Machine Learning Engineer Nanodegree Capstone Proposal

Gang Yang 1/1/2017

Domain Background

"How can we use the world's tools and intelligence to forecast economic outcomes that can never be entirely predictable?"[1] Kaggle's Two Sigma Financial Modelling competition challenge the data science community to explore what they can do in the financial market with the new technology in data science and machine learning. I decided to take up this financial modelling challege.

As financial market is affected by contingent political and economical factors, financial modelling has the complication of being an out-of-sample forecasting problem. In contrast, other problems such as designing advertisement campaigns or recommending systems tend to be more in-sample as personal habits was much more stable and easy to learn compared with the pattern in financial market. Financial market also demonstrate time-dependent features, such as time-varying volatility clustering. Though financial outcomes can never be entirely predictable, we do not need to predict everything correctly to make a profit and newly discovered pattern may lead into profitable strategy. Thus I am curious what can be learned through applying machine learning techniques to financial data.

Problem Statement and data sets

In the two sigma financial modelling competition, a dataset containing time-varying anonymized features for a collections of financial instruments was provided. [2] The entire data contains y values and 110 features including the unique id for each financial instrument, timestamp, 5 derived features, 63 fundamental features and 40 technical features. These features are all numerical features and the y values are also numerical feature. The y value is pretty much normally distributed with a little exception on the boundary.[3] The data has 1710756 rows, 1424 different instruments and 1813 different timestamps.[4] This challenge is a supervised learning problem, as the features and concerned y values are provides are the first 1813 timestamps and we are expected to predict the y value for the next timestamps. (Complication of the public leaderboard will be discussed in the metrics section.) Obviously, this problem is quantifiable, measurable and replicable.

These features are anonymized as to prevent people from using other resources to make prediction to win over the competition. This add complication to the challenge as it's hard for us to build intuitive model or select important features through intuition. Thus a rough interpretation of the dataset would also be part of the project.

[1]https://www.kaggle.com/c/two-sigma-financial-modeling (https://www.kaggle.com/c/two-sigma-financial-modeling)

[2]https://www.kaggle.com/c/two-sigma-financial-modeling/data (https://www.kaggle.com/c/two-sigma-financial-modeling/data)

[3]https://www.kaggle.com/anokas/two-sigma-financial-modeling/two-sigma-time-travel-eda (https://www.kaggle.com/anokas/two-sigma-financial-modeling/two-sigma-time-travel-eda)

[4]https://www.kaggle.com/jeffmoser/two-sigma-financial-modeling/kagglegym-api-overview/notebook (https://www.kaggle.com/jeffmoser/two-sigma-financial-modeling/kagglegym-api-overview/notebook)

The data is available publicly through [2] and it is downloadable as a hdf5 file. After some exploration, there is also certain missed data in the dataset, thus handling the missed data is also part of the project.

Evaluation Metrics

The evaluation metrics is chosen to be the R value between the predicted value and the actual value of the target variable. The R value is the square root of the R square value multiplied by the sign with R square, which is also called coefficient of determination. As a new code competition, Kaggle add a Kaggle Gym API. During the run process in the API, only half of the data mentioned above is provided, and we need to make prediction for the next timestamp based on these available data and the corresponding features for that timestamp. After the prediction, a reward score(R value) instead of the true target variable is provided. Kaggle use this to prevent competitors from probing the test set and leave space for implementing a reinforcement learning strategy. The run is iterated until we reach the final timestamp available. In the submission mode, the process is the same as above except we are provided the full set of data and our model is evaluated against the hold-out test data, which is currently unavailable. 37% of the test data is evaluated to give a public leaderboard score and the remaining 63% is hidden until the end of the competition.

Also the competition put a limit on the running time of the kernel, thus running complicated model on the fly is not favorable.

Solution Statement and Benchmark model

The solution of this challenge would be an algorithm taking the existing training data and updated data from test data through each time stamp and predict the target value. The goodness of the algorithm is evaluated against the R value of the predicted data and actual data.

As established in the notebook [5], this competitor employ a linear regression model (y value ~beta * feature + interception) using only one feature from the all the feature set as iterate through the test set; the hyper-parameter is not changed through the iteration and the newly generated data is only used to predict the target value; the missing data is replaced by the mean value of the corresponding feature after cleaning out the outliers. The selected feature is Technical_20 after some experiments. This algorithm gives a public leaderboard score as 0.0091176.

[5]https://www.kaggle.com/achalshah/two-sigma-financial-modeling/linear-regression-lb-0-0091176/code (https://www.kaggle.com/achalshah/two-sigma-financial-modeling/linear-regression-lb-0-0091176/code)

Project Design

1.Data proprocessing

As in every machine learning problem, we need to do exploratory data analysis for the dataset including basic statistics of single variable, correlation between two variables, time evolution of the features. We also need to study the correlation between each feature and target variable as a hint for feature selection and feature engineering.

We need to explore whether there are outliers in the dataset and find reasonable ways to filll in missing values.

2.Benchmark model reproducing and trial of standardized method

As this competition put on a run time limit and also we need to make iterative prediction on the data, we should push the boundary of simple models and explore what these models can accomplish. Benchmark models may include (but not limit to) linear regression, regularized linear regression, regression tree methods such as extreme gradient boosting or random forest(probably too slow in this case). We also need to do feature selection for these models to improve the performance of the competition.

As time plays an important role in this problem, one would naturally tend to employ time-series approach. However, as the test data is hold out, the distribution of the timestamp in the testdata is unknown and also the scale of the timestamp is not given, which makes it hard to correctly use a time series model. We do

need to take trials to see whether time-series method will improve the result.

3.Make an ensemble of existing simple models

After building these simple models, we could make an ensemble of the simple models as this often will improve the result.

4.Other things to consider

Feature engineering: we could consider adding interaction feature or congregated feature.

Cross validation strategy: The time series problem complicate the usual cross validation routine. We need to find out new cross validation practice.

Reinforcement learning: The fact that kaggle provides the reward seems to suggest we should employ reinforcement learning, but it may subject to the time limitation of the competition.