Approaching the problem of financial forecasting from machine learning perspective

S. Burc Eryilmaz

PhD candidate, Electrical Engineering Dept., Stanford University

09/24/2015

About me

- 5th year PhD student in Stanford-EE
  - Building brain inspired hardware using ‘emerging’ non-volatile-memory devices
    - Storing ‘weights’ using single memory cells instead of 64-bit rows: more efficient NN hardware
  - How to utilize ‘emerging NVM devices’ on architecture level
    - Making memory persistent?
- As such, for my PhD, I do:
  - Memory device design, circuit design, architecture design

About me

- 5th year PhD student in Stanford-EE
  - Building brain inspired hardware using 'emerging' non-volatile-memory devices
    - Storing 'weights' using single memory cell instead of 64-bit rows: more efficient NN hardware
  - How to utilize 'emerging NVM devices' on architecture level
    - Making memory persistent?
- As such, for my PhD, I do:
  - Memory device design, circuit design, memory system design

So, how did I get into finance?

Timeline of my ML/finance experience

- Took ML class from Andrew Ng
  - Yes, with 800 other people
  - Did a class project for using SVMs on finance data (technical report on ArXiv)

Supervised classification-based stock prediction and portfolio optimization

Sercan Arik, Sukru Burc Eryilmaz, Adam Goldberg

(Submitted on 3 Jun 2014)

As the number of publicly traded companies as well as the amount of their financial data grows rapidly, it is highly desired to have tracking, analysis, and eventually stock selections automated. There have been few works focusing on estimating the stock prices of individual companies. However, many of those have worked with very small number of financial parameters. In this work, we apply machine learning techniques to address automated stock picking, while using a larger number of financial parameters for individual companies than the previous studies. Our approaches are based on the supervision of prediction parameters using company fundamentals, time-series properties, and correlation information between different stocks. We examine a variety of supervised learning techniques and found that using stock fundamentals is a useful approach for the classification problem, when combined with the high dimensional data handling capabilities of support vector machine. The portfolio our system suggests by predicting the behavior of stocks results in a 3% larger growth on average than the overall market within a 3-month time period, as the out-of-sample test suggests.

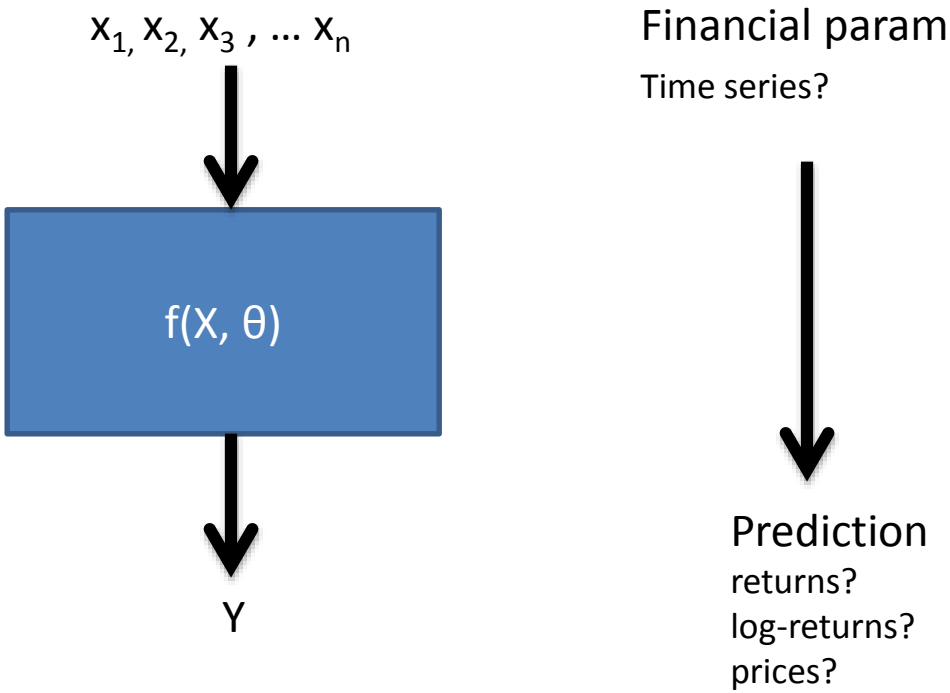
- Quantiacs came up
  - Great opportunity to make some money while having fun
  - Learnt how to be patient: Had to wait 1 year until competition ends
- Got an internship at a hedge fund in NYC

How do I typically approach the problem

- Transform finance data to training data
- Fit your model, but do NOT overfit
- Test your model, be careful with data transformation when predicting
- Your model is weak? Don’t give up, combine multiple models for a better prediction
- Don’t forget the volatility when constructing your portfolio

What does training a model mean?

- We want to find a function that takes some data that we already know, and produces a prediction output

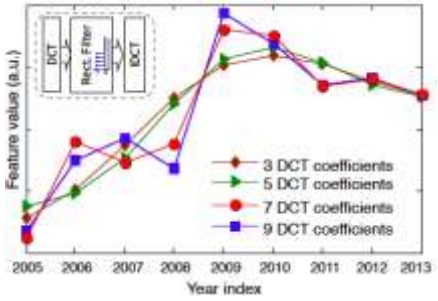


- We use past data to find the parameters θ for guessing Y

Transform finance data to training data

$p_1, p_2, p_3, \ldots p_t \longrightarrow x_1, x_2, x_3, \ldots x_n$

- Data is typically a time series
- Many ML algorithms require training data and training targets (when supervised)
- We might need to do some processing (algorithm dependent)
  - Using the time series directly?
    - Might be a little noisy
  - Produce features from time series
    - Example: moving averages, frequency domain filters
  - Scale/transform your features (within individual features)
    - May or may not work better: experiment
    - Examples: zscore, PCA, autoencoder, whitening…
    - Too aggressive transformation might hurt predictions



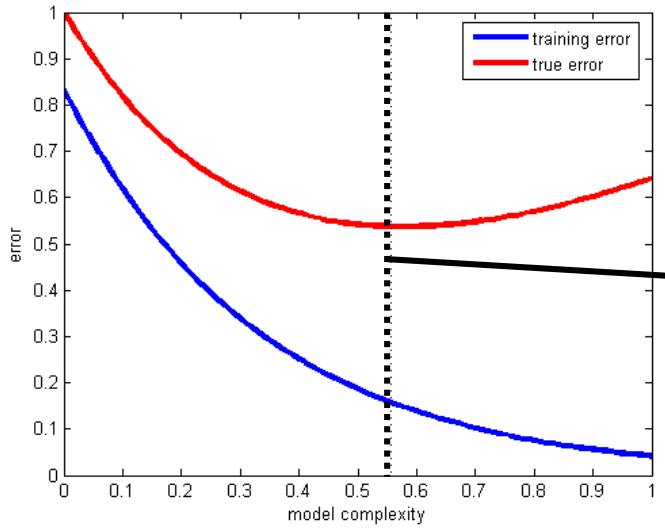
S. Arik et al., [arXiv:1406.0824 [q-fin.ST]]

Fit your model, but do NOT overfit

- Your ultimate goal is to predict the future for which no label is available
  - NOT to obtain the best prediction of the past
- Overfit occurs if model is too complex (too many variables to fit)
- This is what happens when you overfit (I did mistakes in the past):

How to avoid overfitting?

- Use different sets of data to train and validate your model
  - Gives a good idea of how well the model generalizes
- More complex models give better fits for training data, but too complex models will start fitting the noise instead



We need to find the sweet spot

Test your model, be careful with data transformation when predicting

- Use a separate test set to test the performance of your model (10-20% of your data)
  - This will tell you how well your model generalizes
- If you scaled your data in training, make sure to use the scaling factor of the training set when scaling test set
  - Example: If you scale a feature during training from [-10,10] to [-1,1], and if same variable is within [-8,11] in test set, scale it to [-0.8 , 1.1] (algorithm dependent)
  - Example: SVM classification
    - Training and test sets separately scaled to [0,1]: 69% accuracy
    - Test set is scaled with the same factor as training set: 89% accuracy

C.-W. Hsu, et al., Tech. Rep., Department of Computer Science, National Taiwan University

Combining multiple models for a better prediction

- If your model is weaker than you expected but still learns something from data, you can combine it with another model they are not highly correlated
- A simple back of the envelope calculation:
  - Corr(Pred1,Target)=A; Corr(Pred2,Target)=B → Corr(c.Pred1+Pred2,Target)= $(c.A+B)/\sqrt{(1+c^2+2c\sigma_{1,2})}$
    - Assumption: mean and stddev of Pred1, Pred2 and Target are 0 and 1, respectively
- Spending a lot of time to optimize a model or spending less time for multiple models each?

Avoiding overfitting when combining different models

- Final prediction: A.Pred1 + B.Pred2
- How to choose A and B?
  - Treat them as model parameters and train it with a separate set than the data you used for training two models
    - If you use the same set, you might overfit
  - If you do not have enough data, try simply combining them with equal weights

Role of volatility

- While constructing your portfolio, you want to minimize portfolio volatility as well

- In real life, very volatile stocks are typically harder to invest in anyways:
  - Low liquidity
  - You will raise the price when you buy them

Role of volatility

- Use volatility indicators provided by Quantiacs:
  - These indicators had major role in my algorithm that ranked 2nd in kickoff competition
- You can use them:
  - Together with your return predictions to construct your portfolio
  - As a feature to predict returns (may or may not be useful)

Conclusion

- 5 tips on approaching the problem from ML perspective
  - Transform finance data to training data
  - Fit your model, but do NOT overfit
  - Test your model, be careful with data transformation when predicting
  - Your model is weak? Don’t give up, combine multiple models for a better prediction
  - Don’t forget the volatility when constructing your portfolio