Prediction of asset price using news corpus data in a time series analysis

Arka Chakraborty
Business and Analytics
University of Virginia
Charlottesville, VA, USA
arka@law.virginia.edu

Abstract—*Millions of news article are generated each day in various topics like economy, business, and politics. Although there are lots of noise and redundancy, certain news definitely influences specific asset prices and human behavior related to the asset prices. In this research project, the models try to filter out the noise, identify the key issues in daily news in a time sensitive manner, and understand the relation between news sentiment and asset price.*

Keywords - topic modeling, sentiment analysis, time series, machine learning, big data

I. INTRODUCTION

There have been several efforts in the past to study the impact of news and information on financial asset price. The research project wants to add some vital aspects to the overall effort. One of the key highlights of this approach is that the machine identifies the key issues/topics from the huge corpus of news without the help of human guidance. The process avoids any human bias and will get better over time. Secondly, all the natural language processing (NLP) models work on time sensitive data that facilitates back testing and avoids look ahead bias.

II. HIGH LEVEL DESIGN

Overview of time series data

Time series data of topics/entities, weights, and sentiment scores are developed. These are hourly time series. For each time period, a window (w) of 270 previous hours is considered. It takes into account the last 30 business days, the daily hours stock market is open, and one hour of pre and post market. Each data point in the time series contains data for the duration between time (t – w) and t where t stands for current time/hour and w represents the window size. The window size can be changed later based on model performance. With every moving hour in the time series, the window slides by 1 hour.

Data Collection

Crawler downloads new articles every hour. The article database stores article_id, date, time, and article content.

Hourly article processing

For all the NLP related tasks, a function is called every hour. The function takes two parameters - current time/hour and window size. The parent function can be broken down to multiple child functions if required.

Steps in the natural language processing (NLP) processing of news data

- Articles are selected in the time duration between t – w and t.
- Topic modeling is performed on the corpus to identify the key topics.
- Top N topics are selected.
- Topics are a probabilistic distribution of keywords/entities. From each selected topic, N entities are selected. Each entity has a probability% which is treated as weight for the entity.
- Based on the top N entities, related news are selected/parsed from the article corpus.
- Sentiment analysis is performed on each piece of news selected.

Details of natural language processing (NLP) models used

- For topic modeling, latent Dirichlet allocation (LDA) is used.
- For sentiment analysis and scoring, classification algorithms including machine learning and deep learning techniques are used.
- Machine learning includes Naïve Bayes, SVC and Deep learning includes LSTM/RNN, CNN.
- The sentiment scorer model is trained on overall article corpus ignoring the time tag. The more training data, the better the model performance.
- The trained sentiment scorer model in production scores time sensitive articles/text by hour.

Output of the natural language processing (NLP) steps

- Two dictionaries of entities/keywords are created for every hour. One has keyword : sentiment score list and the other has keyword : weight
- Examples:

  {K1:[s1,s2,s3],K2:[s1,s2,s3],K3:…….}

  {K1:w1,K2:w2,K3:w3……}

  K1, K2, K3 stand for entities/keywords

  s1, s2, s3 stand for sentiment scores

  w2, w2, w3 stand for entity weights

Time Series Analysis

Data used for the time series analysis are hourly asset price for the window size (270 data points) and the sentiment scores calculated from the dictionaries above.

Sentiment scores are calculated in two ways:

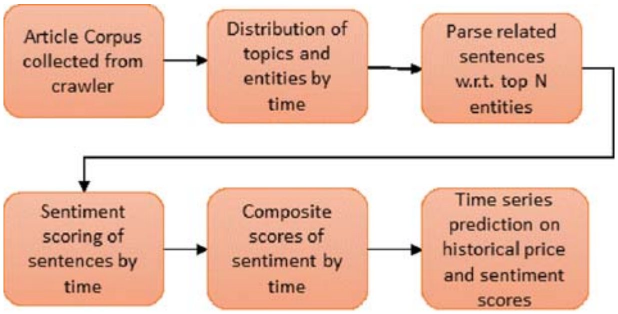
- Score by entity : K1_score multiplied by K1_weight
- Composite score : Weighted average score of all entity scores

Historical chart analysis of price vs score(s) is done to better understand the trends and relationship between asset price and sentiment scores.

Autoregressive Integrated Moving Average with Exogenous Variable (ARIMAX) is used for time series prediction if the price data is found to be auto-correlated. The exogenous variable is the sentiment score(s) from the news corpus.

Deep Learning Seq2Seq model like LSTM/RNN is used to predict asset price using historical asset prices and sentiment scores.

Workflow


Article Corpus collected from crawler
Distribution of topics and entities by time
Parse related sentences w.r.t. top N entities
Sentiment scoring of sentences by time
Composite scores of sentiment by time
Time series prediction on historical price and sentiment scores

Big data and parallel processing

Since the news corpus is a large dataset, parallel computing is required to analyze the data. Spark MLlib offers LDA algorithm in a distributed environment. Deep Learning/Machine Learning algorithms can be run in a distributed environment using Tensorflow / SpaCy environment.

III. Conclusion

Since this approach identifies time sensitive key issues from large corpus of daily news, it automatically filters out noise. Subsequently, the conversion of unstructured news data into a more structured sentiment scores creates a useful feature that is used for downstream time series prediction.

As the corpus changes with every hour, the distribution of topics/entities from topic models changes and therefore the outputs of the NLP process becomes dynamic and time sensitive.

Several algorithms are used in the project. In order to get the most optimal performance, grid-search and hyper-parameter tuning are performed for each algorithm and the overall NLP pipeline.

Next step would be to convert the research project proposal into a real implementation and review the results of the models.