Clustering of stock time series data and price prediction

Yuhao Kang

Physics department, Graduate Center, CUNY

Stock price prediction is of great fundamental interest and has been studied for decades. Here, based on the clustering of stock price fluctuation, we try to predict the stock variation in short period. This approach was first proposed in Ref.[1].

The dataset is extracted from kaggle[2]. It contains the daily data of open, high, low, close and volume information of the whole US stocks until 2017/11/10. Fig. 1 shows the data of AAMC. In the following text, we will take it as an example.

	Date	Open	High	Low	Close	Volume	OpenInt
1	2012-12-13	15	15	15	15	100	0
2	2012-12-14	19	30	19	30	144600	0
3	2012-12-17	31.5	65	31.5	65	68600	0
4	2012-12-18	65	89	65	80	43600	0
5	2012-12-19	80	84	78	84	24000	0
6	2012-12-20	84	84	80	80.25	33300	0
7	2012-12-21	80.5	81.5	75	80	20700	0
8	2012-12-24	81	93.5	80	80	3700	0
9	2012-12-26	80	90	77	77	56100	0
10	2012-12-27	75	75.02	55	59	54300	0
11	2012-12-28	58.5	75	56.5	72.35	94400	0
12	2012-12-31	72	90	72	82	43500	0
13	2013-01-02	83.5	85	75.75	76	26000	0
14	2013-01-03	75.5	83.9	73.5	81.25	80800	0
15	2013-01-04	81.5	86	81.5	85	26000	0
16	2013-01-07	86	87.75	79.8	79.8	31800	0
17	2013-01-08	79	83.99	78.25	80	21800	0
18	2013-01-09	80.75	86.25	80.75	84	53100	0
19	2013-01-10	86.8	99.49	86	98	38900	0
20	2013-01-11	100	107	97.15	106.5	37000	0
21	2013-01-14	111	120	110	120	42000	0
22	2013-01-15	117.05	129.49	108	110	97600	0

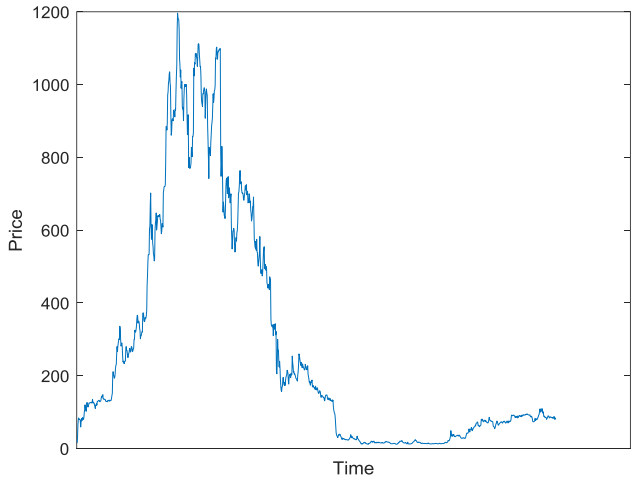


Fig. 1. Time series data of AAMC. (Left panel) original data, here we only focus on the close column. (Right panel) run chart based on the data, time range is 2012/12/13—2017/11/10.

The basic idea is that we suppose the time series data contains a short-range correlation. The real price $y(t)$ can be decompose into two part $g(t) + c(t)$. $g(t)$ represents the whole tendency of the stock which largely depends on the economic environment and the growing of the company, while $c(t)$ indicates the fluctuations around the main line and can be seemed as cyclical and predictable. The short-term operation benefits from the regularity of the $c(t)$ term.

Preprocessing data and regression trees

At the certain time t, we consider the 30-days-ahead series $p(t) = [y(t), y(t-1) \dots y(t-29)]$ as the predictor, and the 5-days-after data $r(t) = [y(t+1), y(t+2) \dots y(t+5)]$ as the result.

Then we construct a 30×35 matrix:

$$[P\ R] = \begin{bmatrix} p(t) & r(t) \\ p(t-1) & r(t-1) \\ \vdots & \vdots \\ p(t-29) & r(t-29) \end{bmatrix}$$

P has 30 columns $V_1 \sim V_{30}$, each of which can be seemed as input features. *R* contains five responses $V_{31} \sim V_{35}$ of these 30 inputs.

In real life, when the interval of series is in minute-scale, the size of this matrix can reach $\sim 10^3 \times 10^3$. Thus, it is necessary to extract the most important inputs of *P*, which are highly related to the result of *R*. Regression trees[3] are used here to classify the relevant features.

```
Call:
rpart(formula = V31 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 +
    V9 + V10 + V11 + V12 + V13 + V14 + V15 + V16 + V17 + V18 +
    V19 + V20 + V21 + V22 + V23 + V24 + V25 + V26 + V27 + V28 +
    V29 + V30, data = xt, method = "anova")
  n= 30

          CP nsplit rel error    xerror      xstd
1 0.62179905      0 1.0000000 1.0972339 0.3162066
2 0.03983376      1 0.3782010 0.5438167 0.1511534
3 0.01000000      2 0.3383672 0.5446017 0.1501308

Variable importance
 V7  V8  V1  V6  V2  V5 V17 V18 V19  V9
 30  15  14  14  11  11   2   1   1   1

Node number 1: 30 observations,    complexity param=0.621799
  mean=2.542333, MSE=0.01068456
  left son=2 (22 obs) right son=3 (8 obs)
  Primary splits:
      V7  < 2.475  to the right, improve=0.6217990, (0 missing)
      V24 < 2.595  to the left,  improve=0.4437202, (0 missing)
      V6  < 2.475  to the right, improve=0.4130582, (0 missing)
      V8  < 2.505  to the right, improve=0.3743746, (0 missing)
      V5  < 2.475  to the right, improve=0.3675202, (0 missing)
  Surrogate splits:
      V1 < 2.705  to the left,  agree=0.867, adj=0.500, (0 split)
      V6 < 2.465  to the right, agree=0.867, adj=0.500, (0 split)
      V8 < 2.465  to the right, agree=0.867, adj=0.500, (0 split)
      V2 < 2.73   to the left,  agree=0.833, adj=0.375, (0 split)
      V5 < 2.455  to the right, agree=0.833, adj=0.375, (0 split)

Node number 2: 22 observations,    complexity param=0.03983376
  mean=2.493182, MSE=0.001548967
  left son=4 (11 obs) right son=5 (11 obs)
  Primary splits:
      V17 < 2.525  to the right, improve=0.3746832, (0 missing)
      V18 < 2.55   to the right, improve=0.2946512, (0 missing)
      V9  < 2.61   to the left,  improve=0.2653112, (0 missing)
      V11 < 2.61   to the left,  improve=0.2166822, (0 missing)
      V16 < 2.5825 to the right, improve=0.2042735, (0 missing)
  Surrogate splits:
      V9  < 2.61   to the left,  agree=0.818, adj=0.636, (0 split)
      V18 < 2.55   to the right, agree=0.818, adj=0.636, (0 split)
      V19 < 2.55   to the right, agree=0.818, adj=0.636, (0 split)
      V7  < 2.595  to the left,  agree=0.773, adj=0.545, (0 split)
      V8  < 2.595  to the left,  agree=0.773, adj=0.545, (0 split)
```

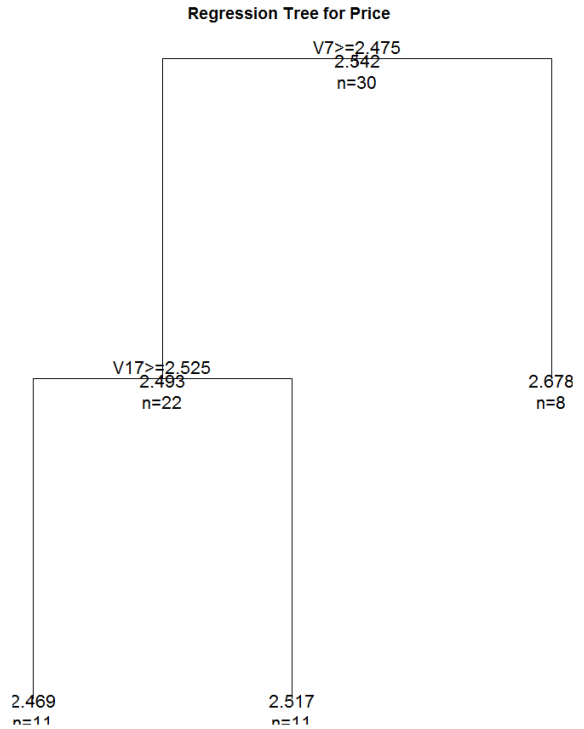


Fig. 2. Regression trees

Combining the important variables corresponding to outputs $V_{31\sim35}$, we obtain the list of principle inputs (In code, it is called 'tt'). In the example of AAMC,

$$tt = [\text{"}V15\text{"}\ \text{"}V14\text{"}\ \text{"}V1\ \ \text{"}V16\text{"}\ \text{"}V19\text{"}\ \text{"}V20\text{"}\ \text{"}V13\text{"}\ \text{"}V12\text{"}\ \text{"}V18\text{"}\ \text{"}V11\text{"}\ \text{"}V17\text{"}\ \text{"}V10\text{"}\ \text{"}V9\text{"}\ \ \text{"}V30\text{"}]$$

Which is far smaller than *P.*

So far, we reduce the dimension of predictor $P_{reduced} = \{V_i\ for\ i\ in\ tt\} \subseteq P$

$final = [P_{reduced}\ \ R]$ is the processed data which contains the features filtered by regression trees and all result vectors. We will impose Self-Organizing Maps and k-means algorithm on the data. The short-range correlation of the time series data makes the clustering meaningful.

SOM

Data frame $final$ has 30 rows, each of them is a high dimensional points. SOM is an efficient way to map the high dimensional data to low dimensional grids. We set a 2×3 grid in this case (Fig. 3). As a result, six centroids are generated: $C_{i=1\sim6} = [P_{ci}\ \ R_{ci}]$

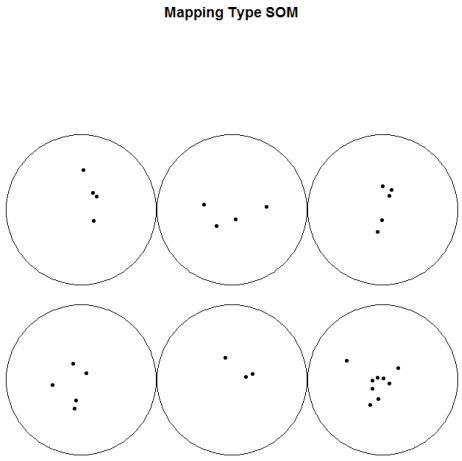


Fig.3. SOM mapping of dataset *final*

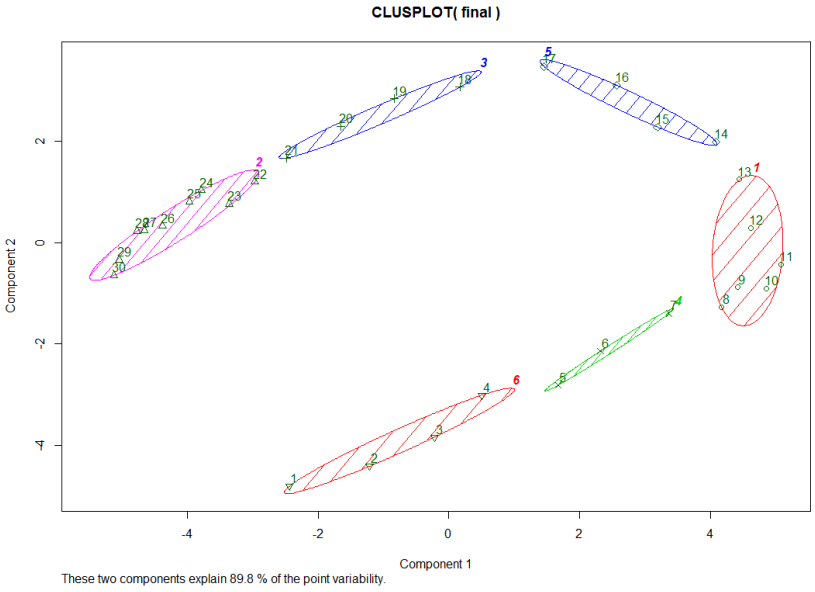


Fig.4. k-means clustering

We define that if $\max(R_c) > mean(P_c)$, the stock will have higher price in the future 5-days (belongs to category 1), or it will go down (belongs to category 0). The six centroids are separated into these two categories.

K-Means

k-means algorithm was also used to do the clustering (Fig. 4). Similarly, we obtains the coordinates of 6 center points and distinguish them into two categories 0 and 1 following the aforementioned definition.

Prediction

Considering a 30-days series data $W(t = 30\sim1)$, we want to determine the category it belongs to. The closest centroid A relative to W can be decided using the distance matrix, which is constructed by Euclidean method. Then W is dropped into the cluster of A, whose category has been already decided.

To test our model, we process the data in Aug. and Sep. 2017 and get the 6 centroids. Then we input a 30-days test data (Sep. 2017) and do the prediction of the beginning 5-days in Oct. Next, we sweep the 30-days windows afterwards and predict the next 5-days. Here, we try 20 test windows in total. Fig.5 shows the prediction results of AAPL using K-means method, SOM respectively, and their comparisons with the reality.

Conclusion

Neither k-means nor SOM has good performance in this case. In the future, we can try various cluster numbers to improve the prediction. In addition, the data set is still small (the interval is 1 day). If apply the method to series data containing the values of every minute, the clustering will be more convincible and the data will have a stronger correlation.

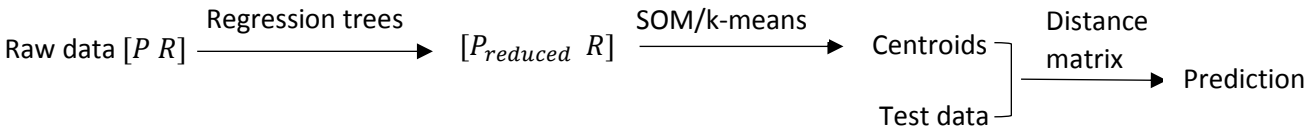
	prediction	reality	correctness
1	1	1	TRUE
2	1	1	TRUE
3	1	1	TRUE
4	0	1	FALSE
5	0	1	FALSE
6	0	1	FALSE
7	0	1	FALSE
8	0	1	FALSE
9	0	1	FALSE
10	0	1	FALSE
11	0	1	FALSE
12	0	1	FALSE
13	0	1	FALSE
14	0	1	FALSE
15	0	1	FALSE
16	0	0	TRUE
17	0	0	TRUE
18	0	0	TRUE
19	0	0	TRUE
20	0	0	TRUE

	prediction	reality	correctness
1	1	1	TRUE
2	0	1	FALSE
3	0	1	FALSE
4	0	1	FALSE
5	0	1	FALSE
6	0	1	FALSE
7	0	1	FALSE
8	0	1	FALSE
9	0	1	FALSE
10	0	1	FALSE
11	0	1	FALSE
12	0	1	FALSE
13	0	1	FALSE
14	0	1	FALSE
15	0	1	FALSE
16	0	0	TRUE
17	0	0	TRUE
18	0	0	TRUE
19	0	0	TRUE
20	0	0	TRUE

Fig.5. Prediction of AAPL. Left panel is k-means method; right panel is SOM

Appendix

a. Methods



b. Code

```
library(R.matlab)
library(gdata)
library(rpart)
library(kohonen)
library(tidyverse)
library(cluster)
```

```
# import data
file<- 'C:/Users/user/Documents/data_mining course/stock/stock/aapl.us.txt'
df<-read.delim2(file, header = TRUE, sep = ",", dec = ",")
head(df)
df_c<-as.matrix(as.numeric(as.character(df$Close)))
writeMat('serv.mat', df_c=df_c)
%%%%% matlab code, generate raw data [P R]
clear
duration=5;
load('serv.mat');
a=flipud(df_c);
% n=size(df_c,1);
x=[];
t=[];
test=[];
future=[];
pro=[];
for i=1:30
   x(:,i)=a(30+i:59+i);
end
for j=1:duration
   t(:,j)=a(30-j:59-j);
end
ans=[x t];
for k=1:20
   test(:,k)=a(10+k:39+k);
   future(:,k)=a(5+k:9+k);
   pro(k)=(max(a(5+k:9+k))>mean(a(10+k:39+k)));
end

csvwrite('serv.csv',ans)
csvwrite('test.csv',test')
csvwrite('pro.csv',pro)

plot(df_c)
set(gca,'XTick',[])
xlabel('Time')
ylabel('Price')
%%%%%

xt <- read.csv(file="serv.csv", header=FALSE)
test <- read.csv(file="test.csv", header=FALSE)
pro <- as.numeric(read.csv(file="pro.csv", header=FALSE))
# regression trees
```

```
fit1<-rpart(V31~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13
        +V14+V15+V16+V17+V18+V19+V20+V21+V22+V23+V24
        +V25+V26+V27+V28+V29+V30,data=xt,method="anova")
fit2<-rpart(V32~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13
         +V14+V15+V16+V17+V18+V19+V20+V21+V22+V23+V24
         +V25+V26+V27+V28+V29+V30,data=xt,method="anova")
fit3<-rpart(V33~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13
         +V14+V15+V16+V17+V18+V19+V20+V21+V22+V23+V24
         +V25+V26+V27+V28+V29+V30,data=xt,method="anova")
fit4<-rpart(V34~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13
         +V14+V15+V16+V17+V18+V19+V20+V21+V22+V23+V24
         +V25+V26+V27+V28+V29+V30,data=xt,method="anova")
fit5<-rpart(V35~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13
         +V14+V15+V16+V17+V18+V19+V20+V21+V22+V23+V24
         +V25+V26+V27+V28+V29+V30,data=xt,method="anova")
printcp(fit1)
plotcp(fit1)
summary(fit1)
plot(fit1, uniform=TRUE,
   main="Regression Tree for Price ")
text(fit1, use.n=TRUE, all=TRUE, cex=1.3)
t1<-names(fit1$var)
t2<-names(fit2$var)
t3<-names(fit3$var)
t4<-names(fit4$var)
t5<-names(fit5$var)
t<-c(t1,t2,t3,t4,t5)
tt<-unique(t)     # obtain principle features of predictor
x_se<-xt[,tt]
t_se<-xt[,c(31:35)]
final<-as.matrix(cbind(x_se,t_se))

#SOM
som_grid <- somgrid(xdim = 3, ydim=2, topo="rectangular") # define rectangular type grid
som_model <- som(final, grid=som_grid, rlen=150, alpha=c(0.05,0.01),
             keep.data = TRUE)
plot(som_model, type = "mapping", pchs = 20, main = "Mapping Type SOM")
cen<-data.frame(som_model$codes)

#K-means
kct <- kmeans(final, 6)
cen<-data.frame(kct$centers)
clusplot(final, kct$cluster, color=TRUE, shade=TRUE,
       labels=2, lines=0)
```

```
# cluster centroids to two categories
n<-length(cen)
cen$mean1 <- apply(cen[, c(1:(n-5))], 1, mean)
cen$max2 <- apply(cen[, c((n-4):n)], 1, max)
cen<-cen %>%
  mutate(profit=(max2>mean1))
test_final<-test[,tt]
cen_x<-rbind(cen[,1:length(tt)],test_final)
# distance matrix, decide test data belong to which cluster
dis<-dist(cen_x, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
print(dis, diag = NULL, upper = NULL,
    digits = getOption("digits"), justify = "none",
    right = TRUE)
dis<-as.matrix(dis)
clus<-apply( dis[,1:6], 1, which.min)
pro2<-as.numeric(cen$profit[clus[-c(1:6)]])
# predict based on the category of the cluster
predict<-data.frame('prediction'=pro2,'reality'=pro) %>%
  mutate(correctness=(pro2==pro))
```

References

1. Nair, B. B., Kumar, P. K. S., Sakthivel, N. R. & Vipin, U. Clustering stock price time series data to generate stock trading recommendations: An empirical study. *Expert Syst. Appl.* **70,** 20–36 (2017).
2. Huge Stock Market Dataset. Available at: https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs.
3. Classification & Regression Trees. Available at: http://www.di.fc.ul.pt/~jpn/r/tree/tree.html.