The prediction of return on stock price——Two approaches

Team10
赖亚王, 5130719085
邓凡意, 515120910159
刘益鹏, 515120910200
LINJIANG LI, 713120990034

Prediction of stock prices via coding and data analysis

Classification: SVM using features of stock prices and market conditions from the past

Regression: Using ARMA model

First Step: Get information and handle information

- Get stock information from Yahoo:

```
def get_stock(stock,n):
    column_names=['Date','Open','High','Low','Close','Volume','Adj Close']
    info=pd.read_csv(stock,names=column_names)
    adj_close=info['Adj Close']
    op=info['Open']
```

- Read local csv file directly:

```
info=pd.read_csv('C:/Users/Yawang/Documents/HS300&A50_data.csv')
```

- Calculate the log return of the asset:

```
def get_return(price_data):
    ret=[0]
    for i in range(1,len(price_data)):
        ret.append(math.log(float(price_data[i]))-math.log(float(price_data[i-1])))
    return ret
```

SVM Classification Method: Introduction

support vector machines (SVM):

- supervised learning models
- constructs a hyperplane or set of hyperplanes in a high-or infinite-dimensional space, which can be used for classification, regression, or other tasks
- a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general the larger the margin the lower the generalization error of the classifier

SVM Classification Method: Training

- Step 1: building a data frame of the features of training samples

```
y, x1, x2, x3=get_stock(stock, n)
x4, x5, x6=get_market(market, n)
data={'ret1_stock':x1 ,
      'ret2_stock': x2,
      'ret3_stock':x3,
      'ret1_market': x4,
      'ret2_market': x5,
      'ret3_market': x6}
X=pd.DataFrame(data)
```

Features	
Stock return	*Market return*
Yesterday	Yesterday
The day before yesterday	The day before yesterday
Three days ago	Three days ago

```
for returns in ret:
    if returns>=0:
        label.append(1)
    else :
        label.append(0)
```

Labels	
The stock prices declines	0
Otherwise	1

SVM Classification Method: Training

- Step 2: Split the data into training data & Train the SVM

```
# split the data into training data and testing data
X_train, X_test,y_train, y_test=train_test_split(X,y,test_size=0.25,random_state=0)

#Using Support Vector Machine to predict whether the stock price will go upwards tomorrow or not
m=svm.SVC()
m.fit(X_train, y_train)
y_pred=m.predict(X_test)
```

- Step 3: Check the accuracy of the algorithm

```
print metrics.accuracy_score(y_test, y_pred)
```

SVM Classification Method: Run and Result

- The accuracy rate is only a little more than 50%
- when using 1000 training data, predicting IBM stock and using the return on Dow Jones index as the market return

```
stock='http://real-chart.finance.yahoo.com/table.csv?s=IBM&d=4&e=23&f=2016&g=d&a=0&b=2&c=1962&ignore=.csv'
market='http://real-chart.finance.yahoo.com/table.csv?s=%5EIXIC&d=4&e=23&f=2016&g=d&a=1&b=5&c=1971&ignore=.csv'
n=1000
main(stock,market,n)
```

```
0.52
```

- The prediction is only a little better than flipping a coin
- However, other models (KNN and logistic regression) and other training data (GOOG, APPL and S&P500) also produce no better accuracy.

The difficulty of stock prediction

- The connection between the past price data and the present price is ambiguous.
- Other factors such as the change of policy, investors' attitudes and the condition of the company can also influence the stock price.
- Besides, if many people predict that the stock price will rise tomorrow, they are likely to buy stock today, causing the price to rise immediately instead of rise tomorrow.(reflexivity)

ARMA model: Introduction

- Autoregressive–moving-average model
- The notation ARMA(p, q) refers to the model with p autoregressive terms and q moving-average terms. This model contains the AR(p) and MA(q) models,

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

- ARMA is appropriate when a system is a function of a series of unobserved shocks (the MA part) as well as its own behavior. For example, stock prices may be shocked by fundamental information as well as exhibiting technical trending and mean-reversion effects due to market participants.

ARMA model: Main Purpose

- Ordinary data also have poor result, high frequency data turns out to have nice quality.
- Using prices and returns of every minute to predict.
- Build the model based on the difference between HS300 and A50 to do arbitrage via shorting one asset and long the other.

ARMA model: Data & Training

- Step 1: get the data and handle it

```
# a function to calculate the log return of the asset
def get_return(price_data):
    ret=[0]
    for i in range(1,len(price_data)):
        ret.append(math.log(float(price_data[i]))-math.log(float(price_data[i-1])))
    return ret
```

The data is downloaded from the Wind Database

```
# read the high frequency data of HS300 and A50 between 2015/12/1 and 2015/12/25
info=pd.read_csv('C:/Users/Yawang/Documents/HS300&A50_data.csv')
```

ARMA model: Data & Training

- Step 1: get the data and handle it

```
# get the price data of HS300 and A50
HS300=info['HS300']
A50=info['A50']
# handle the data at similar size
hs300=np.array(HS300)/HS300[0]*1000
a50=np.array(A50)/A50[0]*1000
# get the return of hs300 and a50
r_hs=get_return(hs300)
r_a=get_return(a50)
# calculate the difference between the returns
r_diff=np.array(r_hs)-np.array(r_a)
```

ARMA model: Data & Training

- Basic statistic information for three series – returns on both HS300 and A50 and the difference between returns:

HS300	
nobs	5129
Minimum	-0.007359
Maximum	0.009338
Mean	0.000021
Median	0
Variance	0.000001
Stdev	0.000999
Skewness	0.219524
Kurtosis	5.44164

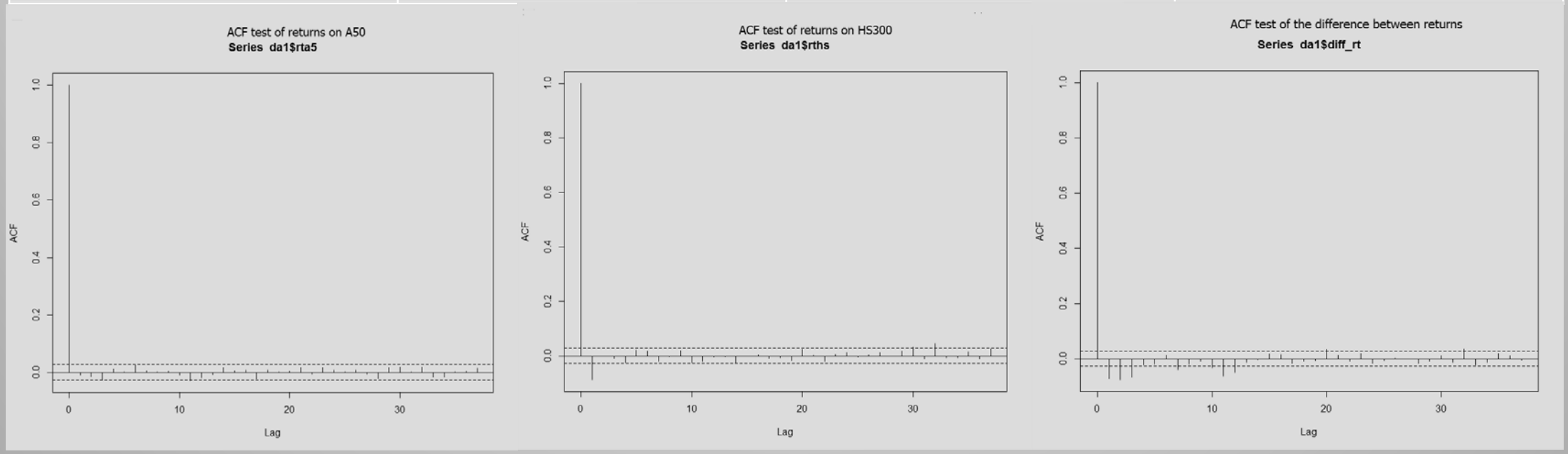
A50	
nobs	5129
Minimum	-0.015929
Maximum	0.009716
Mean	0.000017
Median	0
Variance	0.000001
Stdev	0.000965
Skewness	-0.636954
Kurtosis	23.424518

nobs	5129.000000
NAs	0.000000
Minimum	-0.006915
Maximum	0.009244
1. Quartile	-0.000759
3. Quartile	0.000762
Mean	0.000005
Median	0.000000
Sum	0.023491
SE Mean	0.000018
LCL Mean	-0.000031
UCL Mean	0.000040
Variance	0.000002
Stdev	0.001294
Skewness	0.125291
Kurtosis	2.075146

ARMA model: Data & Training

- Step2: Statistic tests for autocorrelation and for unit root stability

	HS300	A50	R_diff
X-squared	44.28	5.9418	83.369
df	5	5	5
p-value	2.032e-08	0.3119	< 2.2e-16



ARMA model: Data & Training

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -53.2138
  P VALUE:
    0.01
```

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -51.595
  P VALUE:
    0.01
```

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -56.8594
  P VALUE:
    0.01
```

- Summary of the test results: nice quality of the data
- Precondition: Stationarity & Auto correlation ✓

ARMA model: Data & Training

- Step 3: building the model according to the data

The parameter p and q are determined by several tests.

```
# build ARMA(p,q)model
arma=tsa.ARMA(r_diff, order =(p,q))
model=arma.fit()
```

Coefficients:				
	ar1	ar2	ar3	ma1
	0.6463	0.0640	-0.1461	-0.8634
s.e.	0.0544	0.0424	0.0399	0.0454

ARMA model: Results and Analysis

- model.predict(start, end)

	predict
[1,]	-0.0005323726 -4.458022e-04
[2,]	-0.0008481140 -1.795850e-04
[3,]	-0.0004498944 -2.218592e-04
[4,]	-0.0003581392 -8.975605e-05
[5,]	0.0016422886 -4.597222e-05

Training set error measures:					
	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	-1.128083e-05	0.001238978	0.0009569406	NaN	Inf	0.6224464

Summary

- Although the issue is of great complexity and none of the model manages to solve the problem accurately, there are certain circumstances where we can use machine learning and time series analysis to get accurate predictions.

- For example, the second model is extremely powerful, or at least extremely powerful in some circumstances, in the prediction of return series with certain qualities.

- And SVM can be combined with data mining of social network(Twitter and Weibo) to get people's attitudes towards certain stock, which can also be applied to the prediction of stock price