What makes Machine Learning in Finance hard?

Hardik Patel
Machine Learning Engineer, QPLUM

Introduction

- Time series problem involves predicting t^{th} point in the sequence using previous datapoints. Formally, we have $X_1, X_2, \ldots, X_{t-1}$ at time t and the task is to predict X_t.
- Financial time series problem usually involves predicting returns for a security S on day t.
- The model can access historical returns $R_1, R_2, \ldots, R_{t-1}$ and the goal is to predict R_t. Model can also access historical returns for other securities in addition to other historical data like volume, open interest, fundamentals, news and tweets.

Outline

- Factors that make Machine Learning hard:
    - Changing Data Distributions
    - Low Predictive Power
    - High Noise
- What’s the future of Financial Time Series prediction?

Changing Data Distributions

- Let D_{train} and D_{test} be the training set and test set distributions respectively.
- Most machine learning algorithms require these distributions to be same to be effective. ML algorithms work best when the KL-divergence between the two $D_{KL}(D_{train}||D_{test})$ is minimal.
- In reality, most financial time series problems have significantly different D_{train} and D_{test}. A model trained on D_{train} is unlikely to be effective if the production data is going to be completely different than what it has seen during training.

Changing Data Distributions: Reasons

- Past trading patterns failing to materialize in future. This happens because the opportunities have been identified and realized. If the model learns to identify that pattern from the training set, it’s unlikely to work in production.
- Another prominent reason is that the market behavior changes with each major event.
    - Effect of a new war on markets can be very different from the past wars.
    - New policy changes can impact the markets in ways that haven’t been seen in the past.

Changing Data Distributions: Regime Theory

- Market Regime theory is often used to describe the market behavior changing completely.
- However, the issue of Changing Data Distributions is more general compared to Market Regimes.

Low Predictive Power

- Financial time series problems have famously low accuracies. Often, the best possible predictions are just slightly better than random.

Low Predictive Power: Reasons

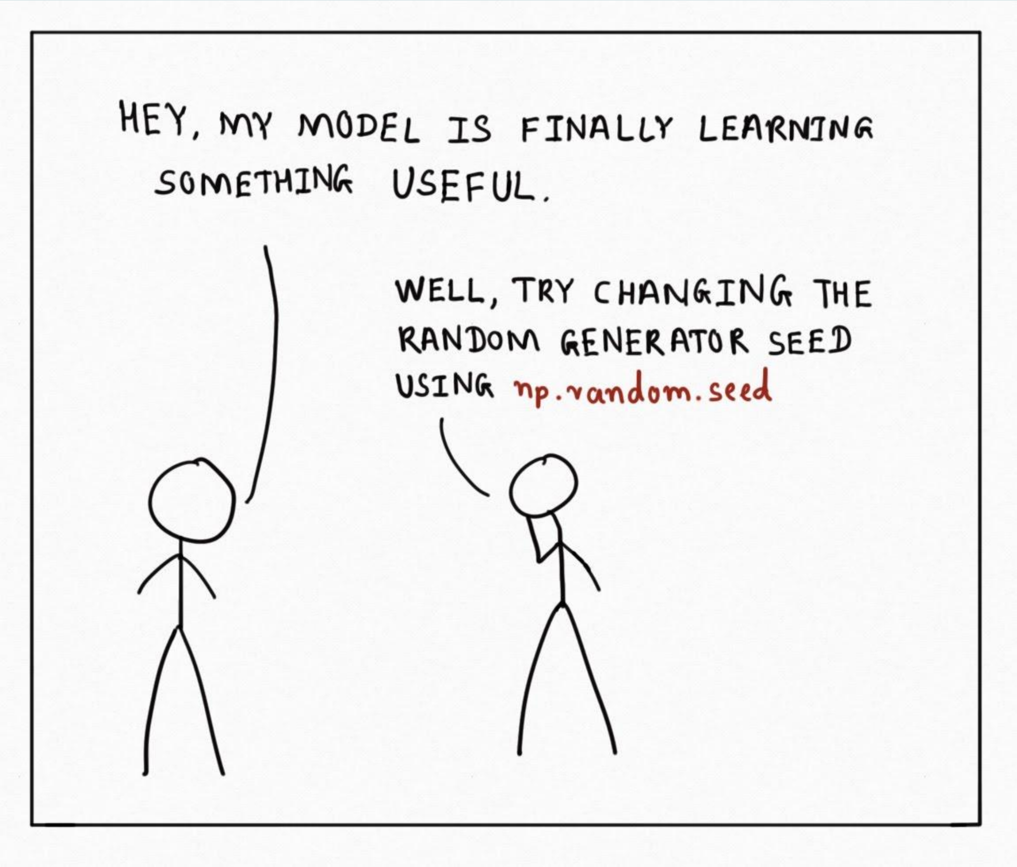
- Historical data inherently have low information content. Examples:
  - A company announces a product tomorrow which goes on to affect its prices - this information is not available in the historical data we possess today.
  - New policy, weather and other updates that affect prices can be missing from X_{t-1} datapoint.

- Therefore, uninterpretable model signals can be significantly harmful at times.

Low Predictive Power: Neural Networks

- Neural networks [Zhang et al. 2016] are good at remembering the data distributions when the predictive power of the dataset is low. Subsequent studies have shown similar properties in more detail.
- It is believed that a neural network does try to identify and learn patterns in the data. But when it fails to find them, it resorts to remembering the data distributions.

High Noise


HEY, MY MODEL IS FINALLY LEARNING SOMETHING USEFUL.
WELL, TRY CHANGING THE RANDOM GENERATOR SEED USING np.random.seed

High Noise: Reasons

- Prices are constantly being influenced by factors working at multiple scales:
  - Short term factors include liquidity constraints, order execution and high frequency trading. For example, if a large investor liquidates their holdings, the price is going to get affected due to liquidity constraints.
  - Medium term factors include news and rumors including product launches and policy change announcements.
  - Examples of long term factors include fundamental data of a company as well as the long-term outlook of the economy.
- When historical data is examined from the point of view of a single factor or a single duration (short/medium/long), it often looks like noise.

Summary (1)

- As a consequence of this being a hard problem, most commercial applications are niche. Additionally, they require:
    - Constant monitoring of the deployed models.
    - Continuous improvements to deployed models.
    - Increased focus on interpretability.

Summary (2)

Potential improvements to overcome unique limitations of financial time series:

- More data from different sources - to overcome “Low Predictive Power”.
- Attention Neural Network Models - to better extract whatever information is available.
- Financial Time Series as a POMDP - to capture causality at various time scales, which would help reduce the problems due to high noise.

Recent Breakthroughs

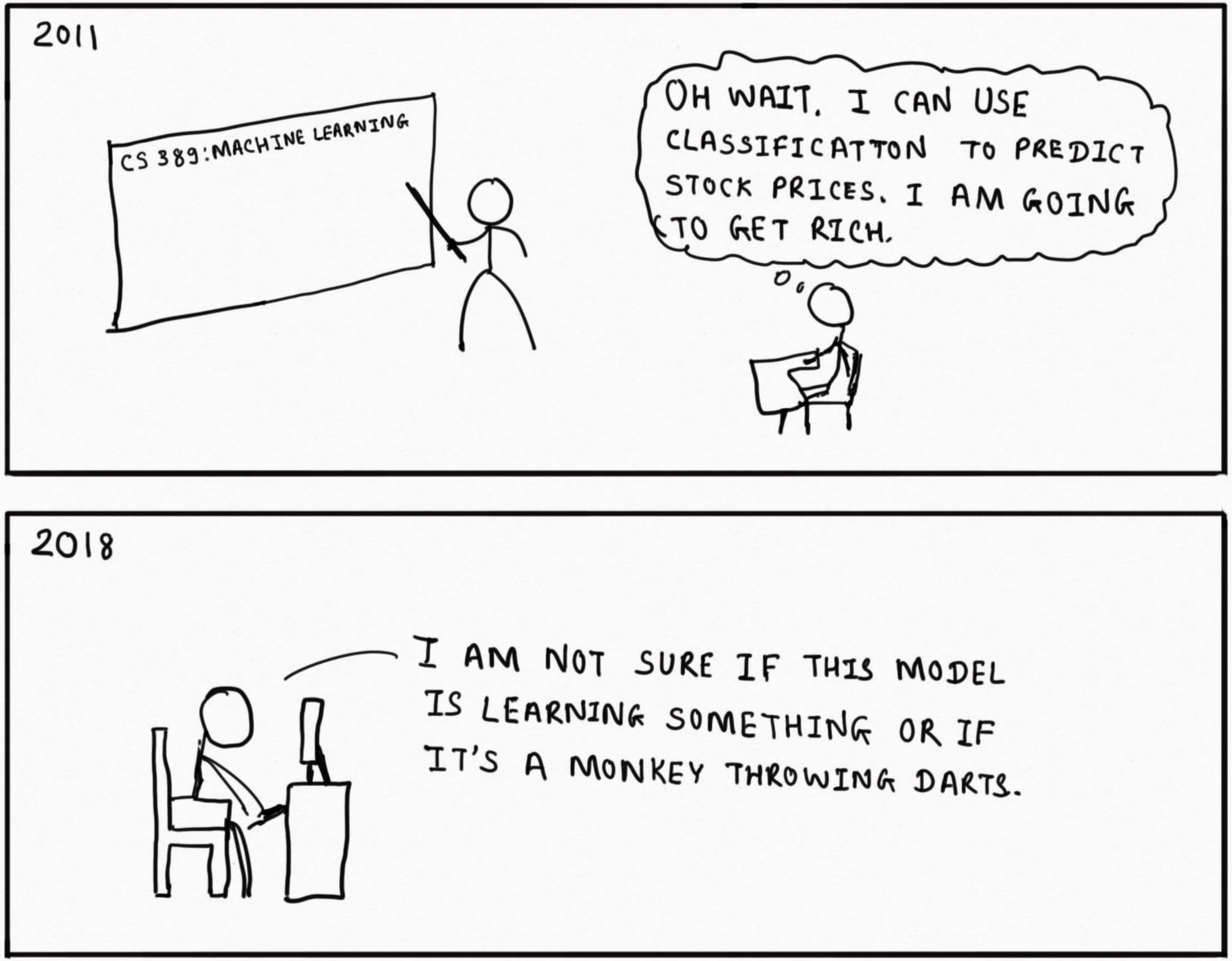
- Rapid improvements are being made to the time series models.
  - A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction - https://arxiv.org/abs/1704.02971
  - Hierarchical Attention-Based Recurrent Highway Networks for Time Series Prediction - https://arxiv.org/abs/1806.00685
- Financial time series as a POMDP problem is gaining more and more attention.
  - Proximal Policy Optimization Algorithms - https://arxiv.org/abs/1707.06347
  - Non-Markovian Control with Gated End-to-End Memory Policy Networks - https://arxiv.org/abs/1705.10993

MDP

- A Markov decision process consists of: a set of states S, a set of actions A, transition dynamics $T(s_{t+1}|s_t, a_t)$ and a reward function R.
- At each time step t, the agent receives a state s_t from the state space S and selects an action a_t from the action space A, following a policy $\pi(a_t|s_t)$. Policy $\pi(a_t|s_t)$ is the agent's behavior. It is a mapping from a state s_t to an action a_t. Upon taking the action a_t, the agent receives a scalar reward r_t, and transitions to the next state s_{t+1}.
- The return $R_t = \sum_k \gamma^k r_{t+k}$ is the discounted, accumulated reward with the discount factor $\gamma \in (0, 1]$. The agent aims to maximize the expectation of such long term return from each state.

Financial Time Series as a PODMP

- Each state S_t follows the Markov property. This makes it possible for the RL algorithms to generate action a_t by only looking at state s_t.
- In practice, most practical scenarios do not hold Markov property. They are often described as Partially Observable Markov Decision Processes(POMDP).
- Financial time series can be viewed as a POMDP.
    - State s_t - historical data for day t,
    - Action a_t - a trade action on day $t+1$
    - Reward r_t - PNL impact on day $t+1$
    - Next state s_{t+1} - historical data for day $t+1$


2011
CS 389: MACHINE LEARNING
OH WAIT. I CAN USE CLASSIFICATION TO PREDICT STOCK PRICES. I AM GOING TO GET RICH.
2018
I AM NOT SURE IF THIS MODEL IS LEARNING SOMETHING OR IF IT'S A MONKEY THROWING DARTS.