Price Prediction Evolution: from Economic Model to Machine Learning

Stanford University

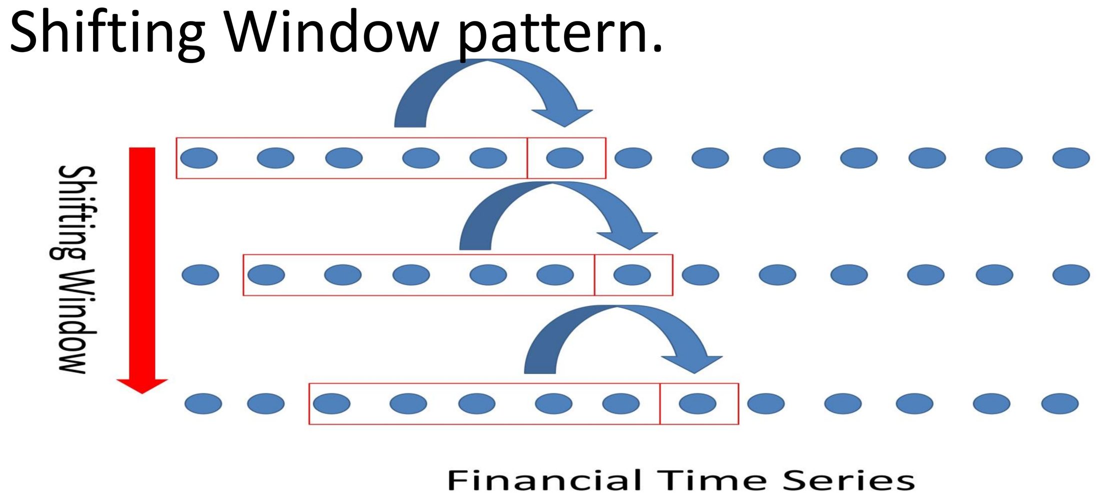
Hugh Ren(hughrxy) Xiangguang Zheng(zhengxg3) Erico Cruz Lemus(ecruzlem)

Problem

Latest decades have seen a rapid growth of the trade market. Part of this rapid growth has been driven because of the result of two significant factors, the reduction of trade barriers and the exponential improvement of technology. Naturally, price forecast haven been tried to predict by multitude experts using different approaches. Initially, given that the trade market and its closing prices can be treated as times series, most of the experts started to use basic statistic tools. Nowadays, experts are using more sophisticated statistic tools, including different machine learning approaches. We focus our work on the Low-frequency trading, trying to predict price and index using various approaches, from macroeconomic features to using ML algorithms.

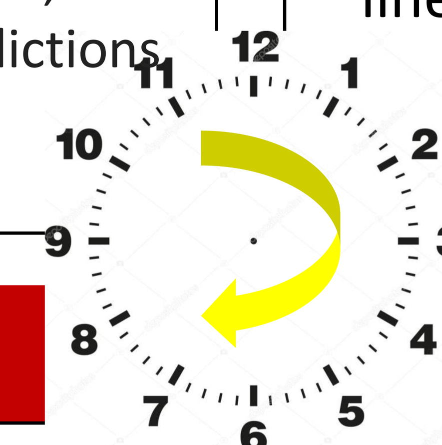
Data and Preprocess

S&P500 price and index datasets are applied to run our algorithms. For LWR intuition, we only take opening stock prices of one company as datasets. For SVR and NN, we develop our dataset by utilizing the stock index, which reflect a comprehensive market tendency. The index information is extracted as four features (max/min/mean/SD) among every five days. All data is inputted by Shifting Window pattern.



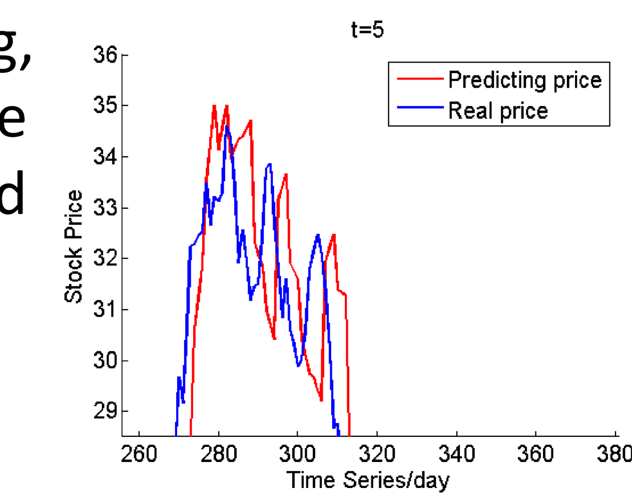
Feature Selection

As a baseline, we first run multilinear regressions adding each macroeconomic features one by one to see the outputs predictions. Total 4 different input feature (univar, bivar, CPI and GDP). GDP are significant predictors of tomorrow's close, even after accounting for the previous 5 day's close as a variable. CPI does not significantly predict tomorrow's close. We believe that averaging the data of the five previous days in order to predict the fifth one, also smooths the data so we could miss the real behavior of the trade market and its data. As we observed, adding more macroeconomic features worsen our predictions. Therefore, we believe that only using closing prices with macroeconomic features, is a good approach for obtaining accurate predictions in the trade market.



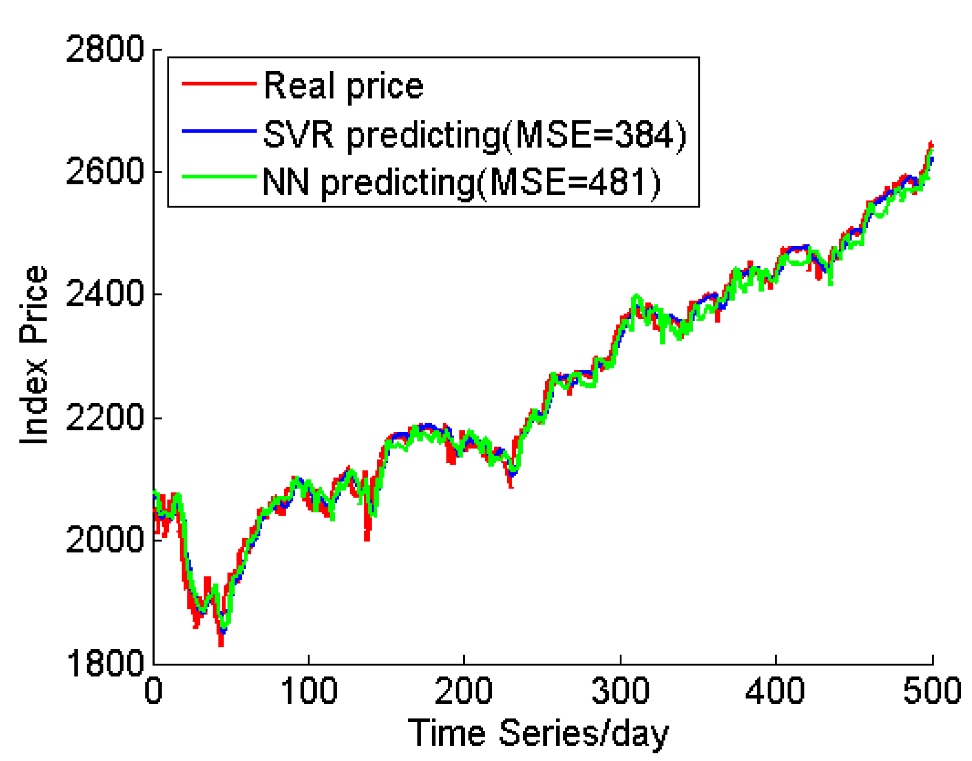
LWR

LWR is utilized to show some intuitions about how one might think and act on the stock market. Let us consider novices in stock market: if giving them money and require them of getting profit in stock market, how they will do? A simple idea is just weightily reviewing recent stock price and tendency, and approximate future price, which is LWR's behavior. LWR present a good intuition of price prediction, but have serious time lag, giving rise to naïve input pattern and linear nature of LWR



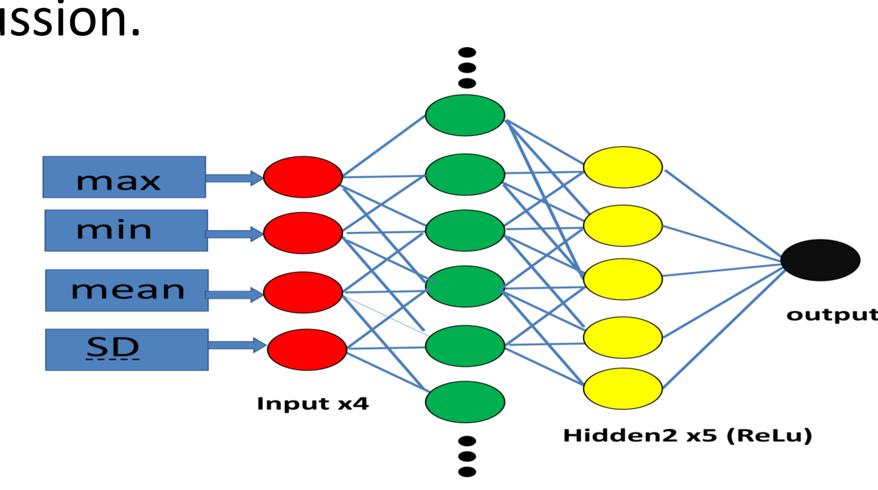
Result and Discussion

SVR and NN are the final evolution version, and it output similar MSE comparing with actual data, while each hold various advantages and drawbacks. First of all, for model itself, NN is heuristic, while SVR are theoretically built. Specifically, SGD cannot guaranteed to find the optimal parameters when NN implementations employ it. People say that NN get stuck in a local minima while SVR don't. But to some extent, NN is outperform than SVR, we could develop other proper neural network to refine model. Second, sensitivity shown in figure, NN always overreact in tiny fluctuation, which implies NN "prefer" to capture signal and endow high weight on it, while SVR behavior stable on stimulation. Considering the index variation is a mild, it is reasonable that SVR is better than NN, but we still expect NN might perform better for fluctuating market.



NN

We used the four preprocessed feature as the input to feed into the neutral network, which has 2 hidden layers and has ReLU as the activation function of hidden layers. And we didn't use any activation for the output layer. To tune the parameters, We used cross validation to adjust my model, where the we split the data set into three: train set, dev set and test set. Dev set is used to adjust my model's topology, including the number of layers, number of neuron per hidden layer, as well as the size of mini batch. The output 500 data with 481 MSE is plotted in same figure in discussion.



SVR

Support vector regression with Gaussian Kernel is a significant machine learning method in data mining, since it overcomes difficulties curse of dimensionality. SVR is developed by SVM and we used it because predicting of stock behavior means forecasting the curve tendency (regression). SVR is developed by adding relaxation variables so as to make SVM "flexible". The initial loss function and restriction could be:

$$\min_{w,b,\lambda_1,\lambda_2} \left[\frac{1}{2} w^T w + C \sum_{i=1}^{n} (\lambda_1^i + \lambda_2^i)\right]$$

$$s.t.\quad y_i - w^T\varphi(x_i) - b \le \varepsilon + \lambda_1^i$$

$$w^T\varphi(x_i) + b - y_i \le \varepsilon + \lambda_2^i$$

$$for\ all\ \lambda_1^i, \lambda_2^i \ge 0$$

We use libsvm in MATLAB and divide data into three groups: 4000(train), 500(dev), 500(test). The dev set optimize two parameters (C in target function and sigma in kernel The output with 384 MSE is plotted in discussion part.

Future Work

If we had more time working on this interesting topic, we might explore the following ways:

1.Train our model based on a more various (fluctuating or abrupt) market

2.Other than comparing every single model. We might combine them with respect to optimized weight parameters.

Reference

[1] B. A. hnaity and M. Abbod. "Predicting Financial Time Series Data Using Hybrid Model." ISA 2016.

[2] M. E. Gerlow, et.al. "Economic evaluation of commodity price forecasting models." IJF 1993.

[3] C. O. Tiong, C.L. Ngo, and Y. Lee. "Stock Price Prediction Model using Candlestick Pattern Feature." IJIDM 2013.