CS446 Class Project Proposal - Stockit

dmcquil2@illinois.edu, mcconne7@illinois.edu

April 23, 2015

Abstract

We would like to given a set of articles create a similarity matrix between each document and then create a simple kNN clustering algorithm to predict the trend of a given stock.

1 Introduction

We would like to create an algorithm that would be able to forecast stock price changes from a given news article content. This would help to test ways that one could predict a stock trend and could act as a hint as to whether a stock is about to increase or decrease. We will be using a text search algorithm to rank the tickers on document relevance. Then we will create a kNN implementation to cluster the documents according to a positive or negative increase in the mentioned stock.

Our null hypothesis is that we would like to prove that calculating stock trends using news articles is more accurate than observing the stocks history.

2 Background

There have been other attempts to predict stock market prices using machine learning. For example, Textual Analysis of Stock Market Prediction Using Financial News Articles approaches this challenge from a very similar direction. Their research has shown that text information can provide a 1% to 2% increase in accuracy of predicting the change in a stock price than a purely regressive model [3]

3 Task and Data

Now describe the task and data in more detail.

3.1 The Task

In order to predict stock trends we will be using a k nearest neighbor algorithm that will cluster the documents. This algorithm will first generate a similarity metric for all documents. The similarity between documents will be calculated using the following scoring function as described in the Lucene Similarity implementation [2]

$$sim(d_i, d_j) = coord(d_i, d_j) \cdot queryNorm(d_i) \cdot \Sigma_{t \in d_i} \left[tf(t \in d_j) \cdot idf(t)^2 \cdot norm(t, d_j) \right]$$

where

$tf(t \in d_j)$: this is the term frequency of a term in document d_j where t is a term present in document d_i

$idf(t)$: this is the inverse document frequency for a given term in a document as defined as

$$idf(t) = 1 + \log\left[\frac{numDocs}{docFreq(t)+1}\right]$$

$coord(q,d_j)$: this is a score based on how many of the query terms are found in a given document d_j

$$coord(q,d_j) = \frac{\text{\# of query terms found in } d_j}{\text{\# of query terms}}$$

3.2 The Data

For the purpose of this project we have set up a solr instance to use as our datasource and text computing platform to handle the large amounts of data. We split our data into three datasets. This solr instance is currently deployed to solr.deepdishdev.com/solr.

To find relevant articles we were required to search the articles for a given ticker name and retrieve results whether the ticker was present or the name of the company. We used a synonyms file generated from the tickers that were present in the stocks dataset. We also used an implementation of the Porter Stemmer algorithm [1] to improve search results.

Using our three datasets we will create a fourth composite dataset that will be a document set that contains an article, the most relevant ticker, and the history for the day following the release of the article. We use a set of synonyms generated from our stock stockers from For a given query and a document we calculate the score using cosine similarity

$$score(q,d) = cosine\_similarity(q,d) = \frac{V(q) \cdot V(d)}{|V(q)||V(d)|}$$

where V is a mapping function that translates both the query and document to a search vector

To find the most relevant ticker we will find the ticker for each article that has the highest correlated score. This will give us a new composite dataset which we will use as our training and test data for our algorithm.

- **Stock history** - This dataset will represent the historical stock data going back to 2001 for all stocks available on Yahoo's financial data platform (24,000 stocks). This historical information was retrieved via the Yahoo data platform.
- **Stocks** - This dataset is a listing of all stocks currently available on the Yahoo financial data platform and also the categories and indexes they are associated with. This index also contains the current aliases for each stock as well.
- **News Articles** - The dataset is a collection of news articles that have been retrieved from various website via web page scraping. Currently this dataset is made up of content from:
  - newsmax.com
- **Stock Articles** - The dataset is a joined collection of the above three data sets. This will be used as our training and testing data for our model. Each document in the dataset, D will consist of:

title : The title of an article

content : The contents of a scraped article

date : The date that an article was published

symbol : The symbol of the stock with the highest correlation to the article

stockHistoryDate : The day after the article was released. This will be the date for which we will retrieve stock history.

close : The closing price of a given stock for the specified date

open : The opening price of a given stock for the specified date

4 The Models

4.1 Baseline Models

In order to know how difficult the task is and how well we are doing, we need to know how well a suitable baseline model would perform. Define a baseline model for your task. This may not necessarily be a learned model.

4.2 Existing Models

If people have worked on this task before, summarize (and cite) some of the existing models

4.3 Proposed Model(s)

Your models and your procedure for learning them go here. Describe both in detail, even if the learning procedure is standard.

5 Experiments

5.1 Experimental Hypotheses

Summarize the hypotheses (research questions) your experiments are designed to test (address). (Note that some of these hypotheses may emerge as you keep working on a problem; you will not necessarily have come up with all the questions you wish to address before you have started building a models for the specific task.

5.2 Experimental setup

Define test/training/dev data splits, describe how you tuned performance. Describe and your evaluation metric, and define it mathematically. List the models you will evaluate. Cite any existing tools or software you use to perform your experiments; describe what you implemented yourself. Describe how you obtained the features used by each of the models.

5.3 Experimental results

Now give the actual experimental results (use figures/tables/graphs as appropriate), and discuss whether they verify or falsify your hypotheses. How important are the various features your models use (consider ablation studies). How robust are your results? (Look at learning curves, or the variance when you perform cross-validation). Can you perform an error analysis?

6 Conclusion

Summarize your findings, and discuss their implications, e.g. for future work, or for related tasks. Discuss also the shortcomings of your proposed approach. .

Bibliography

References

[1] S.E. Robertson C.J. van Rijsbergen and M.F. Porter. *New models in probabilistic information retrieval*. British Library, London, 1980.

[2] The Apache Apache Software Foundation. Lucene similarity. `https://lucene.apache.org/core/2_9_4/api/core/org/apache/lucene/search/Similarity.html`. Accessed: 2015-04-16.

[3] Robert P. Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using financial news articles. *Artificial Intelligence Lab, Department of Management Information Systems*, 2009.