Group: Daniel Rozenzaft, Yaroslava Shynkar, Jonathan Kelaty
Project: Stock Movement Prediction

ML Final Project Proposal

Description

For our final project, we will be using various machine learning algorithms to predict the optimal times (days) to buy and sell stock shares. Ours is an **application project**. In essence, we want to predict local maxima and minima for time series data. Existing literature employs a wide array of methods for local extrema prediction, including (and certainly not limited to) support vector machines (SVMs), random forest (RF), reinforcement learning (RL), artificial neural networks (ANN), and long short-term memory neural networks (LSTMs). We plan to implement several of these and compare the investment strategies that they produce using the foundational evaluation metric of our field: profit. We would also like to quantify the performance-boosting impact of sentiment analysis techniques on company news or tweet data; we would integrate this as another feature for our models. For our novel approach, we would like to devise a high-performing, machine learning-informed portfolio allocation strategy that adheres to investor capital and risk tolerance constraints.

Roles:

The division of the first phase of our project will be based on algorithms: each of us will work simultaneously and (mostly) individually on incorporating one of the following approaches with our data: RL, SVM, RF, ANN, and LSTM. After those are completed, we will collaborate on merging our algorithms with the NLP and devising user-readable output for the portfolio recommendation component, fleshing out those elements as much as we can before the final deadline.

- Daniel
    - Individual assignment tentatively set as building and training RL agent
        - Use *Yang et al.* for performance benchmark
    - Contribute to NLP component (sentiment analysis on tweets)
        - Reconcile project ideas with Tweet Catcher library methods and other existing NLP techniques
- Yaroslava
    - Individual assignment tentatively set as building LSTM/ANN
        - Compare to results obtained by *Patel et al.* and *Nabipour et al.*
    - Contribute to NLP component
        - Focus on theoretical basis from *Bing et al.*
- Jonathan

- Individual assignment tentatively set as building SVM and RF
  - Compare to results obtained by *Hiba Sadia et al.*
- Contribute to NLP component
  - Emphasis on coding and evaluation using results from *Bing et al.*

Related Topics

1. LSTM (ANN) - long short-term memory is an artificial recurrent neural network architecture that can be used for classifying and making predictions using time series data. Since LSTM can store past information (previous stock prices), we hope to use it to predict the local minima and maxima in future prices.

2. Random Forest (RF) constructs decision trees, which allow them to output classifications based on the mode or the average of the classes that make up a tree. Randomly generating decision trees and averaging their values will help us build out a model that does not overfit on one (type of) stock or one time period.

3. Support Vector Machines (SVM) build out a decision function using support vectors—a subset of training samples—and produce binary classifications. Our SVM will predict whether the stock price will decrease or increase on a given day. Each "direction change" implies a local extreme: a point at which we will buy or sell.

4. Reinforcement Learning (RL) is predicated on maximizing a reward function, which we will develop by measuring the accuracy of our extrema predictions and by applying our profit metric.

5. Natural Language Processing (NLP) involves quantifying and analyzing features of textual data. We will use NLP methods to explore the relationship between relevant tweet sentiments and concurrent stock movements.

Data

The best freely available historical stock price data set that we could find is this one from Kaggle. It is a CSV file consisting of daily price data from over 5,800 stocks, dated between 1980 and April 2020. We will use only a small segment of it (probably less than five years worth) to test and train our models. If we can get access to more detailed proprietary data—which would ideally have more features or include hourly prices—then we may look to use that instead.

As for the tweet sentiment analyzer, we will use Tweet Catcher, a Python package that accesses historical tweets based on user-entered keywords. We will use company names or stock tickers as keywords and analyze the top results.

Novel Approach

For our novel approach, we would like to weigh the daily price changes based on the stock's average volatility. This could help to improve performance by predicting major ebbs and flows before they happen. Some simple trading strategies involve selling or buying after a predefined jump or fall in the stock price (a reasonable choice is 10%). We can expand upon this by weighting a set percentage by the average volatility, encouraging the algorithms to avoid buying during short price falls (hoping for a larger decrease), and conversely, avoid selling during short price jumps (hoping for a larger increase).

Our more ambitious novel idea involves creating user-specified parameters for acceptable risk tolerance and capital constraints. Higher risk tolerance would permit less frequent buying and selling, in which the investor would hope for larger price decreases or increases, respectively, as marginal (10 percent?) changes occur. Investors with access to more funding may be drawn to higher-risk strategies, while investors with less funding may prefer a more conservative approach. We can limit purchases and adjust the number of shares of each stock that are bought and sold based on the constraints (the risk tolerance and the size of the investor's account), allowing our algorithm to make comprehensive portfolio recommendations. None of our reference papers appeared to devise specific portfolio allocation strategies. We can superimpose the stock price charts and local extrema on a single time series graph, which would tell the investor when to buy or sell a certain amount of shares of each stock.

Timeline

We would like to complete our "individual" algorithm assignments within four weeks. If we can complete our implementations of the selected algorithms and compare their performance evaluations to those of our references and to each other by late November, then we will use our remaining weeks to apply tweet sentiments as features in our existing models and incorporate our novel performance boosting and portfolio allocation strategies. Our current timeline is:

<u>Week 1:</u> Attempt to apply familiar algorithms (SVM, RF) to dataset, look further into methods and variables provided by selected libraries (Tweet Catcher, etc.)

<u>Week 2 (November 1-8) :</u> Yaroslava will create the initial version of ANN and LSTM code, define the key aspects of novelty in data mining applicable for the project. Daniel will start to train the RL agent on the dataset. Jonathan will finish the SVM/RF code and gather Twitter data for the NLP component.

<u>Week 3:</u> Flesh out code from Week 2 (LSTM/ANN for Yaroslava, RL for Daniel, Twitter EDA for Jonathan), run preliminary NLP analyses on Twitter data

<u>Week 4:</u> Complete LSTM/ANN and RL code, work collectively to complete Twitter sentiment analysis and look for ways to integrate it with the other algorithms

Week 5: Fix any bugs or issues in the code, begin evaluation comparison for each algorithm and its related paper(s), explore novel performance boosting and portfolio allocation strategy

Week 6: Complete evaluation, compare with reference literature, put together basic slides for final presentation

Week 7: Polish final presentation, work on written report

Demo

For our demo, we plan to display and compare the results (predicted local minima and maxima) generated by each of our models. We will include the total return yielded by following the investment strategy informed by each model. Finally, we will compare the performances of each algorithm on our data (using standard and trading-specific evaluation metrics) to those reached by the implementations in our reference papers. We would also like to present an example of a recommended investment strategy over a set period of time, as informed by our best-performing models: we would not want to offer suboptimal investment advice!

Evaluation

Our goal is to develop a trading strategy that will maximize return for a stock trader over a specific period of time. Thus, our models will be evaluated on how much profit would be generated by buying at their identified local minima and selling at their identified local maxima. Our algorithms will be measured against each other, as well as against the actual maximum profit (calculated from the global minimum and maximum) over a specified time period. Separately, we will evaluate the individual performance of our models on our dataset using standard evaluation metrics (F1 score, ROC curve, etc.), which we will then compare to the metrics achieved by corresponding implementations in our reference papers. We will conclude by measuring the impact of the sentiment analysis and the novel volatility weighting approaches on performance, and by comparing these "enhanced" models' effectiveness on stocks of different sectors. *Bing et al.* found that sentiment analysis performed best on media and tech stocks, but this conclusion was based on 2014 tweet data; we would like to see if this pattern continues to hold in the present day.

References

Bing, L., Chan, K. C. C., Ou, C. (2014). Public Sentiment Analysis in Twitter Data for Prediction of a Company's Stock Price Movements, presented at IEEE 11th International Conference on e-Business Engineering, Guangzhou, 2014. Guangzhou: IEEE Publishing. https://ieeexplore.ieee.org/document/6982085.

Hiba Sadia, K., Sharma, A., Paul, A., Padhi, S., Sanyal, S. (2019). Stock Market Prediction Using Machine Learning Algorithms. *International Journal of Engineering and Advanced Technology*, *8*(4), 25–31. https://www.ijeat.org/wp-content/uploads/papers/v8i4/D6321048419.pdf.

Nabipour, M., Nayyeri, P., Jabani, H., Shahab S., Mosavi, A. (2020). Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis. *IEEE Access*, *8*(1), 150199–150212. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9165760.

Patel, J., Shah, S., Thakkar, P., Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268. https://www.sciencedirect.com/science/article/abs/pii/S0957417414004473

Yang, H., Liu, X., Zhong, S., Walid., A. (2020). Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy, presented at ACM International Conference on AI in Finance, New York, 2020. New York, NY: Association for Computing Machinery. Available at SSRN.