Random forest

Theoretical foundation

Unsupervised pattern recognition

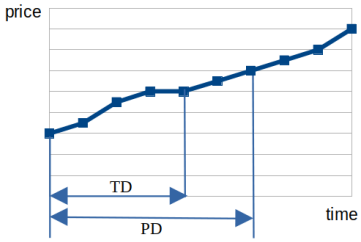
(废话)

In today's big data age, "Let the data speak for themselves" has been proposed and drawn attention. And applying machine learning on stock price prediction has been one popular research topic in recent years. For instance, neutral networks were utilized to predict the stock price based on the past stock prices. Other methods such as ensemble learning, or combinations of different models are also used on this issue. However, above studies selected only a few training data and labels from the past stock data to train and test models, which cannot adequately reflect tendency of the whole stocks, therefore limited the availability and capabilities in real applications.

The stock market releases a huge amount of information every day. Manually labeling the stock data is time consuming and inefficient. In our project, an unsupervised pattern recognition algorithm[1] is used to automatically label the data of stocks.

Training samples generation

In the random forest model, stock prices of the early trade days are used to predict whether the predefined shape will appear in the whole fixed duration. Here the fixed duration is defined as pattern duration denoted as PD, the duration of stock prices used for constructing train samples is defined as training duration denoted as TD. The correlation between PD and TD is illustrated as the figure below. After PD and TD are set, we slide a window of length PD on the original dataframe which contains time series of stock prices. Then we get a large number of frames containing prices information in a fixed length duration PD. Next we can label each frame by applying pattern recognition algorithm. Finally training samples are extracted from the first few rows of these frames where the number of rows is TD. And the label of each training sample is the label of corresponding frame in the pattern duration.



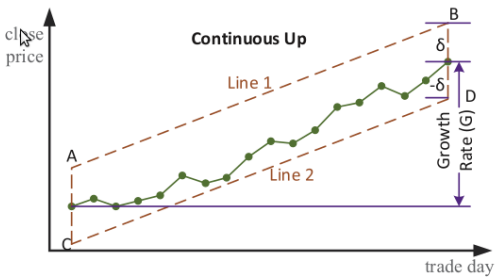
Introduction of classes

Since our project aims to predict the tendency of stock prices, six classes are defined in our model based on the shapes of close prices of a stock, i.e., the price will (1) rise up steadily (Continuous Up), (2) rise up more and more rapidly (Slides Up), (3) drop down steadily (Continuous Down), (4) drop down more and more rapidly (Slides Down), (5) stay approximately the same (Flat), (6) fluctuate without predictable tendency (Unknown).

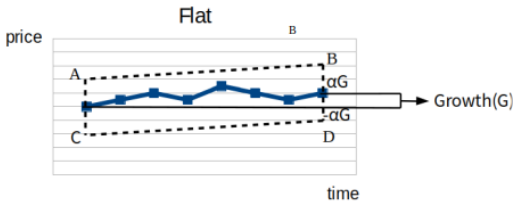
Unsupervised labeling algorithm

To classify different shapes of close prices automatically, some labeling algorithms[1] were applied on the close prices in the pattern duration to label each frame of stock prices. In this report we will not introduce algorithms of identifying *continuous down* and *slides down* labels since the *continuous up* and *slides up* labels are mirrors of the previous two.

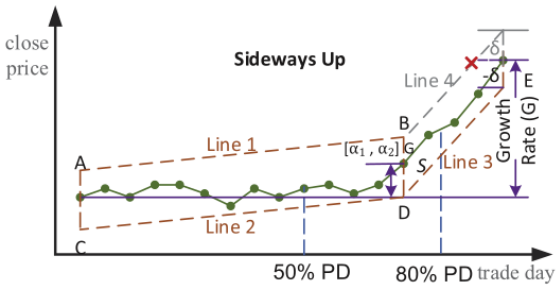
Algorithm for recognizing *continuous up* is relatively simple. First, we calculate the growth of stock price in the pattern duration by comparing the close price on the first day and last day. Then we can get four points A, B, C, D shown in the figure below, where AC, BD denote the price vibrating between the range [-δG, δG] (e.g., δ=0.5) of first day's price and last day's price respectively. Finally if a proportion larger than η (e.g., η=0.95) of points of the period are located in the area ABCD, the duration of stock data will be labeled as continuous up.



Flat pattern recognition is similar with the algorithm above, if the growth is less than a threshold and the proportion of points in the pattern period located in the area ABCD is larger than δ (e.g., δ=0.95), then it can be classified as *flat* pattern. The figure below illustrate flat pattern algorithm.



Investors may care more about slides up pattern because it means that the stock price is possible to rise more rapidly than ever, which indicates good opportunity for warehousing. *Slides up* pattern recognition is more complicated. First calculate growth G of the stock price as before. Then suppose there exists a point located in the time range [50%PD, 80%PD] denoted as *k*. The points between the first point and k can form the continuous up pattern where AC and BD denote the price vibrating between the range [α_2G, α_1G] (e.g., α_1=0.05, α_2=0.4) of first day's price and last day's price respectively and points between *k* and the last point always lie above the line DE where E is the price of last point subtracted by δG (e.g., δ=0.5). If there exists such k point the frame of stock prices will be labeled as *slides up*.



Undersampling

Random sample the training samples to make number of samples of each class equal.

Experiment setup

We use a database containing stock prices with 15 minutes time interval between neighbor records as well as the technical indices of stock market. The time span of our database is within the range from February 22, 2015 to September 3, 2018.

Evaluate

we first measure accuracy of test samples with different combinations of PD-TD. We generated training samples and labels by applying slide window method on the dataframe, which only contain open, close, high and low prices of the duration of PD*15 minutes.then use 70% of the training samples and labels to train the random forest model. And the remaining 30% are used as test data. Prediction accuracy is calculated by validating the test data. The results are listed in the following table.

PD-TD	10-6	15-10	20-13	30-20
Accuracy	0.520	0.590	0.629	0.653
PD-TD	40-26	50-33	60-40	avg.
Accuracy	0.693	0.714	0.738	0.648

From the result it can be found that our model is more suitable for long period prediction because of higher accuracy with longer PD-TD combinations, because longer duration can provide more information to the model.

Feature selection

Initially only four features of the stock market are used to train the model, which are the close price, open price, high price and low price. To achieve better performance of the model, I extend the features with technical indices to train and evaluate the model. Since there are totally over 30 features in our dataframe, it is time-consuming to evaluate every combination of features. We use *Forward Sequential Search method*, which evaluate the model by

adding one feature into the original feature set in one iteration and cannot go back. It may not find the optimal feature combination but has high search speed. The table below shows part of experiment result with different combinations of features when the PD-TD is fixed as 30-10.

features	'close', 'high', 'low', 'open'	'close', 'high', 'low', 'open', 'MACD'	'close', 'high', 'low', 'open', 'BIAS10'	'close', 'high', 'low', 'open', 'BIAS10', 'AMA'
accuracy	0.5941469689660717	0.5391542048560862	0.6558932304228976	0.6771185077986814
features	'close', 'high', 'low', 'open', 'BIAS10', 'AMA', 'denoised_data'	'BIAS10', 'AMA', 'close', 'high', 'low', 'open', 'denoised_data', 'AR'	'denoised_data', 'BIAS10', 'AMA', 'AR', 'close', 'high', 'low', 'open', 'WAWVAD'	'denoised_data', 'BIAS10', 'AMA', 'AR', 'close', 'high', 'low', 'open', 'WAWVAD', 'up'
accuracy	0.6982851018220794	0.7001607717041801	0.6915862808145766	0.6993569131832797
features	'denoised_data', 'BIAS10', 'AMA', 'AR', 'WAWVAD', 'up', 'close', 'high', 'low', 'open', 'sar'	'denoised_data', 'BIAS10', 'AMA', 'AR', 'WAWVAD', 'up', 'close', 'high', 'low', 'open', 'sar', 'dn'	'denoised_data', 'BIAS10', 'AMA', 'AR', 'WAWVAD', 'up', 'close', 'high', 'low', 'open', 'sar', 'dn', 'ADR10'	'denoised_data', 'BIAS10', 'AMA', 'AR', 'WAWVAD', 'up', 'close', 'high', 'low', 'open', 'sar', 'dn', 'ADR10', 'TRMA'
accuracy	0.702572347266881	0.7017684887459807	0.6966773847802786	0.6915862808145766
features	'denoised_data', 'BIAS10', 'AMA', 'AR', 'WAWVAD', 'up', 'close', 'high', 'low', 'open', 'sar', 'dn', 'ADR10', 'TRMA', 'RSI10'	'denoised_data', 'BIAS10', 'AMA', 'AR', 'WAWVAD', 'up', 'close', 'high', 'low', 'open', 'sar', 'dn', 'ADR10', 'TRMA', 'RSI10', 'rise10'	'denoised_data', 'BIAS10', 'AMA', 'AR', 'WAWVAD', 'up', 'close', 'high', 'low', 'open', 'sar', 'dn', 'ADR10', 'TRMA', 'RSI10', 'rise10', 'OBV'	'denoised_data', 'BIAS10', 'AMA', 'AR', 'WAWVAD', 'up', 'close', 'high', 'low', 'open', 'sar', 'dn', 'ADR10', 'TRMA', 'RSI10', 'rise10', 'OBV', 'roc'
accuracy	0.7023043944265809	0.687566988210075	0.6878349410503751	0.6752411575562701

From the result we find that the recommended feature combination is ['denoised_data', 'BIAS10', 'AMA', 'AR', 'WAWVAD', 'up', 'close', 'high', 'low', 'open', 'sar', 'dn', 'ADR10', 'TRMA', 'RSI10']. In fact the prediction accuracy of the combinations are so close that there may not exists a global best feature combination which can be applied in all cases based on different data sets. We can only find the relatively better feature combination for our model.

Reference

[1] Zhang, J., Cui, S., Xu, Y., Li, Q., & Li, T. (2018). A novel data-driven stock price trend prediction system. *Expert Systems with Applications*, *97*, 60-69.