Stock Portfolio Selection using Learning-to-Rank Algorithms with News Sentiment Summary

Blake Hillier
Advanced Big Data Analysis

March 3, 2020

1 Introduction

This paper is a summary of *Stock Portfolio Selection using Learning-to-Rank Algorithms with News Sentiment* by Anqi Liu, Qiang Song, and Steve Y. Yang. This paper attempts to use two common ranking algorithms, ListNet and RankNet, with a new metric for sentiment analysis: sentiment shock. They claim their algorithms perform better than the S&P500 index, as well as produce a better sharpe ratio for both high and low volatile environments.

2 Learn-to-Rank Algorithms

These algorithms attempt to rank a set of items (typically documents) using one of three methods. The first method is pointwise, where they predict the exact rank for each item. The second is pairwise, where they rank each item in relation to other items. The last method is listwise, where it creates a list based on item ranks and then uses this list to learn how to rank. They are typically unsupervised, causing overfitting and high varience to be a problem.

3 Sentiment Indicators

Instead of using common models to estimate an articles sentiment, they have created their own equation:

$$S_{sentiment} = relevance(pos - neg)$$

where relevance is how relevant the article is to the stock, pos is the probability the sentiment is positive, and neg is the probability it is negative

3.1 Sentiment Shock

Immediate news can cause people to buy or sell a stock when released. To capture this, the paper uses the formula

$$S_{shock}(t) = \frac{S_{sentiment}(t) - \mu(t-N, t-1)}{\sigma(t-N, t-1)}$$

where $S_{sentiment}$ is the sentiment rating for week t, and $\mu(t-N, t-1)$ and $\sigma(t-N, t-1)$ are the average and standard deviation sentiment rank from week $t-N$ to $t-1$.

3.2 Sentiment Trend Score

The overall market attitude towards a stock typically affects peoples strategy for long-term strategy. The authors believe the change in sentement can paint a good picture of the sentiment trend, and use the following formulas to capture it:

$$S_{trend}(t) = \sum_{i=t-N}^{t-1} \Delta S_{sentiment}(i)$$
$$\Delta S_{sentiment}(i) = S_{sentiment}(i) - S_{sentiment}(i-1)$$

4 method

The paper proposes to turn ListNet and RankNet into learning-to-rank algorithms instead by giving them past rankings based on returns. For each week, they form a list of stocks with both sentiment scores, the 1-week leading return, 1-month leading return, 1-week leading average sentiment, and 1-month leading average sentiment, and ranks the stocks by their weekly return. They rank stocks based on their quartile, giving the top 25% the rank 4 and the bottom 25% rank 1. Both models calculate the probability of a stock belonging to a rank, but they each use different approaches to do so. RankNet applies a logistic function to the difference between the rank of two stocks

$$P_{ij} = \frac{e^{s_i - s_j}}{1 + e^{s_i - s_j}}$$

while ListNet calculates the probability

$$P(R_k(i_1, \ldots, i_k)) = \prod_{j=1}^{k} \frac{e^{s_{i_j}}}{\sum_{l=j}^{n} e^{s_{i_l}}}.$$

Both models use a modified cross-entropy function as their loss function:

$$C_{RankNet} = -\sum \bar{P}_{ij} \log|P_{ij}| + (1 - \bar{P}_{ij}) \log|1 - P_{ij}|$$
$$C_{ListNet} = -\sum \bar{P}(R_k(i_1, \ldots, i_k)) \log|P(R_k(i_1, \ldots, i_k))|$$

as well as gradient descent for their update function.

5 Experiment

The authors used 512 stocks based on two criteria:

1. The stocks must be in the top 1000 stocks with the highest average trading volume

2. The stocks must have at least one news article per week

Both models use 1 hidden layer, 10 hidden nodes, and a learning rate of 0.005%. They pick the number of training iterations based on the Normalized Discounted Cummulative Gain method. To determine the look-back window for the sentiment values, they use the Spearman rank correlation. To test the algorithms, they used 2 strategies: Long-only and Long-Short. They then tested the algorithms on data from 2006 to 2014, and found the returns to be better than the S&P500. They also tested it on high and low volatile windows of the market and found it still performed well, indicating how robust it is.

6 Conclusion

This is an interesting way to approach trading as opposed to the typical forcasting model, and could be a great portfolio managment tool when deciding what stocks should be in the portfolio. It could also be used to filter down what stocks should be forcasted to better maximize your returns. I would be curious if it would perform better with XLNet instead of their custom sentiment function?