BLACK-LITTERMAN ASSET ALLOCATION VIA MARKET SENTIMENT VIEWS

Xiangyu Meng

Rui Zhong

Mentor: Yanjun Li

December 2019

New York, New York

Abstract

In this project, we investigate the role of market sentiment in the equity portfolio construction problem. We propose to compute the sentiment time series from social media with the help of natural language processing methods. Long short-term memory is used to formalize sentiment information into market views. Market views are integrated into modern portfolio construction theory by using the Black-Litterman model. We analyze the performance of this asset allocation method from many aspects and compare it with a benchmark portfolio. The experimental results show that this model outperforms the benchmark portfolio.

Keywords: Black-Litterman Model, LSTM, Sentiment Analysis

Introduction

Financial markets are among the most complex and chaotic dynamic systems in human history. Many factors could influence the stock price change. Recently, with the development of the Internet and social media, market participants' views and moods have more influence on the stock market. Investors' new information and beliefs could help us capture the price movement of the stocks.

Another reason we incorporate the market view is the shortcoming of traditional portfolio construction theories. Much technical analysis relies solely on mining patterns of past price series while ignoring the public moods. What's more, there are many drawbacks of mean-variance asset allocation methods, such as highly-concentrated portfolio large long or short positions, the input-sensitivity small change in expected returns will result in large weight change and estimation error maximization. a new model and new kinds of factors are needed in portfolio constructions.

The Black-Litterman asset allocation model, created by Fischer Black and Robert Litterman, is a sophisticated portfolio construction method that overcomes the problem of traditional Mean-Variance Asset Allocation. The market views are formed computationally from the sentiment time series as a prior belief of the investor.

The rest of this report will be organized as follows: the background of modern portfolio construction theory and the Black-Litterman model, asset allocation methods, data collection, sentic computing, LSTM time series prediction, portfolio profit and loss and results.

Asset Allocation Methods

A brief introduction to Mean-Variance model

The most basic model in portfolio construction is the Mean-Variance model. This model is based on the trade-off between portfolio returns and the risk taken by the portfolio investors.

The concept of an efficient portfolio was brought by Markowitz in 1952. There will be N assets in the portfolio and each asset is assigned with weight w_i the portfolio return would be the weighted average of the expected return of every asset in the portfolio. The efficient portfolio meets the following conditions.

$$\text{Maximize } U = w^T\mu - \frac{1}{2}\lambda w^T \Sigma w$$

w means the weights of the assets, μ is a vector of excess returns,Σ is the variance-covariance model of excess returns. λ is the risk aversion factor.To get the maximized weights, we could take the first derivative on U and make it to zero:

$$\frac{\partial U}{\partial w} = \mu - \lambda \Sigma w = 0$$

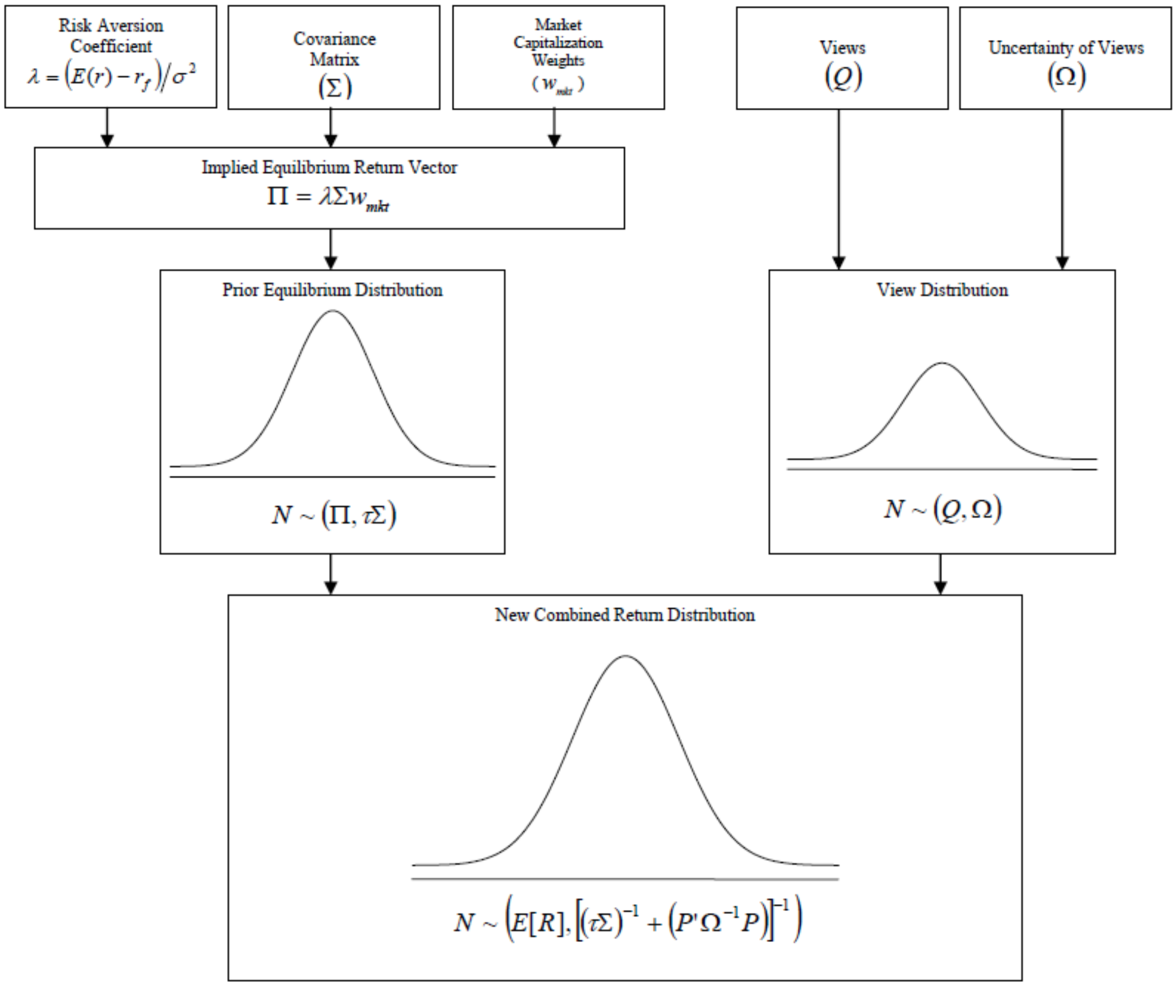
And we could get:

$$w = (\lambda\Sigma)^{-1}\mu$$

In this model, both μ and Σ could be calculated from the historical price.

Black-Litterman model introduction

The big picture of Black-Litterman model could be shown as the following graph:



** The variance of the New Combined Return Distribution is derived in Satchell and Scowcroft (2000).*

According to the above model, we could get the market equilibrium implied return by replacing the weights with market capitalization weight w_{mkt} and the market equilibrium implied return would be:

$$\Pi = \lambda \Sigma w_{mkt}$$

For Black-Litterman model, we need to quantify investor 's view with three matrices(vectors):

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} \qquad Q = \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_n \end{bmatrix}$$

$$\omega_i = \tau P_i \Sigma P_i$$

$$\Omega = \begin{bmatrix} \omega_1 & 0 & 0 & \cdots & 0 \\ 0 & \omega_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \omega_n \end{bmatrix}$$

τ:A scalar number indicating the uncertainty of the CAPM distribution (It is usually within the range of 0.025-0.05)

P:A matrix with investors views; each row a specific view of the market and each entry of the row represents the weights of each asset (KxN matrix) K is the number of views and N is the number of assets

Q:The expected returns of the portfolios from the views described in matrix (Kx1 vector)

Ω:A diagonal covariance matrix with entries of the uncertainty within each view (KxK matrix)

and as described in previous section.

we can compute the posterior variance matrix, which will be used to compute the new covariance matrix. The new covariance matrix:

$$M = [(\tau\Sigma)^{-1} + P^T\Omega^{-1}P]^{-1}$$

$$\Sigma_p = \Sigma + M$$

With the new covariance matrix, we can then calculate the new portfolio weights:

$$\omega = (\delta\Sigma_p)^{-1}\Pi$$

For each day, we could get the prediction for the next day's portfolio weights and we could adjust our position every day. In this project, we used the absolute view so the P would be an identity matrix with dimension of number of assets. Q would be the prediction of excess return for each asset in the portfolio every day. LSTM and NLP are used here to get the Q vector for each day.

Data Collection

The market views require summarizing sentiment from a great deal of textual data.In this study, we collect the opinion messages from StockTwits, which is a popular social network for investors and traders to share financial information. Besides, we obtain the historical closing prices of stocks, market capitalization and daily trading volumes from yahoofinancials(a python package for stock data gathering). We investigated a time period of a month from 2019-10-30 to 2019-11-27. We choose Twitter, Apple, Goldman Sachs, Amazon and Boeing as the assets in our portfolio. We collected the tweets of those stocks from StockTwits.

We used selenium to scrape the twits. For each day and for each stock, we collected hundreds of tweets published by many people. For the tweets on Saturday and Sunday, we treated them as the tweets on next Monday because the effect of market views could only influence the stock price of business days.

Our dataset comprises 19293 messages for Apple, 9828 messages for Amazon, 6505 messages for Boeing, 5570 messages for Twitter and 1538 messages for Goldman Sachs.

Sentic Computing

The quality of sentiment time series is obviously critical, because the data is later employed in the model training of estimating expected return Q. Sentic computing is the state-of-the-art framework that enables sentiment analysis of text not only at document or paragraph level, but also at sentence, clause, and concept level. In contrast to the statistical approaches, sentic computing combines both knowledge-based polarity inference and a back-up machine learning technique. A basic statistical approach counts the positive and negative words in a sentence; however, the sentence structure is not taken into account. By averaging the word polarities, positive and negative words will nullify each other, which brings about difficulties for analyzing sentiment in complicated contexts. Sentic computing mainly leverages a concept-level knowledge base termed SenticNet, a common- sense knowledge base of 100, 000 concepts, and sentic patterns, a group of linguistic rules to explicitly catch the long-term dependency in texts. First, multiple relation tuples are extracted from the sentence. with the Stanford typed dependency parser. Then, a semantic parser further extracts concepts. We look up the concepts from SenticNet, and trigger sentic patterns to process the relations and intrinsic polarities of these concepts. If the concepts are not in SenticNet, the method resorts to a classifier built by machine learning. Here is an example for Sentic computing:

Example : I had a feeling $AAPL would go down, but this is stupid

The preprocessing of this sentence will need to know that "$AAPL" refers to "Apple company" and completes the missing period at the end of the sentence. However, the interesting part is that by denying his own previous opinion, the speaker actually advocates his bullish mood of Apple company and labels. Sentic computing would first identify the concept "go_down" from SenticNet, which is negative. This polarity will be passed through a nominal subject relation to "Apple company", and the relative clause modifier relation to "feeling". Note that this whole structure and "stupid" are linked by an adversative but-conjunction, thus the sentic pattern "negative but negative => positive" is triggered, giving the overall sentence a positive polarity.

LSTM Time Series Prediction

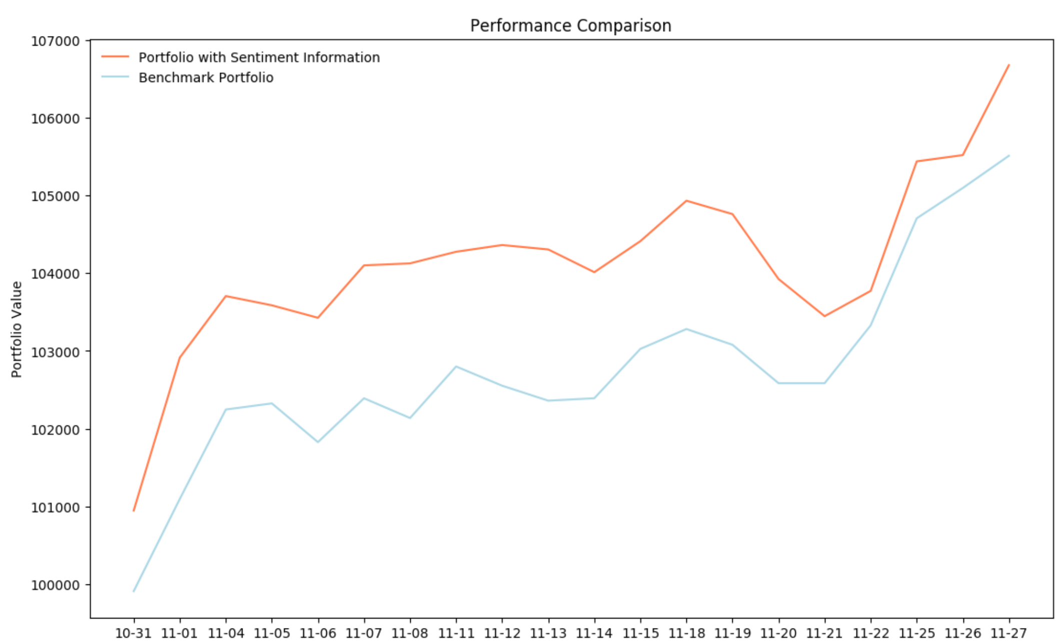
To estimate expected return Q, we use vanilla LSTM here. Long Short-Term Memory (LSTM) have gained lots of attention in recent years with their applications in many disciplines including computer vision, natural language processing and finance. Deep learning methods are capable of identifying structure and pattern of data such as non-linearity and complexity in time series forecasting. The expected return has the most salient relation to the market sentiment. Our hypothesis is that there exists a responding strategy to surf market sentiment that statistically makes profits (generates alpha). Assuming the Black-Litterman agent uses the past price series (p t,k) and trading volumes (v t,k) to empirically form and update the expected return of their views, we further use the current time market sentiment on assets (s t) as a prior. $Q_{t-1}$is the model forecasting of the previous state and it is a guideline for the investor's expected returns.

Portfolio Profit and Loss

After getting the weights for each day, we can get the profit and loss of the portfolio from 2019-10-30 to 2019-11-27. We need to compare the performance of the portfolio with information from StockTwits with benchmark portfolio. Here we choose the equal weighted portfolio as the benchmark portfolio. The initial investment for each portfolio is 100000 dollars. To measure the performance of investments, we used Sharpe ratio and annualized average return.

Results

Here is the portfolio comparison between portfolio with sentiment information and benchmark portfolio:



From the above graph we could know that the portfolio with sentiment information always outperform the benchmark portfolio. The annualized average return for portfolio with sentiment information(portfolio A) is 73.235% and for benchmark portfolio(portfolio B) is 72.366%. However, the portfolio A is more volatile than the portfolio B. The Sharpe ratio of portfolio A is 6.75 and the Sharpe ratio of portfolio B is 8.2252. Portfolio B has a higher Sharpe ratio.

Our recommendation is that: for investors who would like to pursue a high return, portfolio A would be a better choice because portfolio A could yield higher returns. For those investors who would like to keep a balance between risk and return, other kind of portfolio construction strategies may be better.

Review

As we have shown the results above, the asset allocation method which includes the sentiment of the market outperforms the benchmark which only considers the price and volume. During the preparation of data, we have collected tweets that contain emojis, which are also great representation of people's attitude towards stocks performance. However, we were not able to include the sentiment analysis of emojis due to time issue. In the future, we would like to include emojis in our model to construct an image that is more representative of the market sentiment.

References

Xing, Frank Z. et al. “Intelligent Bayesian Asset Allocation via Market Sentiment Views.” (2018).

Black Litterman Asset Allocation Framework - 01Basic Black Litterman