

What We Look For

- PhD and postdocs studying math, physics, statistics, EECS, or operations research
- Exceptional problem-solving skills
- Solid communication skills
- Practical computing skills (no need to be an expert)
- Finance experience not required