Kaggle Competition: 2sigma Using News to Predict Stock Movements

Barthold Albrecht, Yanzhuo Wang, Xiaofang Zhu

Stanford
Computer Science

Abstract

The 2sigma competition at Kaggle aims at advancing our understanding of how news analytics might influence stock prices.

A large set of daily market and news data from is provided for US-listed financial instruments. This data shall be used to predict future stock market returns.

By applying different supervised learning algorithms we try to continuously increase the quality of our predictions and our position on the leaderboard of the competition.

Our best performing algorithm is FCNN which makes us #900 out of more than 2200 as of now.

Stanford University

Motivation

In our project we want to apply our newly obtained knowledge from the course to a real world problem. **Predicting stock returns** is a challenging field of broad interest for machine learning. Thus, the currently running 2sigma competition at Kaggle is a good opportunity to put various algorithms at use and improve our understanding of their strengths and limits.

Setting of the problem

The competition comprises two stages: In the first stage the predictions are tested against historical data. It ends early next year at which time the final submissions must be handed in. Those will then be evaluated against future data for about six months to identify the best performing submission which will be disclosed 7/15/2019.

The objective function is set as follows: for each day *t* within the evaluation period the value x_t is calculated as

$$x_t = \sum_i \hat{y}_{ti} r_{ti} u_{ti}$$

And

$$score = \frac{\overline{x}_t}{\sigma(x_t)}$$

where the score (mean divided by standard deviation of daily predictions) is used to determine the competition winner.

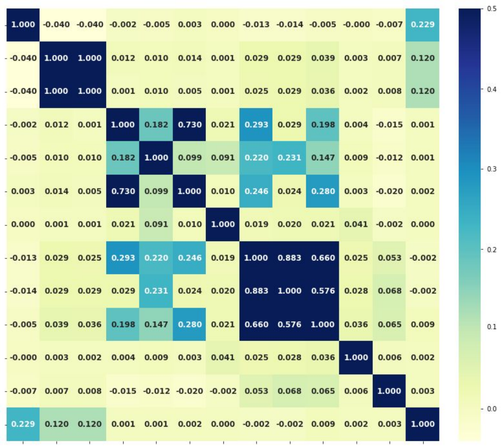
Data

Two datasets for the period 2007 - 2016 are provided, one for market data and one for news data, for more than 3700 assets.

The **market data** comprises more than 4 million samples with 15 features which are mainly trading data (like volume, open/close, raw/market-adjusted returns etc.), and identifiers. The number of all tradable assets at a given day ranges from roughly 1300 to 1800. The market-adjusted ten-days future return serves as the ground truth value for the prediction task.

The **news data** comprises more than 9 million samples with 35 features (like urgency, audiences, subjects, sentiment class, word count or novelty). Only one feature called „headline“ actually contains text.

A Correlation Heatmap on the market data after removing the outliers

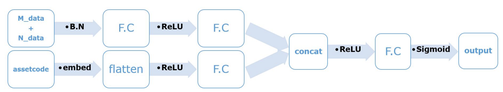


Methods

Data preprocessing
After joining the market and news data on asset codes we normalize numerical data an encode the asset codes. The ground truth data is transformed into labels for classification.

Models
We run **logistic regression** for baselining with partial features selections. Given the large size of overall data, small regularization is added.

We implement a fully connected **neural network** (FCNN) with Keras. The NN has 5 hidden layers (4 with relu activation plus the last one with sigmoid). An embedding layer learns the encodings of the asset codes which are then concatenated to the numerical data. We use binary-crossentropy as loss function and the Adam optimizer. The model has ~50k trainable parameters



We implemented **Gradient boosting regression tree** in LightGBM library using binary loss with bagging and feature sampling rate of 0.9. We regularize the number of leaves to be 60 and depth to be unlimited.

$$F = \sum_k^K F_k(x)$$

$$r_k = -[\frac{\partial L(y, F_{k-1}(x))}{\partial F_{k-1}(x)}]$$

$$L = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(F_k)$$

Results

	LR	FCNN	LGBM
V-Acc	0.485	**0.557**	0.538
V-score	0.247	**0.781**	0.731
Score	0.259	**0.645**	0.644

For FCNN we also ran the model with market data only to assess the impact of news data. The results are 0.553 for accuracy and 0.617 for score .

Discussion

FCNN and LGBM perform the best. The poor result of the logistic regression is expected because the algorithm assumes linear relationships. Adding news data does not have noticeable effect on the performance. This can be explained by that the news data is more subjective and there are many noises and non-related features. The validation accuracy is similar for all three algorithms. The competition scores, however, are quite different. One explanation could be that the score calculation considers not only the binary prediction, but also market return and standard deviation of the prediction.

Future

- More feature engineering to filter out bad data and outliers
- Combine FCNN and GBM to create an ensemble model
- Add Natural Language Processing algorithms for data consisting of words
- Fine tuning FCNN architectures as well as LightGBM hyperparameters