Final Project Report: Stock Trading Analysis

MSBD5013

Statistical Prediction

12/17/2017

Yilan Yan	**20475213**
Lingxiao Zhang	**20475043**
Guannan Lu	**20454477**
Chengming Wang	**20450392**

TABLE OF CONTENTS

1. Abstract

In this project, we are assigned to come up with stock trading strategies based on the data from the real market. We need to write a strategy function our own to perform transactions from the data using Python. We have used various methods including some indicators and machine learning models to try to maximize our returns. The feedbacks of our results turn out that the indicators we've used have higher performance than the models.

2. Introduction

The purpose of this project is to design and implement trading strategies on 13 items from China futures market, including ten commodity futures and three stock index futures with minute-level future data. We need to write strategy functions that will update the position matrix on a specific time interval we specified based on the data passed in every minute. The position matrix is used for determining the number of lots to buy or sell so the size of the it is 1x13 and a positive value means to buy and sell otherwise. For trading indicators, we've three indicators, which are MACD, RSI and KD. For statistical models, we've used two models which are SVM and LSTM. Our strategies need to be tested weekly under a standard testing program, which will generate feedbacks of our trading strategies, such as cumulative returns, sharp ratio and maximum dropdown.

3. Discussion of the Project

3.1. Design Procedure

At first, we created some helper functions that is used for storing and generating input features/data for indicators and models such as average price, close price, difference ratio of a certain index of the past given minutes.

3.1.1. Indicators

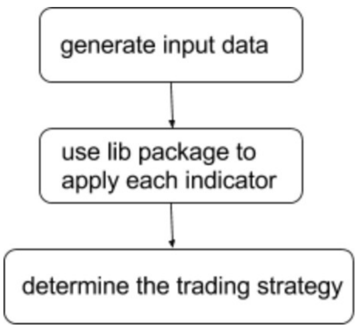

Figure1. The design procedure of indicators

As the figure shown above, the design procedure of indicators can be separated into following 3 steps:

1. Generate the input data required for each particular indicator.
2. Use the TA-Lib package to apply each indicator.
3. Use the feedbacks/outputs of each indicator to determine the trading strategies.

3.1.2. Models

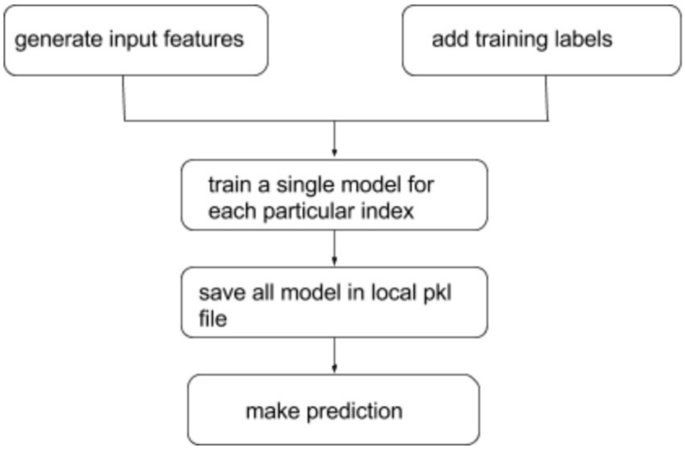

Figure2. the design procedure of model

As the figure shown above, the design procedure of models can be separated into following x steps:

1. Generate the input features we've selected for the model
2. Add the training label according to the rules we've made
3. Train a single model for each particular index and save all the models to local pkl files
4. Load the model for each particular index and use the prediction from the model to determine the trading strategies

3.2. System description

3.2.1. Helper Functions:

generate_data_helper() -- retrives the open, high, low, close price of a given index.

get_past_data() -- retrives the open, high, low, close price of a givenindex for the past few days.

compute_average() -- compute average of an array among high, low, open, close prices.

3.2.2. Main Functions:

handler_bar() -- main function of algorithms of the trading strategy.

svmtest() – method for training the SVM model.

LSTMtest() -- method for training the LSTM model.

3.2.3. Test Functions:

backTest() -- evaluation method for testing the performance of the strategy.

backTest_2() -- evaluation method for measuring the cumulative returns of all indexes on a particular dataset.

4. Description of algorithms, functions, or procedures:

We have set a certain time interval to make decisions of transactions or not for both indicators and models. we have also set the capitals for each transactions by using a certain rate of the initial cash for each transaction.

4.1. Indicators

As mentioned, we have used three investment indicators in total. For each indicator, we've set and adjusted their corresponding parameters to try to achieve better performance possible on each weekly dataset.

MACD

For MACD, we gave the following values: fast_span = 12, slow_span = 26 and signal_span = 9. These values are suggested from the articles we've found online and indeed works well for several indexes.

RSI

For RSI, we gave the following values: rsi_lower = 40, rsi_higher = 60, rsi_period = 21. The suggested values of the lower bound and higher bound are actually 30 and 70. However, we've tested and checked the actual values on weekly datasets, and 40 and 60 fits better.

KD

For KD, we gave the following value: kd_low = 10, kd_high = 90, fastk_period =5, slowk_period = 3, slowd_period = 3. These values are also suggested from the information we've searched online and also works well for certain indexes on weekly datasets.

We have tested the performance of each particular indicator for each weekly dataset and make comparisons of among all for a given index.
We have implemented majority of votes on these indicators above, which use the counter to count the number of votes for 1. For two actions buy and sell, we have used two counters for each. If the count is greater than 2 for the buy counter of a particular, a buy transaction will be performed on the position matrix. Similarly, if the counter of the sell is greater than 2, we will sell that particular indexes of a certain amount of lots.

4.2. Models

SVM

As we know, SVM has less dependence on experience, which can obtain the global optimal solution and has good generalization performance. So it can also effectively overcome the local optima problems that can not be avoided by neural network and other methods. In addition, SVM is a learning machine specially designed for limited sample. It adopts the principle of minimizing structural risk to control the experience risk and the complexity of learning machine, effectively avoids the phenomenon of over-learning, and achieves better than traditional learning methods.

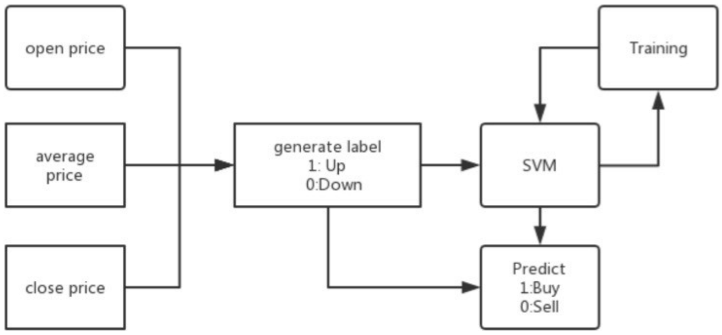

Figure3. Trading strategies using SVM

From the changes of price in one minute, we can predict whether the price will increase or decrease by using SVM model. Open, average and close price of each minute, which are the input of SVM model, shows the trends of price in this minute. The label will be regards as 1 if the price of next minute is higher and 0 if it is lower. SVM model will be trained by using the several month of historical data and tested by the data of latest weeks. Moreover, we choose several stocks or products with the best performance as the targets we want to trade.

LSTM

LSTM is short for Long Short Term Memory. It is a modified version of Recurrent Neural Network which can solve the problem of vanishing gradients. It takes as input a series of values and outputs a series of values.

In this case， we use LSTM to do multi-step time series forecasting. We train the model on the close price of asset A.DCE. First we smooth the data using 15-minutes average. The input is one time step and the output is three time steps forward. The network structure is one layer with 10 neurons. We train for 10 epochs and set batch size to be 100.

Every 15 minutes we make a decision whether buy or sell based on the value change between the prediction price of the third step forward and the current price. Also to make the model more stable, we also use KD indicator to help make the final decision.

5. Testing

5.1. Test plan

The test plan of our trading strategies is to use the control variables method, which means we set a given time interval of trading such as 1-hour for a particular indicator or 15-minutes for LSTM model, and then check and evaluate the cumulative returns of each index on each weekly dataset. Here we only take the most recent seven datasets for testing. Finally, compare results among all indexes.

In order to generates the returns values of all indexes on a given weekly dataset, we've create the test method ourselves and use the same way of measuring the returns from the backTest() file to ensure the consistency.

5.2. Discussion and Analysis of Results

5.2.1. Indicators

0	1	2	3	4	5	6	7	8	9	10	11	12	macd
9438065	9839627.5	9811400	9953675	10003570	9935950	9972970	8482500	9364555	10061402.5	9656575	9991930	10821215	
9643605	10032478.8	10314175	10580437.5	9836770	9719950	10117525	9044800	9733570	9297042.5	10463575	10246020	9967385	
9779577.5	10157076.3	9988800	10050500	9667240	9773650	9933640	9400725	9108265	9790245	9994587.5	9781615	10155685	
10250920	10027026.3	8009337.5	9806662.5	10031980	10184275	10740985	9244100	9968320	9992115	9680960	9515960	8957245	
9292627.5	10004605	10041462.5	9764000	9606670	10010950	10641550	10930337.5	10506295	10277125	10200740	10457950	10114230	
10519110	10471836.3	10092037.5	8966800	9701440	10529650	10551040	9728187.5	8584217.5	10055425	9486970	10212965	9351080	
9326712.5	11062738.8	11067837.5	10120337.5	9919960	10081525	10210420	7441862.5	9571795	9833010	10219725	11672890	9848750	

Figure 4. Cumulative returns of each index on the most 7 recent datasets using MACD

From this figure, we save the updated 7 datasets to supervise the returns of each commodity index in each indicator and mark red to display the profitable futures (above 10,000,000). We find that future 1 and 6 are stable and profitable for the given transaction interval. In the usual performance, we choose indexes that has positive returns every week and change the strategy based on the performance. Finally, we choose the index 1 using MACD indicator to perform in playoff and assign highest weight to it.

0	1	2	3	4	5	6	7	8	9	10	11	12	rsi
10694560	9799198.75	10314975	10141562.5	9755960	10354720	9941005	11119800	10703522.5	9895957.5	11516577.5	10347640	10295460	
8925145	10013773.8	10107687.5	9702825	10123190	9327520	9371020	10633187.5	10595410	9256877.5	8825022.5	9940405	9827880	
10073515	10142365	10054087.5	10115487.5	10446200	10453510	10208890	10627537.5	10667882.5	10178975	11077480	9756180	10835730	
9418837.5	10000000	9261050	10284150	9887900	9574135	9807280	10859300	10530205	10665685	10181407.5	11257035	8807180	
9746790	10053700	10071137.5	10408950	9775990	10014400	9876430	10691300	9474842.5	10451312.5	9419695	9473460	10826740	
9179502.5	9933298.75	10000000	9757025	10095500	10082575	9979525	8076500	8707945	11195145	8769757.5	9779245	10516440	
8039655	8863858.75	9567637.5	9323375	10037800	10127575	9671650	9536775	12180467.5	9163710	10547025	11663705	11184780	

Figure 5. Cumulative returns of each index on the most 7 recent datasets using RSI

For RSI indicator, we seldomly choose the futures because they are not stable enough in the usual season. However, we choose indexes 12 to perform in the playoff because it performed better in the latest datasets. For this reason, we assign the lowest weight for it.

													kd
0	1	2	3	4	5	6	7	8	9	10	11	12	
10000000	10000000	10052237.5	9612625	9984250	9705040	10020700	9927112.5	11420410	8959922.5	8903265	9734230	10000000	
10032095	9785995	10093625	9960400	10170640	10000000	10003795	9440300	10000000	10072392.5	10000000	10531830	9655575	
10711070	10678720	10177712.5	10174075	10329440	10595200	9753490	10000000	9611387.5	9020567.5	8881802.5	10012975	10000000	
10191992.5	10905576.3	10052375	10431900	10139100	10185280	10052140	10875700	11032480	9897200	10256525	9987160	10013580	
10000000	9981452.5	12062900	10008650	9787880	9774100	9807610	10124000	10000000	10341130	10000000	11059555	10791210	
10727247.5	9912463.75	10301200	10068250	9637960	10029250	10000000	10025000	9947485	9625600	10000000	9707635	10000000	
10086725	10157421.3	9851500	9252675	9770400	10080775	10471240	10000000	6603917.5	9892360	9471022.5	9847480	10307120	
11395695	11636312.5	9586000	9740075	10352200	9949120	9836320	6981962.5	7995137.5	9478797.5	10351752.5	9644590	10000000	

Figure 6. Cumulative returns of each index on the most 7 recent datasets using KD

For KD indicator, we find index 0 is always profitable, which means it is the most stable one in all 13 commodities of this indicator. We choose the index 0 to perform in play-off and assign the 1.0 weight to it. In the usual season, we choose the index that has the greatest returns and change strategy every week based on the performance.

	Backtest
Annual return	181.7%
Cumulative returns	50.8%
Annual volatility	69.3%
Sharpe ratio	1.84
Calmar ratio	7.83
Stability	0.58
Max drawdown	−23.2%
Omega ratio	1.41
Sortino ratio	2.74
Skew	−0.39
Kurtosis	2.09
Tail ratio	1.06
Daily value at risk	−8.2%
Alpha	0.00
Beta	1.00

```
Entire data start date: 2017-07-18
Entire data end date: 2017-12-11
Backtest months: 4
```

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	23.19	2017–08–04	2017–09–05	2017–10–17	53
1	20.80	2017–11–14	2017–11–21	2017–11–30	13
2	14.44	2017–12–07	2017–12–08	NaT	NaN
3	6.47	2017–11–03	2017–11–08	2017–11–14	8
4	2.89	2017–11–30	2017–12–01	2017–12–05	4

Figure 7. backtest results of 4-month datasets (Indicators)

This is the result we test on 4-month datasets, the cumulative returns is 50.8 percent and Sharpe ratio is 1.84. Indicators performed better than statistical model.

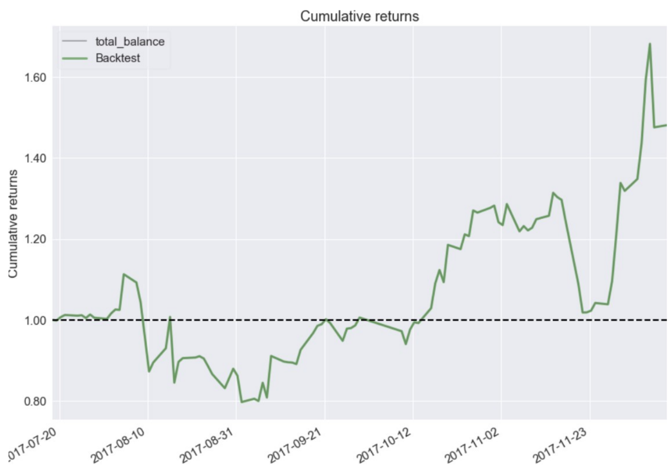

Figure 8. cumulative returns in 4-month (Indicators)

From this cumulative returns line chart, we find a big sharp down happened in august. We use three indicators and the total markets dropped down a lot during August. Then, we changed the strategy on each indicators and earned money from September. Peak time happened in November. The total trend is upward.

5.2.2. Models

From the result of the cumulative returns on the whole 4-month dataset, it turns out that either SVM or LSTM has much higher returns than using indicators in a long term.

SVM

Below is the result on the whole 4-month dataset using SVM on 4 indexes, 3, 8, 10 and 12 with 20, 30, 60 and 60 thousand of capitals for each transaction:

	Backtest
Annual return	283.1%
Cumulative returns	71.3%
Annual volatility	45.9%
Sharpe ratio	3.15
Calmar ratio	21.54
Stability	0.77
Max drawdown	−13.1%
Omega ratio	1.81
Sortino ratio	6.18
Skew	1.23
Kurtosis	4.65
Tail ratio	1.37
Daily value at risk	−5.2%
Alpha	1.38
Beta	−3.90

```
Entire data start date: 2017-07-18
Entire data end date: 2017-12-11
Backtest months: 4
```

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	13.14	2017-09-27	2017-11-22	2017-12-06	51
1	7.73	2017-08-09	2017-08-22	2017-08-24	12
2	5.07	2017-12-07	2017-12-08	NaT	NaN
3	4.38	2017-09-19	2017-09-20	2017-09-25	5
4	1.26	2017-08-03	2017-08-07	2017-08-09	5

Figure 9. backtest results of 4-month datasets (SVM)

Figure 10. cumulative returns in 4-month (SVM)

LSTM

Below is the result on the whole 4-month dataset using LSTM on the index 0 with 30 thousand of capitals for each transaction:

	Backtest
Annual return	709.3%
Cumulative returns	131.2%
Annual volatility	113.5%
Sharpe ratio	2.43
Calmar ratio	20.67
Stability	0.89
Max drawdown	−34.3%
Omega ratio	1.66
Sortino ratio	3.74
Skew	−0.35
Kurtosis	8.46
Tail ratio	1.86
Daily value at risk	−13.2%
Alpha	2.83
Beta	−2.22

```
Entire data start date: 2017-07-18
Entire data end date: 2017-12-11
Backtest months: 4
```

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	34.31	2017-12-07	2017-12-08	NaT	NaN
1	25.46	2017-11-14	2017-11-21	2017-11-30	13
2	23.25	2017-09-19	2017-10-10	2017-10-16	20
3	11.11	2017-11-03	2017-11-06	2017-11-14	8
4	11.03	2017-09-08	2017-09-13	2017-09-19	8

Figure 11. backtest results of 4-month datasets (LSTM)

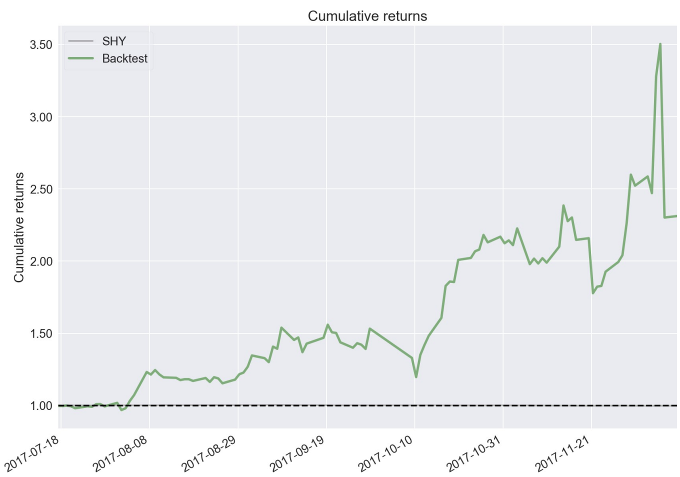

Figure 12. cumulative returns in 4 months (LSTM)

6. Conclusion

In conclusion, we have used both indicators and machine learning models for our trading strategies. For indicators, we've tried and tested the performance of each particular indicator. We've also use majority of votes by using all three indicators. However, the results turn out that sometimes one particular indicator works better than using the majority of votes, especially on weekly datasets. So we create a strategy that use specific indicator for a particular index and we've used all three indicators. The implementation of our final strategy can be think of as a group of people that each has the specific commodity/futures/index they are adept at. Moreover, we assign more transaction capitals for commodities/futures/indexes we've selected that has greater cumulative returns on the 7 most recent weekly datasets. For models, we've used and implemented SVM and LSTM and results shows that they both higher returns than using indicators, however, they are stable on short-time datasets. We conclude that our models are more suitable for long term datasets while indicators have better performance on short terms such as in weeks. Accordingly, we've used the indicators for our strategy on the playoffs since it will be tested on a weekly dataset. In general, we have gained some practical experience on applying both trading indicators and machine learning models on the stock market trading, which can help us to further explore on the real market.