Case 2 **Yiming Wang**

China Government Bond Future Price Prediction by Kernel SVM

Introduction

Interest rate is the cost of money. In all of the financial asset valuation (stocks, options, foreign currencies, bonds), interest rate is one of the most important factor. Especially in treasury bond (government bond) pricing.

$$P = \sum_{t=1}^{T} \frac{\mathrm{C}}{(1+i)^t} + \frac{F}{(1+i)^T}$$

The formula above is treasury bond (government bond) pricing model.

- P means the present value of the bond (the reasonable price of the bond now).
- F means the face value of the bond. It is a fixed value once the bond issued. Usually, F = 100;
- C means coupon rate (for each bond, the fixed yield paid to bond holders every year) and the value of C would be set as a fixed number when the bond was issued.
- T means the time of maturity (for 10 year bond, T = 10; for 5 year bond, T=5. etc.).
- t means the day time since the bond was issued (e.g. a 10 year bond issued in the begin of 2015, then, in the end of 2015, t=1; in the end of 2016 t=2; etc.).
- i means interest rate of the country.

So, in the bond pricing valuation model, F, C, T are constants when one bond was issued. t would change each year automatically. No need to do any prediction. The only core factor of valuing a treasury bond price, would be the value of i which is the interest rate. From the formula, we also know, interest rate is negative correlated with bond price.

The People's Bank of China (PBoC) would use financial tools to control the interest rate depending on the economic situation of China. Inflation and economic growth are the 2 core economic factors that PBoC would consider before interest rate changing decisions are made. That means, under similar economic situation, similar interest rate decision would be made by PBoC.

In this case study, I would try to build a model to predict interest rate of China. Base on the bond pricing formula if interest rate of the next month goes up, the bond price should go down, investors can short bond future contract now and make profit. If interest rate of the next month goes down, the bond price should go up, investors can long bond future contract now and make profit. The reason I would not do direct prediction of China government bond is that the China government bond future market was open in 2013. There are only 4 years of monthly historical data which is too small to build model. Since interest rate and bond price are high negative correlated and there is more interest rate historical data, I think predict interest rate then indirectly predict government bond price would be more appropriate.

Data Description

I downloaded all the data from Choice Financial Terminal. As I mentioned in last part, inflation status and economic growth are the 2 core economic factors for PBoC interest rate decision making.

Case 2 **Yiming Wang**

I would choose some economic indicators to represent these 2 factors.
Features for Inflation:

Feature	Data Type	Description
CPI%	Numeric	Consumer price index (CPI) measures changes in the price level of market basket of consumer goods and services purchased by households. The annual percentage change in a CPI is used as a measure of inflation.
Core-CPI%	Numeric	Another method for measuring core inflation. It is the consumer price index (CPI) excluding energy and food prices.
M2	Numeric	M2 is a measure of the money supply of a county. Depends on the theory of Milton Friedman, money supply changing would affect future inflation rate of a country.
M2%	Numeric	Annual change of M2

Features for economic growth:

Feature	Data Type	Description
PMI	Numeric	The Purchasing Managers' Index (PMI) is an indicator of the economic health of the manufacturing sector.
GDP%	Numeric	Gross domestic product (GDP) is the monetary value of all the finished goods and services produced within a country's borders in a specific time period. GDP is a broad measurement of a nation's overall economic activity.

Interest Rate (prediction target of the model):
Shibor is the short form of Shanghai Interbank Offered Rate. It is a benchmark interest rate of China announced by People's Bank of China. There are 6 different Shibor, they represent short term interest rate and long term interest rate.

Prediction target	Data Type	Description
ShiborO/N	Numeric	Overnight interest rate.
Shibor1W	Numeric	1 week interest rate.
Shibor1M	Numeric	1 month interest rate.
Shibor3M	Numeric	3 months interest rate.
Shibor6M	Numeric	6 months interest rate.
Shibor1Y	Numeric	1 year interest rate.

I would do transformations to these Shibors in the next part to get direction of interest rate movement.

Bond future price (indirect prediction target):

Prediction target	Data Type	Description
TF00C1	Numeric	5 year china government bond current quarter future contract price.

I would use the predicted interest rate to predict the 5 year government bond price.

Since most of the economic indicators data are monthly data, so, for all of the features and 2 targets, I downloaded the monthly data. Except GDP which is a quarterly data (would do cleaning in next part). Because historical data of Shibor started on Oct 2006 (Shibor announced by PBoC that time), all data would be in the range of Oct 2006 to Apr 2017. Except TF00C1, its historical data start with 2013. I put TF00C1 in a separated data frame, would not use this data frame until the experimental analysis part.

Data Cleaning and Transformation

GDP% is a linear quarterly data. Since it is an average value of a whole quarter, we can do weighted average to get approximately value of each month. What I did:

- GDP% of first month of a quarter = GDP% of that quarter * 2/3 + GDP% of last quarter * 1/3
- GDP% of second month of a quarter GDP% of that quarter
- GDP% of third month of a quarter = GDP% of that quarter * 2/3 + GDP% of next quarter * 1/3

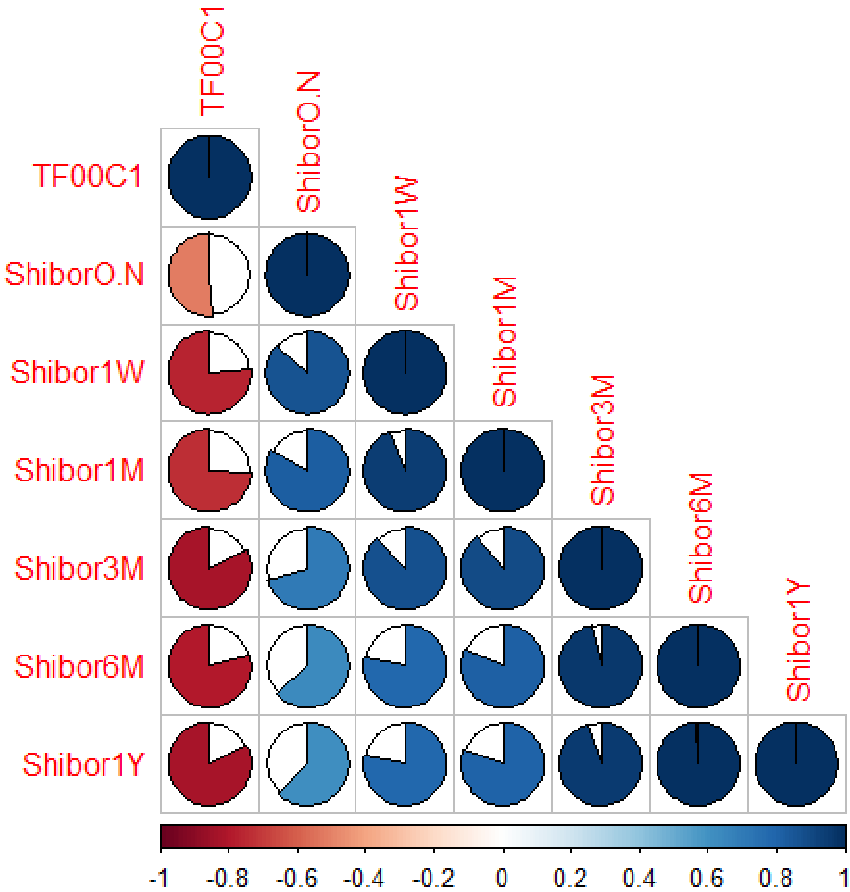
The new monthly GDP% would be a new feature called "GDP%_M"

- For the GDP% of Apr 2017, I use the first quarter of 2017 GDP%.

Because under similar economic situation, similar interest rate decision would be made by PBoC. It is a process of classification. I would require a value of interest rate movement direction: "up", "down" or "same". Then classification can be performed.

There are 6 different kinds of Shibor rate can represent different time duration of interest rate: 1 day, 1 week, 1 month, 3 months, 6 months and 1 year. Since the aim of this case is to finally predict the price of TF00C1 (government bond price). We need a kind of Shibor interest rate highest correlation with TF00C1. I performed correlation matrix:

	TF00C1	*ShiborO/N*	*Shibor1W*	*Shibor1M*	*Shibor3M*	*Shibor6M*	*Shibor1Y*
TF00C1	1						
ShiborO/N	-0.51878	1					
Shibor1W	-0.7623	0.870773	1				
Shibor1M	-0.74071	0.829437	0.944302	1			
Shibor3M	-0.82696	0.701781	0.879617	0.892298	1		
Shibor6M	-0.80013	0.630638	0.791652	0.816508	0.968324	1	
Shibor1Y	-0.82653	0.618112	0.788224	0.806855	0.952204	0.992672	1



Depends on the table and graph above. I would create a new feature called i_next_M to represent the movement of interest rate next month:

- If Shibor1Y next month > Shibor1Y this month, i_next_M of this month would equal to "up".
- If Shibor1Y next month < Shibor1Y this month, i_next_M of this month would equal to "down".

The reason is, Shibor1Y has highest negative correlation with TF00C1.

If the Shibor1Y next month = Shibor1Y this month, we would use Shibor3M (second highest negative correlation with TF00C1) value to decide the i_next_M value:

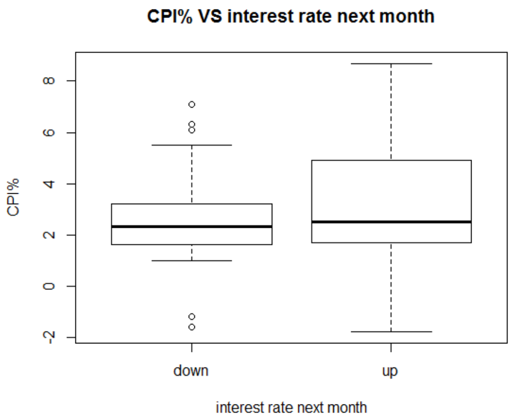
- If Shibor3M next month > Shibor3M this month, i_next_M of this month would equal to "up".
- If Shibor3M next month < Shibor3M this month, i_next_M of this month would equal to "down".

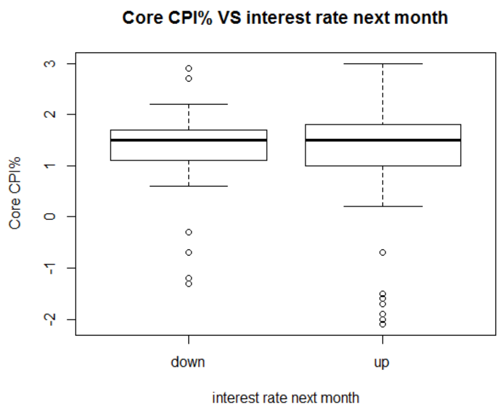
If the Shibor3M next month = Shibor3M this month, we would use Shibor6M (third highest negative correlation with TF00C1) value to decide the i_next_M value. Same rule until ShiborO/N, which has the lowest negative correlation with TF00C1. After this transformation, all i_next_M values are gotten, except the first line, which is for May 2017. Delete that line of i_next_M.

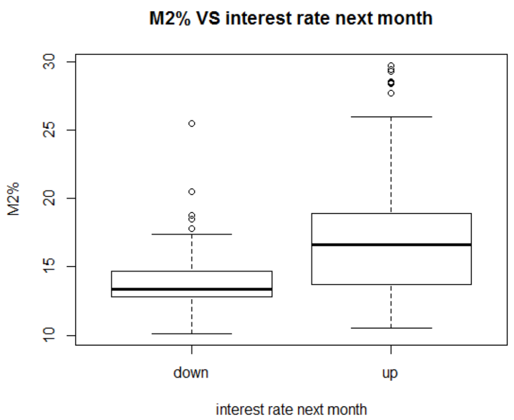
After all of the data cleaning and transformation, I got a clean data set with 6 features, and 127 instances.

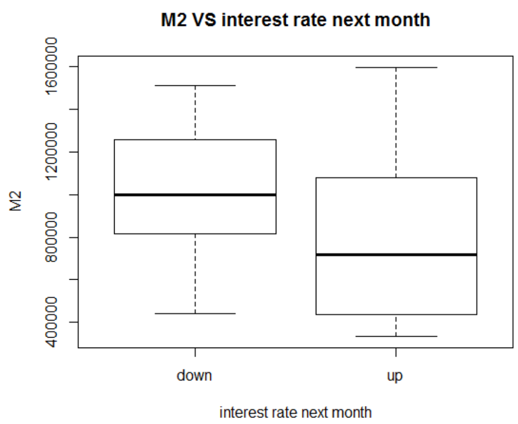
Data Analysis

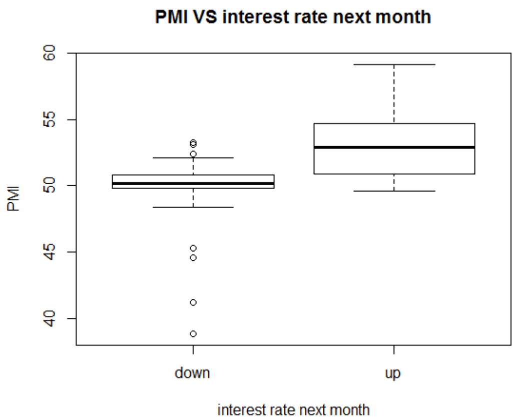
I started with box plotting the features, since all of them are numeric.

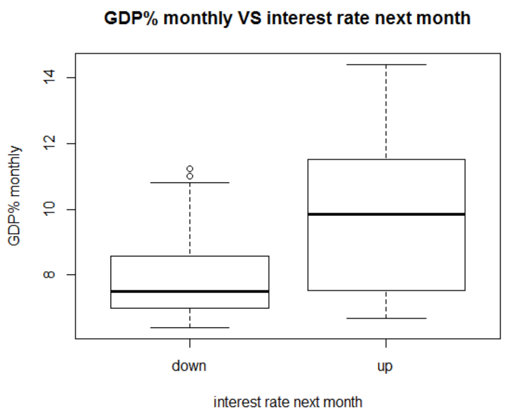






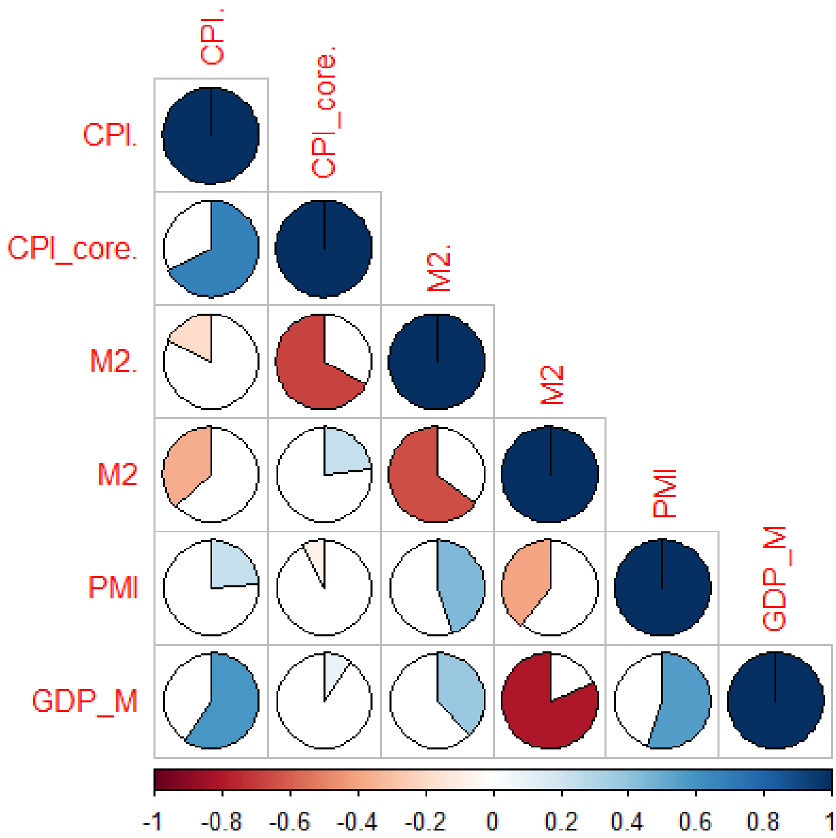


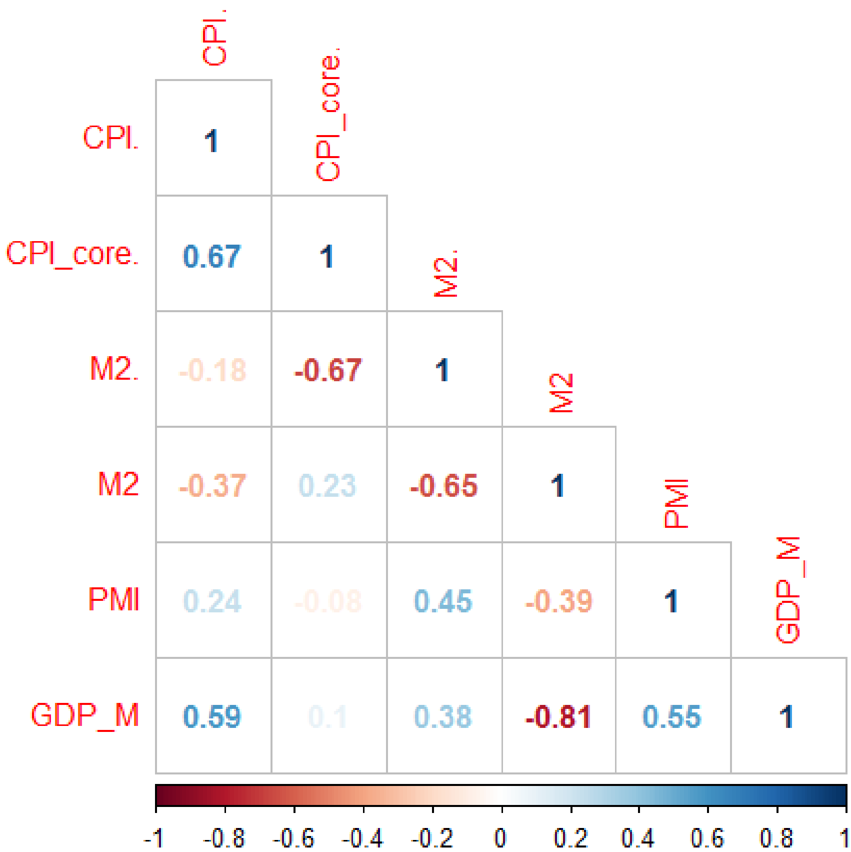




From the boxplots, we can see that, the movement of interest rate of next month tends to go when CPI%, core CPI%, M2%, PMI, and GDP% are high. The interest rate of next month tends to go down if M2 is low.

Then I check the correlations between features:


CPI.
CPI_core.
M2.
M2
PMI
GDP_M
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1


CPI.
CPI_core.
M2.
M2
PMI
GDP_M
1
0.67 1
-0.18 -0.67 1
-0.37 0.23 -0.65 1
0.24 -0.08 0.45 -0.39 1
0.59 0.1 0.38 -0.81 0.55 1
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1

GDP% monthly and M2 are high correlated. This is reasonable. Because for most of emerge country (such as China), investment is one of the key to GDP growth, while money supply (M2) is the power of investment.
CPI% is high correlated with both GDP% monthly and core CPI %.
In the next parts, if I use some high correlation sensitive algorithms, I would remove M2 and CPI% from the features.
Depends on last parts of analysis, a classification algorithm would be most appropriated to build most in this case. Since the data set is relatively small, with only 127 instances. SVM would be most suitable algorithm. I would build model and tuning parameters in the following part.

Experimental Results

I split data into 80% training subset and 20% testing subset.
Starting with linear kernel SVM. Since the linear kernel is sensitive to high correlation features, I removed M2 and CPI%. For the rest of SVM kernels, though they are not high correlation sensitive, I would compare the result with and without M2 and CPI%.

Result of linear kernel SVM after 10 folder cross validation tuning:

	Down	Up	Sensitivity	Specificity	Accuracy	95% CI
Down	9	2	0.6923	0.8333	0.76	(0.5487, 0.9064)
Up	4	10				

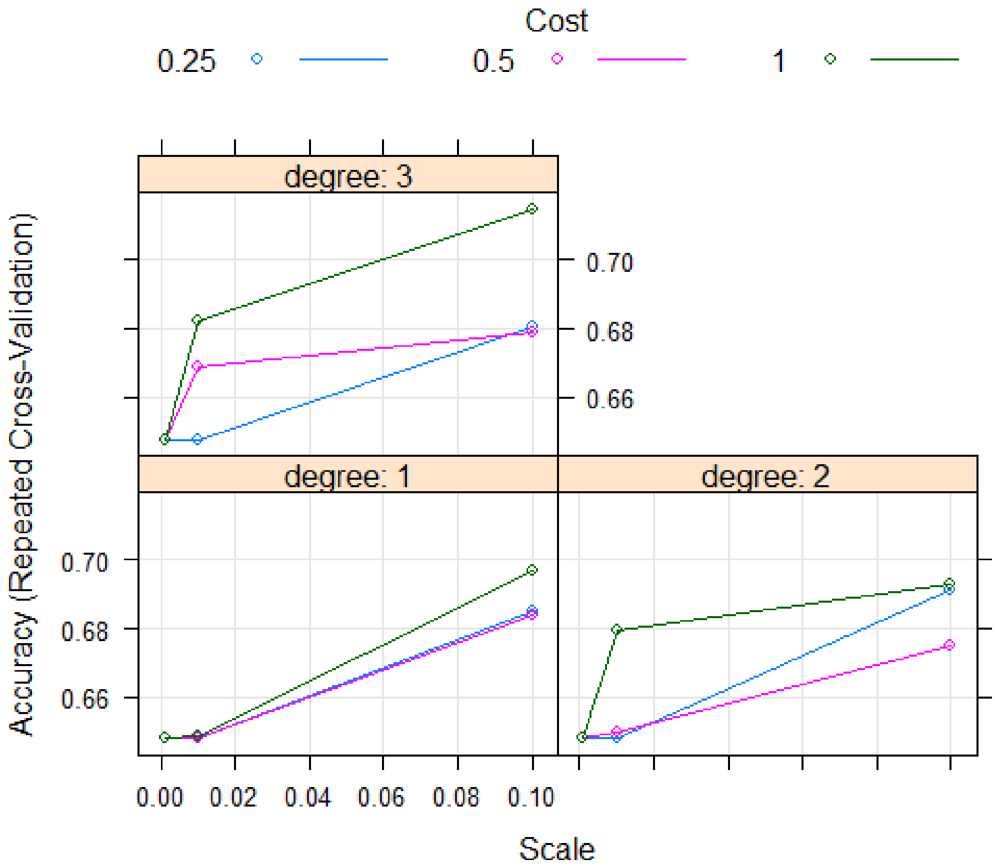
- Best parameter after tuning: cost C = 1
- Number of Support Vectors : 60
- Objective Function Value : -55.0763
- Training error : 0.22549

Since in the final aim of this case is to long/short bond by the prediction of interest rate, so, both of the Sensitivity and Specificity of the model is important. The result of linear kernel SVM is ok on prediction of “UP”, but not good enough on “Down” prediction.

Result of polynomial kernel SVM after 10 folder cross validation tuning (without M2 and CPI%):

	Down	Up	Sensitivity	Specificity	Accuracy	95% CI
Down	9	2	0.6923	0.8333	0.76	(0.5487, 0.9064)
Up	4	10				

- Best parameter after tuning: cost C = 1; degree = 3; scale = 0.1; offset = 1
- Number of Support Vectors : 63
- Objective Function Value : -56.8616
- Training error : 0.196078

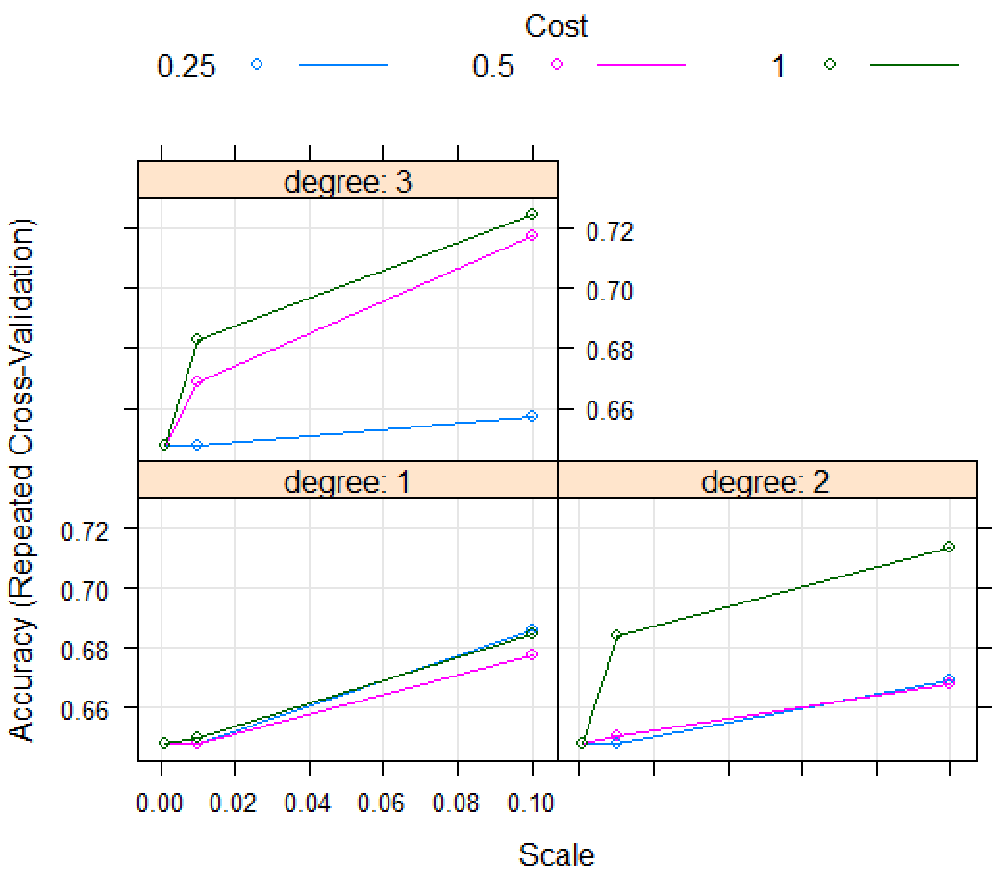


The result is the same with linear kernel SVM. Since the polynomial kernel is not sensitive to high correlation, I would try data which includes M2 and CPI%.

Result of polynomial kernel SVM after 10 folder cross validation tuning (with M2 and CPI%):

	Down	Up	Sensitivity	Specificity	Accuracy	95% CI
Down	10	2	0.7692	0.8333	0.8	(0.593, 0.9317)
Up	3	10				

- Best parameter after tuning: cost C = 1; degree = 3; scale = 0.1; offset = 1
- Number of Support Vectors : 61
- Objective Function Value : -51.9934
- Training error : 0.22549

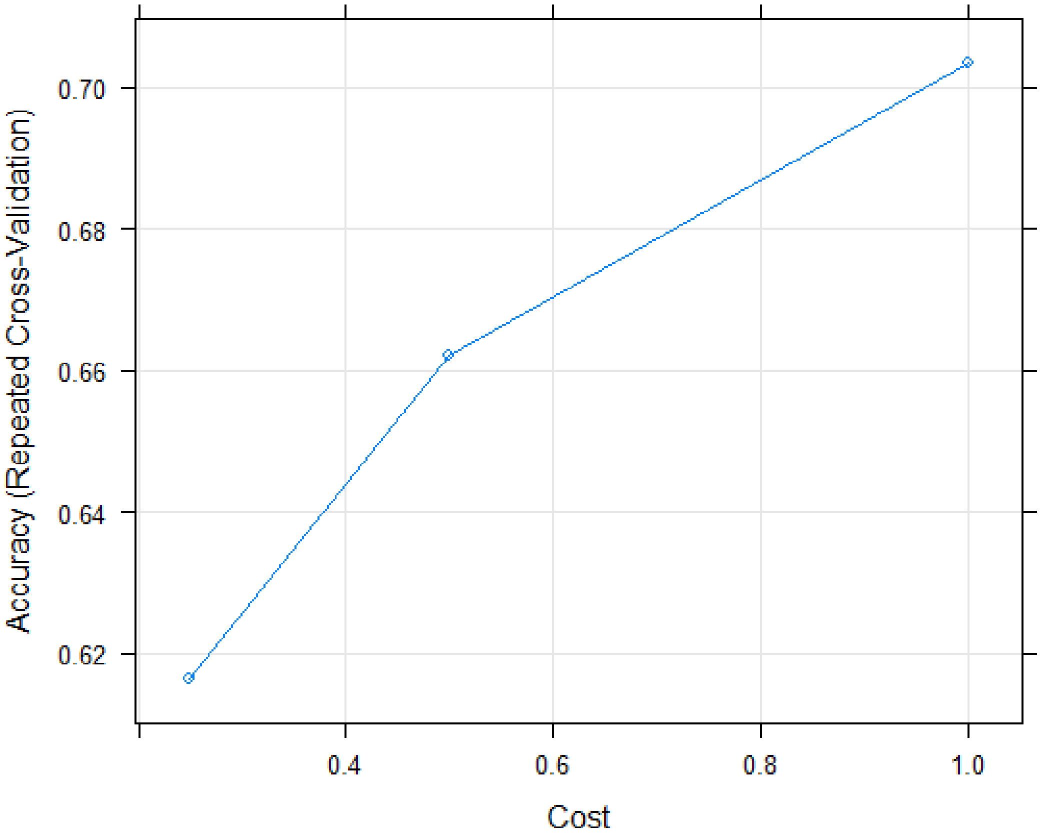


The result of polynomial kernel SVM with M2 and CPI % is better than without M2 and CPI% on the prediction of interest rate goes down. The accuracy of predicting "Down" is better.

Result of radial basis kernel SVM after 10 folder cross validation tuning (without M2 and CPI%):

	Down	Up	Sensitivity	Specificity	Accuracy	95% CI
Down	9	2	0.6923	0.8333	0.76	(0.5487, 0.9064)
Up	4	10				

- Best parameter after tuning: cost C = 1; sigma = 1.66137269556777
- Number of Support Vectors : 83
- Objective Function Value : -51.8134
- Training error : 0.156863

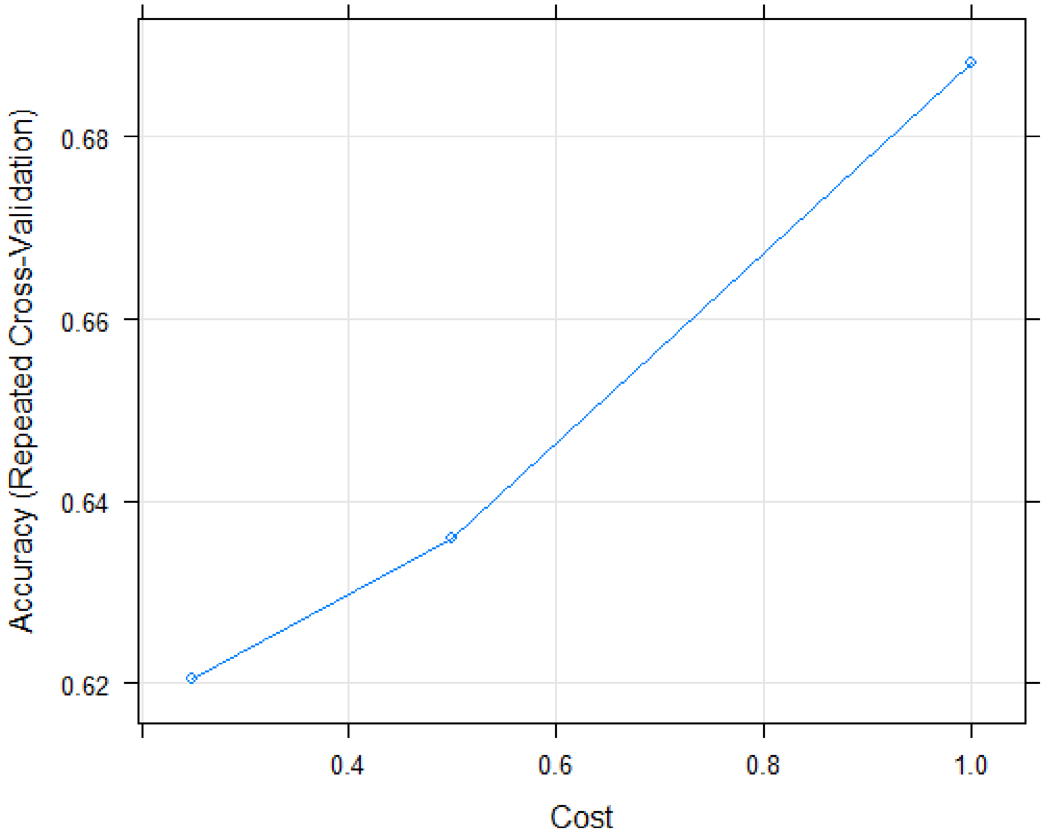


The result is the same with linear kernel SVM. Since the polynomial kernel is not sensitive to high correlation, I would try data which includes M2 and CPI%.

Result of radial basis kernel SVM after 10 folder cross validation tuning (with M2 and CPI%):

	Down	Up	Sensitivity	Specificity	Accuracy	95% CI
Down	9	1	0.6923	0.9167	0.8	(0.593, 0.9317)
Up	4	11				

- Best parameter after tuning: cost C = 1; sigma = 0.887225361131005
- Number of Support Vectors : 84
- Objective Function Value : -51.2706
- Training error : 0.156863



This model returned best result on predicting “Up”, but the prediction of “Down” is 0.6923 which is not good enough.

Considering the balance of both good performance on predicting “Up” and “Down”, I think the polynomial kernel SVM (C =1, degree =3, scale =0.1, offset =1) built from all the features (including M2 and CPI%) is the best model.

Experimental Analysis

Now I would use the interest rate prediction result to predict the TF00C1 trading profit.
I create a new variable called “profit”:
Profit = TF00C1 value the next month - TF00C1 value this month
As I mentioned in data description part, the historical data of TF00C1 is only 44 months, it is why I have to do indirectly prediction of the TF00C1 price by interest rate.
I pulled out the index of instances chosen as test set. Then, if the “Profit” data exist on that index (that month), I added the profit value and i_next_M into new data frame testProfit.
In the new data frame testProfit, I would test the potential profit:

- If the model predicts the interest rate of next month equals to “up”, then it means the bond price would go down depending on the bond pricing formula. So, short sell bond. Otherwise, buy in bond. If the prediction is correct we make profit. Add the absolute value of “profit” into “overall profit". Otherwise, we lose money. Subtract the absolute value of "profit" from "overall profit".

- If the model predicts the interest rate of next month equals to "down", then it means the bond price would go up depending on the bond pricing formula. So, buy in bond. Otherwise, short sell bond. If the prediction is correct we make profit. Add the absolute value of "profit" into "overall profit". Otherwise, we lose money. Subtract the absolute value of "profit" from "overall profit".

The result of overall profit:

Profit	i_next_M	Actual Profit
1.22	up	-1.22
-0.325	up	0.325
0.08	up	-0.08
0.465	down	0.465
0.56	down	0.56
0.7	down	0.7
1.565	up	-1.565
0.305	down	0.305
1.366	down	1.366
0.782	down	0.782
Overall Profit		**1.638**

From the table, we can see in the 10 month's testing data, it make profit of 1638 points. Since the TF00C1 future contracts are on 5% margin rate (20 times of leverage), it could be also approximately calculated as 1.638% * 20 = 32.76% profit in 10 month. Which is really good.

But, in the real world of trading, this model / strategy should be used very carefully. Because:

1. Since the future contracts offers 5% margin rate. The strategy of this model, always holding the positions for 1 month, it is possible that some extremely high volatilities in some trading hours in that month, which might cause margin call or even force close on position (huge loss) even though we successfully predict the direction of interest rate movement of this month.
2. The accuracy of prediction is good. The model performed well on both Sensitivity and Specificity. However, it is very possible that one single month of wrong prediction causes huge loss and eats up all the previous profits. For example, the model predicts correctly in continuous 10 months, and makes 2000 points of profit. In the 11^{th} month, financial crisis happens, and the model makes wrong prediction this month. This single month the whole account loss 3000 points. Which means 2000 – 3000 = -1000 net loss.

Conclusion

Based on the data and the analysis we have conducted, there are some meaningful conclusions:

- From the idea of "under similar economic situation, similar interest rate decision would be made by PBoC", we can build classification model to predict the interest rate of next month. In this case, due to the data set size, we only test 3 different kernel of SVM algorithms. The best model is polynomial kernel SVM with parameter cost C = 1, degree = 3, scale = 0.1 and offset = 1. The accuracy of this model is 0.8.

- Since the bond price are high negative correlated with interest rate, we can use the predicted interest rate to trade the bond future contracts. In this case, the potential profit of testing data set (10 month) is 1638 points. Which is good.

As I mentioned in the end of experimental analysis part, it is possible that margin call or force close happens due to high volatility of price even though the model predicts correctly on the direction of interest rate movement. It is also possible that, only one single month of incorrectly prediction cause a huge loss. So, the further study could work on the “stop loss” of the strategy. Try statistics or some data mining techniques to find a best value of “stop loss”.

Lastly, in the future, when there are enough data on TF00C1 future contract, we could try to directly predict the price of TF00C1. Probably the accuracy can be higher than the indirectly prediction method I used in this case. What is more, with the larger data set, some other classification algorithms can be used, such as decision tree and random forest.

Appendix

```
#import economic indicators data set
df <- read.csv("e:/Case2Bond1.csv")
head(df)

nrow(df)
df$GDP_M = 0
head(df)
for (i in 3:(nrow(df)-2)){
  if (i %% 3 ==0){
    df$GDP_M[i] <- df$GDP[i-1]
  }
  if (i %% 3 ==1){
    df$GDP_M[i] <- 2/3 * df$GDP[i-2] + 1/3 * df$GDP[i+1]
  }
  if (i %% 3 ==2){
    df$GDP_M[i] <- 2/3 * df$GDP[i] + 1/3 * df$GDP[i-3]
  }
}
df$GDP_M[1] <- df$GDP[2]
df$GDP_M[2] <- df$GDP[2]
df$GDP_M[nrow(df)] <- df$GDP[nrow(df)-2]
df$GDP_M[nrow(df)-1] <- df$GDP[nrow(df)-2]

head(df)
tail(df)
```

Case 2 **Yiming Wang**

```
#load Shibors (interest rate data set) and TF00C1 (5 year government bond) dataset
Shibors <- read.csv("e:/ShiborsVSbond.csv")
head(Shibors)

library(corrplot)
M <- cor(Shibors)
M
corrplot(M, method="pie", type = "lower")

Shibors$i_next_M = NA
Shibors
for (i in 2:nrow(Shibors)){
  if (Shibors$Shibor1Y[i] < Shibors$Shibor1Y[i-1]){
    Shibors$i_next_M[i] = "up"
  }
  else if (Shibors$Shibor1Y[i] > Shibors$Shibor1Y[i-1]){
    Shibors$i_next_M[i] = "down"
  }
  else{
    if (Shibors$Shibor3M[i] < Shibors$Shibor3M[i-1]){
      Shibors$i_next_M[i] = "up"
    }
    else if (Shibors$Shibor3M[i] > Shibors$Shibor3M[i-1]){
      Shibors$i_next_M[i] = "down"
    }
    else{
      if (Shibors$Shibor6M[i] < Shibors$Shibor6M[i-1]){
        Shibors$i_next_M[i] = "up"
      }
      else if (Shibors$Shibor6M[i] > Shibors$Shibor6M[i-1]){
        Shibors$i_next_M[i] = "down"
      }
      else{
        if (Shibors$Shibor1W[i] < Shibors$Shibor1W[i-1]){
          Shibors$i_next_M[i] = "up"
        }
        else if (Shibors$Shibor1W[i] > Shibors$Shibor1W[i-1]){
          Shibors$i_next_M[i] = "down"
        }
        else{
          if (Shibors$Shibor1M[i] < Shibors$Shibor1M[i-1]){
            Shibors$i_next_M[i] = "up"
          }
```

```
            else if (Shibors$Shibor1M[i] > Shibors$Shibor1M[i-1]){
              Shibors$i_next_M[i] = "down"
            }
            else{
              if (Shibors$ShiborO.N[i] < Shibors$ShiborO.N[i-1]){
                Shibors$i_next_M[i] = "up"
              }
              else {
                Shibors$i_next_M[i] = "down"
              }
            }
          }
        }
      }
    }
}

Shibors

Sh <- cbind (df[2:6],df[9],Shibors[8])
sapply(sh,class)


boxplot(CPI_core. ~ i_next_M, data = sh, main= "Core CPI% VS interest rate next month",
        xlab = "interest rate next month", ylab = "Core CPI%")
boxplot(CPI_core. ~ i_next_M, data = sh, main= "Core CPI% VS interest rate next month",
        xlab = "interest rate next month", ylab = "Core CPI%")
boxplot(M2. ~ i_next_M, data = sh, main= "M2% VS interest rate next month",
        xlab = "interest rate next month", ylab = "M2%")
boxplot(M2 ~ i_next_M, data = sh, main= "M2 VS interest rate next month",
        xlab = "interest rate next month", ylab = "M2")
boxplot(PMI ~ i_next_M, data = sh, main= "PMI VS interest rate next month",
        xlab = "interest rate next month", ylab = "PMI")
boxplot(GDP_M ~ i_next_M, data = sh, main= "GDP% monthly VS interest rate next month",
        xlab = "interest rate next month", ylab = "GDP% monthly")

summary(sh)
M.sh <- cor(sh[1:6])
M.sh
corrplot(M.sh,method="pie", type = "lower")
corrplot(M.sh,method="number", type = "lower")


#20% test, 80% training
```

Case 2 **Yiming Wang**

```
#library(rpart)
#library(caret)
set.seed(123)
ind <- sample(2, nrow(sh), replace=TRUE, prob=c(0.8, 0.2))
train <- sh[ind==1,]
test <- sh[ind==2,]

head(train)

#Remove M2 and CPI%
train_r <- cbind(train[2:3],train[5:7])
test_r <- cbind(test[2:3],test[5:7])
library(e1071)


#linear kernel SVM with 10 fold cross validation tuning
fitControl <- trainControl(method = "repeatedcv",number = 10,repeats = 10)
set.seed(123)
svm_l <- train(i_next_M ~ ., data = train_r, method = "svmLinear", trControl = fitControl,verbose
= FALSE)
svm_l$finalModel
plot(svm_l)

pred <- predict(svm_l,test_r)
t <- table(pred,test_r$i_next_M)
#library(caret)
confusionMatrix(t)



#polynomial kernel SVM with 10 fold cross validation tuning (without M2 and CPI%)
set.seed(123)
svm_p <- train(i_next_M ~ ., data = train_r, method = "svmPoly", trControl = fitControl,verbose =
FALSE)
svm_p$finalModel
plot(svm_p)

pred <- predict(svm_p,test_r)
t <- table(pred,test_r$i_next_M)
confusionMatrix(t)


#polynomial kernel SVM with 10 fold cross validation tuning (with M2 and CPI%)
set.seed(123)
```

```
svm_p <- train(i_next_M ~ ., data = train, method = "svmPoly", trControl = fitControl,verbose =
FALSE)
svm_p$finalModel
plot(svm_p)

pred <- predict(svm_p,test)
t <- table(pred,test$i_next_M)
confusionMatrix(t)


#radial basis kernel SVM with 10 fold cross validation tuning (without M2 and CPI%)
set.seed(123)
svm_p <- train(i_next_M ~ ., data = train_r, method = "svmRadial", trControl = fitControl,verbose
= FALSE)
svm_p$finalModel
plot(svm_p)

pred <- predict(svm_p,test_r)
t <- table(pred,test_r$i_next_M)
confusionMatrix(t)


#radial basis kernel SVM with 10 fold cross validation tuning (with M2 and CPI%)
set.seed(123)
svm_p <- train(i_next_M ~ ., data = train, method = "svmRadial", trControl = fitControl,verbose =
FALSE)
svm_p$finalModel
plot(svm_p)

pred <- predict(svm_p,test)
t <- table(pred,test$i_next_M)
confusionMatrix(t)


Shibors$profit = 0

for (i in 2:nrow(Shibors)){
  Shibors$profit[i] = Shibors$profit[i-1] -Shibors$profit[i]
}
testProfit <- Shibors[ind==2,]
testProfit
testProfit <- testProfit [1:10,]
test_p = test [1:10,]
testProfit <- cbind(testProfit,test_p)
```

```
testProfit <- cbind(testProfit[8],testProfit[15])
testProfit

testProfit$A_P =0
for (i in 1:nrow(testProfit)){
    if (testProfit$i_next_M[i]=="up" & testProfit$Profit[i]<0){
        testProfit$A_P[i] = abs (testProfit$Profit[i])
    }else if(testProfit$i_next_M[i]=="down" & testProfit$Profit[i]>0){
        testProfit$A_P[i] = abs (testProfit$Profit[i])
    }else{
        testProfit$A_P[i] = - abs (testProfit$Profit[i])
    }
}
testProfit
overallProfit = sum(testProfit[3])
overallProfit
```