林禹
14353189

利用 R 语言预测股票市场回报率

一、 选题背景

股市交易是数据挖掘的一大重要应用领域。金融行业有着十分丰富的历史数据，通过对这些历史数据的挖掘，人们能够发现一些隐藏在历史数据中的“宝矿”。经济学的有效市场假设认为，股市交易没有一劳永逸的策略，所有的市场参与者都高度关注市场价格走向，并且总能迅速的对价格变化作出应对。但理想情况下的市场有效是很难达到的，市场中总存在着一些非效率的地方，而这些地方就蕴藏着巨大的商机。

“Fama-French 3 因子模型”的出现对之前使用的股票市场 beta[1]值构成了挑战，认为上市公司的市值、账面市值比、市盈率可以更好地解释股票回报率的差异，受此启发，人们开始将精力集中到对可能影响股价的因素的挖掘上，并在挖掘出因素的基础上，通过回归分析等方法，对不同的因素进行筛选、组合，最终形成一个能够盈利的投资策略。随着大数据的发展，人们更加深入地对以往金融数据进行挖掘，企图在数据矿井中挖到宝藏，一旦发现了能够显著解释某一支股票甚至某一指数的走势的因子或因子组合，可以说就发现了金融数据宝矿中的金矿。

所以，借鉴前人的研究经验，本文旨在利用 R 语言对 S&P500（标普 500）的历史（1970.01.01-2017.05.31）数据进行数据挖掘，挖掘对股价影响较为显著的变量，并根据数据挖掘的结果进行市场回报的预测。

二、 分析逻辑

交易的基本思路是建立一个能够成功预测接下来 n 天的价格走势的模型，根据预测的结果选择是卖出、买入还是继续持有。

假设价格变动超过了$\pm p\%$，就进行交易（因为交易需要一定的手续费，所以只有当价格变动所带来的收益超过手续费是，交易才能真正带来收益）为了建立模型，首先，定义平均日价为:

$$\overline{P_i} = \frac{C_i + H_i + L_i + AC_i}{3}$$

[1] CAPM 模型中，市场的风险为 bata 值

其中，C_i、H_i、L_i、AC_i分别代表每日的收盘价、最高价、最低价、调整收盘价。

同时，将之后 n 天的价格变动定义为集合

$$V_i = \left\{ \frac{\overline{P}_{i+j} - \overline{P}_i}{\overline{P}_i} \right\}_{j=1}^{n}$$

所以我们所需要的交易价格信号集合就是变化范围超过$p\%$的集合：

$$T = \sum \{v \in V_i : |v| > p\%\}$$

得到集合 T 之后，就有了基于历史数据的，对应的 i 天接下来 n 天的价格走势，下一步是将这一集合作为目标变量，选择不同的影响因子，试图解释已经得到的历史价格走势。由于影响因子的种类太多，在这里选取以下有代表性的影响因子进行筛选[2]：

1.Average True Range—ATR

真实波动幅度均值，ATR 指标主要是用来衡量市场波动的强烈度，是一个为了实现市场变化率的指标

2.Stochastic Momentum Index—SMI

随机动荡指标，用现时段的收盘价和默认周期内最低价做比较，用来衡量市场动量的大小

3.Average Directional Movement Index—ADX

平均方向性指标，是一个振荡指标，用于指示市场趋势的强弱程度。

4.Aroon indicator—Aroon

阿隆指标，该指标是通过计算自价格达到近期最高值和最低值以来所经过的期间数，用于预测价格趋势到趋势区域（或者反过来，从趋势区域到趋势）的变化。

5.Bollinger Bands—BB

布林线指标，利用统计原理，求出股价的标准差及其信赖区间，从而确定股价的波动范围及未来走势，利用波带显示股价的安全高低价位。

6.Chaikin Volatility—Chaikinvol

[2] 这些因子都已实现在 R 语言的 “TTR” (Technical Trading Rules) 包中

佳庆指标，用以分析市场的内在动能

7.Close Location Value—CLV

收盘价定位值，计算收盘价处于一个时期价格变动范围中的位置的指标

8.Ease of Movement Value—EMV

简易波动指标，根据成交量和人气的变化，指示投资者在人气聚集且成交热络的时候买进股票，并且在成交量逐渐展现无力，而狂热的投资者尚未察觉能量即将用尽时，卖出股票。

9.MACD oscillator—MACD

指数平滑移动平均线，利用收盘价的短期指数移动平均线与长期指数移动平均线之间的聚合与分离状况，对买进、卖出时机作出研判的技术指标

10.Money Flow Index—MFI

资金流量指标，是某一时间周期内上涨的成交量之和与下跌的成交量之和的比率，反映市场的运行趋势。

11.Parabolic Stop-and-Reverse—SAR

抛物线指标或者停损转向操作点指标，对证券价格振幅及时间的分析研究，随时设立停损点以观察股票卖出时机的一种技术指标

12.Volatility—Volat

方差，最简单的反映市场波动性的指标。

13.CMO

钱德动量摆动指标，为寻找极度超买和极度超卖的条件。

14.RSI

相对强弱指标，适用于短线投资，应用在股价升跌的测量分析中

三、 环境说明

笔记本型号：MacBook Pro (15-inch, 2016)

操作系统：macOS Sierra 10.12.5 (16F73)

R 语言版本：version 3.3.3

四、 程序实现

载入需要的包

```
library(quantmod)
library(DMwR)
library(randomForest)#用于筛选变量的随机森林
library(xts)#时间序列对象类
library(nnet)#神经网络
library(e1071)#支持向量机
library(earth)#多元自适应回归
```

获取数据源：直接使用 quantmod 包中的 getSymbols 函数，该函数默认接口为雅虎，可以直接获得标普 500 的历史数据，设定开始日期为 1970 年之后：

```
getSymbols("^GSPC",from = "1970-01-01")
```

提示数据中存在缺失值，

```
[1] "GSPC"
Warning message:
^GSPC contains missing values. Some functions will not work if objects contain miss
ing values in the middle of the series. Consider using na.omit(), na.approx(), na.f
ill(), etc to remove or replace them.
```

利用 summary()函数查看：

```
> summary(GSPC)
     Index              GSPC.Open        GSPC.High         GSPC.Low
 Min.   :1970-01-02   Min.   :  62.28   Min.   :  63.23   Min.   :  60.96
 1st Qu.:1981-11-02   1st Qu.: 126.11   1st Qu.: 127.76   1st Qu.: 124.97
 Median :1993-08-30   Median : 449.58   Median : 451.13   Median : 447.81
 Mean   :1993-09-05   Mean   : 712.24   Mean   : 716.63   Mean   : 707.61
 3rd Qu.:2005-07-13   3rd Qu.:1210.96   3rd Qu.:1219.54   3rd Qu.:1204.41
 Max.   :2017-05-30   Max.   :2414.50   Max.   :2418.71   Max.   :2412.20
                      NA's   :1         NA's   :1         NA's   :1
   GSPC.Close       GSPC.Volume        GSPC.Adjusted
 Min.   :  62.28   Min.   :6.650e+06   Min.   :  62.28
 1st Qu.: 126.28   1st Qu.:5.132e+07   1st Qu.: 126.28
 Median : 449.60   Median :2.604e+08   Median : 449.60
 Mean   : 712.43   Mean   :1.217e+09   Mean   : 712.43
 3rd Qu.:1211.83   3rd Qu.:1.872e+09   3rd Qu.:1211.83
 Max.   :2415.82   Max.   :1.146e+10   Max.   :2415.82
 NA's   :1         NA's   :1           NA's   :1
```

发现每一栏均存在一个缺失值，推测可能是具体某一天没有交易活动，具体查看：

```
> which(is.na(GSPC))
[1] 11731 23692 35653 47614 59575 71536
> GSPC[11731]
           Open High Low Close Volume Adjusted
2016-06-29   NA   NA  NA    NA     NA       NA
```

发现是 2016-06-29 这一天没有交易数据，删除这一天的缺失值即可，删除

后再检查数据：

```
> GSPC = na.omit(GSPC)
> summary(GSPC)
     Index              GSPC.Open        GSPC.High         GSPC.Low
 Min.   :1970-01-02   Min.   :  62.28   Min.   :  63.23   Min.   :  60.96
 1st Qu.:1981-11-01   1st Qu.: 126.11   1st Qu.: 127.76   1st Qu.: 124.97
 Median :1993-08-28   Median : 449.58   Median : 451.13   Median : 447.81
 Mean   :1993-09-05   Mean   : 712.24   Mean   : 716.63   Mean   : 707.61
 3rd Qu.:2005-07-12   3rd Qu.:1210.96   3rd Qu.:1219.54   3rd Qu.:1204.41
 Max.   :2017-05-30   Max.   :2414.50   Max.   :2418.71   Max.   :2412.20
   GSPC.Close       GSPC.Volume         GSPC.Adjusted
 Min.   :  62.28   Min.   :6.650e+06   Min.   :  62.28
 1st Qu.: 126.28   1st Qu.:5.132e+07   1st Qu.: 126.28
 Median : 449.60   Median :2.604e+08   Median : 449.60
 Mean   : 712.43   Mean   :1.217e+09   Mean   : 712.43
 3rd Qu.:1211.83   3rd Qu.:1.872e+09   3rd Qu.:1211.83
 Max.   :2415.82   Max.   :1.146e+10   Max.   :2415.82
```

数据正常，未出现缺失值，接下来修改数据中的变量名称：

```
colnames(GSPC) = c("Open","High","Low","Close","Volume","Adjusted")
```

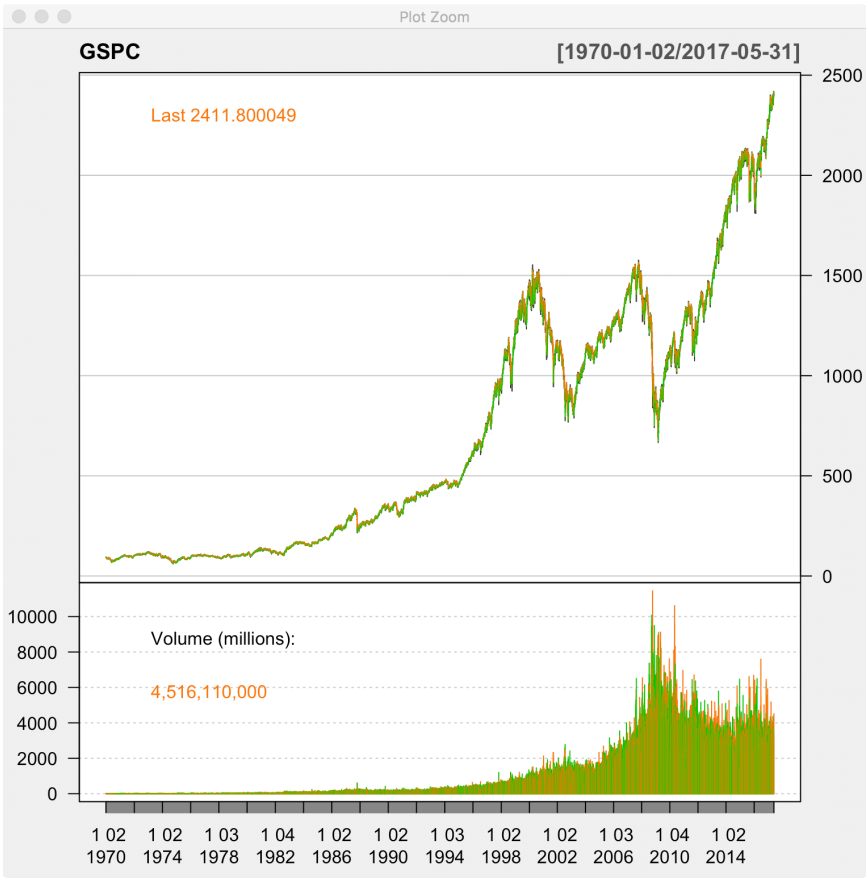
查看成功修改过的数据

Filter

	Open	High	Low	Close	Volume	AdjClose
1970-01-02	92.06	93.54	91.79	93.00	8050000	93.00
1970-01-05	93.00	94.25	92.53	93.46	11490000	93.46
1970-01-06	93.46	93.81	92.13	92.82	11460000	92.82
1970-01-07	92.82	93.38	91.93	92.63	10010000	92.63
1970-01-08	92.63	93.47	91.99	92.68	10670000	92.68
1970-01-09	92.68	93.25	91.82	92.40	9380000	92.40
1970-01-12	92.40	92.67	91.20	91.70	8900000	91.70
1970-01-13	91.70	92.61	90.99	91.92	9870000	91.92
1970-01-14	91.92	92.40	90.88	91.65	10380000	91.65
1970-01-15	91.65	92.35	90.73	91.68	11120000	91.68
1970-01-16	91.68	92.49	90.36	90.92	11940000	90.92
1970-01-19	90.72	90.72	89.14	89.65	9500000	89.65
1970-01-20	89.65	90.45	88.64	89.83	11050000	89.83
1970-01-21	89.83	90.61	89.20	89.95	9880000	89.95

Showing 1 to 14 of 11,962 entries

简单利用 chartSeries(GSPC, theme = ‘white’)查看标准普尔 1970 年至今的走势：



接下来用函数实现之前定义的集合 T，也就是本次分析的目标变量：

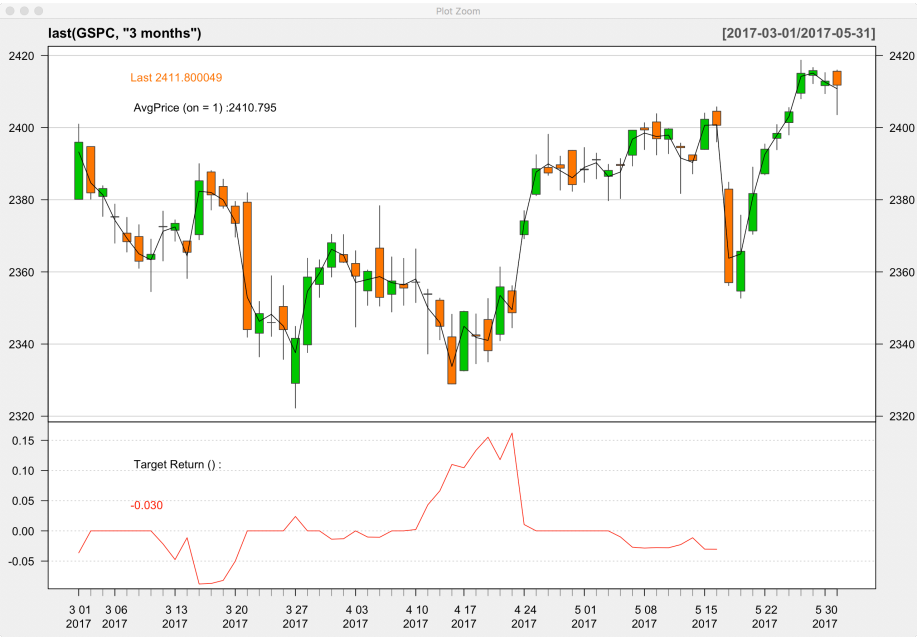
```
T.ind = function(quotes, tgt.margin = 0.01, n.days = 10){
  v = apply(HLC(quotes),1,mean)#逐行求平均数，得到每日均价
  r = matrix(NA, ncol = n.days, nrow = NROW(quotes))
  #生成一个和数据集行数一样，记录接下来十天波动情况的矩阵
  for (x in 1:n.days) r[,x] = Next(Delt(v, k = x),x)#计算波动
  x = apply(r, 1, function(x) sum(x[x> tgt.margin | x< -tgt.margin]))
  #逐行求和
  if (is.xts(quotes))
    xts(x, time(quotes))#将原矩阵的时间标签转移到函数生存矩阵中
  else x
}
```

在这个函数中，设置目标波动为 0.01，也就是假设交易的手续费为每笔 1%，所以股价波动超过 1%就可以认为是有利可图的。不过，实际上这一手续费是不统一的，不同的券商有不同的收取手续费的标准，在后期可以修改这一参数。

接下来尝试使用我们所定义的指标，查看过去 3 个月的情况，检测是否能起到指示的作用：

```
candleChart(last(GSPC,"3 months"), theme = 'white', TA = NULL)
avgPrice = function(x) apply(HLC(x),1,mean)
addAvgPrice = newTA(FUN = avgPrice, col = 1, legend = "AvgPrice")
addT.ind = newTA(FUN = T.ind, col = 'red', legend = "Target Return")
addAvgPrice(on=1)
addT.ind()
```

得到下图：



上图中，上半部分的图片展示的是 S&P500 过去三个月的价格走势，而下半部分是定义的目标变量 T 的走势，可以看到，在未来十天内若价格与当前价格相差过大，相应地 T 的数值也会越高，所以说，T 能够指示未来时间内的价格变化。

接下来，将上文提到的变量加入到模型中：

```
#预处理TTR包中的函数
myATR = function(x) ATR(HLC(x)) [,"atr"]
mySMI = function(x) SMI(HLC(x)) [,"SMI"]
myADX = function(x) ADX(HLC(x)) [,"ADX"]
myAroon = function(x) aroon(x[,c("High","Low")])$oscillator
myBB = function(x) BBands(HLC(x)) [,"pctB"]
myChaikinVol = function(x) Delt(chaikinVolatility(x[,c("High","Low")]))[,1]
myCLV = function(x) EMA(CLV(HLC(x))) [,1]
myEMV = function(x) EMV(x[,c("High","Low", "Close")],x[,"Volume"]) [,2]
myMACD <- function(x) MACD(Cl(x))[, 2]
myMFI = function(x) MFI(x[,c("High","Low","Close")], x[,"Volume"])
mySAR <- function(x) SAR(x[, c("High", "Close")])[, 1]
myVolat = function(x) volatility(OHLC(x), calc = "garman") [,1]
```

建立起以 T 为目标变量，上文提到的指标为输入变量的预测模型

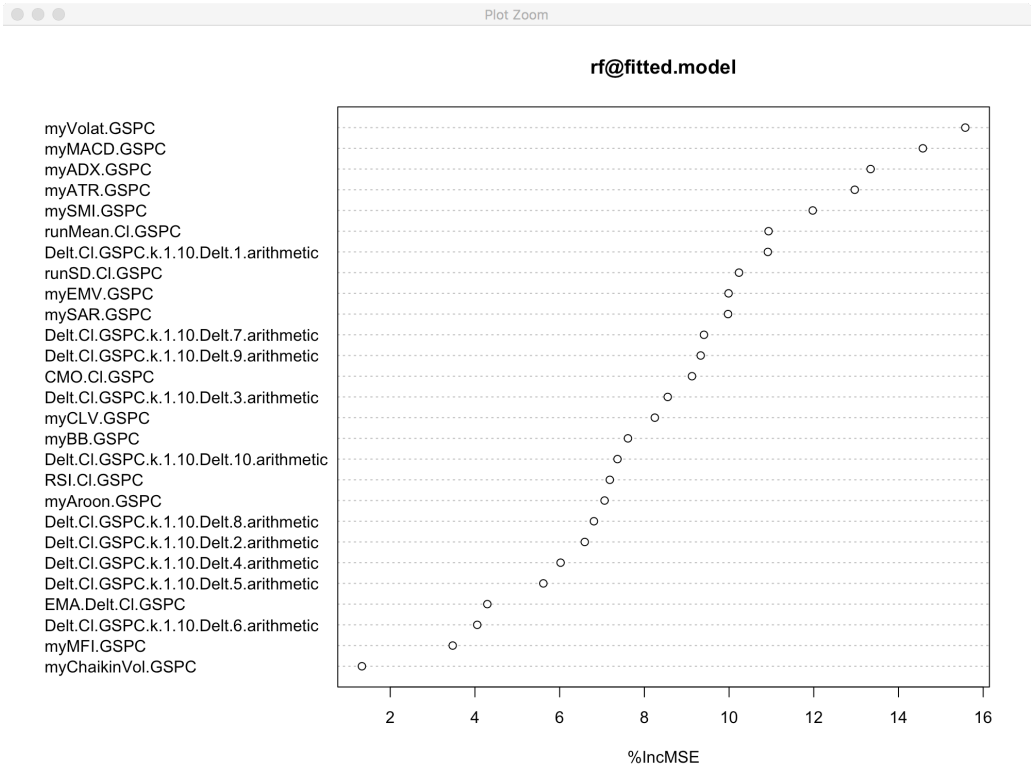
```
#建立模型
MGSPC = na.omit(GSPC)#忽略缺失值的的GSPC数据集，
data.model = specifyModel(T.ind(GSPC) ~ Delt(Cl(GSPC),k=1:10)+myATR(GSPC)
                          + mySMI(GSPC) + myADX(GSPC) ++ myAroon(GSPC)
                          + myBB(GSPC) + myChaikinVol(GSPC) + myCLV(GSPC)
                          + CMO(Cl(GSPC)) + EMA(Delt(Cl(MGSPC)))
                          + myEMV(na.omit(GSPC)) + myVolat(GSPC)
                          + myMACD(GSPC)+ myMFI(GSPC) + RSI(Cl(GSPC))
                          + mySAR(GSPC) + runMean(Cl(GSPC))
                          + runSD(Cl(GSPC))
)
```

再利用随机森林进行变量的筛选，并显示显著性大于 10 的变量：

```
#设置随机种子，用随机森林筛选变量
rf = buildModel(data.model, method = 'randomForest',
                training.per = c(start(GSPC), index(GSPC["20081231"])),
                ntree = 50, importance = T)
varImpPlot(rf@fitted.model, type = 1)
imp = importance(rf@fitted.model, type = 1)

rownames(imp)[which(imp>10)]
```

下图展示的是变量重要性的分布图：



其中大于 10 的是：

```
> rownames(imp)[which(imp>10)]
[1] "Delt.Cl.GSPC.k.1.10.Delt.1.arithmetic"
[2] "myATR.GSPC"
[3] "mySMI.GSPC"
[4] "myADX.GSPC"
[5] "myVolat.GSPC"
[6] "myMACD.GSPC"
[7] "runMean.Cl.GSPC"
[8] "runSD.Cl.GSPC"
```

所以将这些变量纳入模型中，同时将数据分区，2008 年 12 月 31 日之前的数据用于训练，之后的数据用于评估。之所以这样分，是因为 2008 年金融危机带来影响较大，正好可以作为数据的分割点：

```
data.model = specifyModel(T.ind(GSPC) ~ Delt(Cl(GSPC), k=1)
                          + myATR(GSPC) + mySMI(GSPC) + myADX(GSPC)
                          + myVolat(GSPC) + myMACD(GSPC) + runMean(Cl(GSPC))
                          + runSD(Cl(GSPC))
                          )
Tdata.train = as.data.frame(modelData(data.model,
                            data.window = c('1970-01-02','2008-12-31')))
Tdata.eval = na.omit(as.data.frame(modelData(data.model,
                            data.window = c('2009-01-01','2017-05-30'))))
Tform = as.formula(paste("T.ind.GSPC ~ ",
                         paste(colnames(Tdata.train)[2:8], collapse = "+")))
```

接下来使用神经网络模型：

```
norm.data = scale(Tdata.train)#数据规整
nn = nnet(Tform, norm.data[1:5000,],size=10, decay =0.01,
          maxit = 10000, linout = T, trace = F)
norm.preds = predict(nn, norm.data[5001:9805,])
preds = unscale(norm.preds,norm.data)#反规整
sigs.nn=trading.signals(preds, 0.1, -0.1)
true.sigs = trading.signals(Tdata.train[5001:9805,"T.ind.GSPC"],0.1,-0.1)
sigs.PR(sigs.nn, true.sigs)#查看模型精确度与召回率
```

支持向量机：

```
sv = svm(Tform, Tdata.train[1:5000, ], gama = 0.001, cost = 100)
s.preds = predict(sv, Tdata.train[5001:9805, ])
sigs.svm = trading.signals(s.preds, 0.1, -0.1)
true.sigs = trading.signals(Tdata.train[5001:9805,"T.ind.GSPC"],0.1,-0.1)
sigs.PR(sigs.svm, true.sigs)
```

多远自适应回归：

```
#多远自适应回归
e = earth(Tform, Tdata.train[1:5000, ])
e.preds = predict(e, Tdata.train[5001:9805,])
sigs.e = trading.signals(e.preds, 0.1,-0.1)
true.sigs = trading.signals(Tdata.train[5001:9805, "T.ind.GSPC"],0.1,-0.1)
sigs.PR(sigs.e, true.sigs)
```

在模型选择上，可以使用蒙特卡洛方法，但是计算量巨大，对设备要求极高，此处直接使用 DMwR 包中的蒙特卡洛方法源码，对上面建立的 nnet、svm、mar 模型进行评估选择。

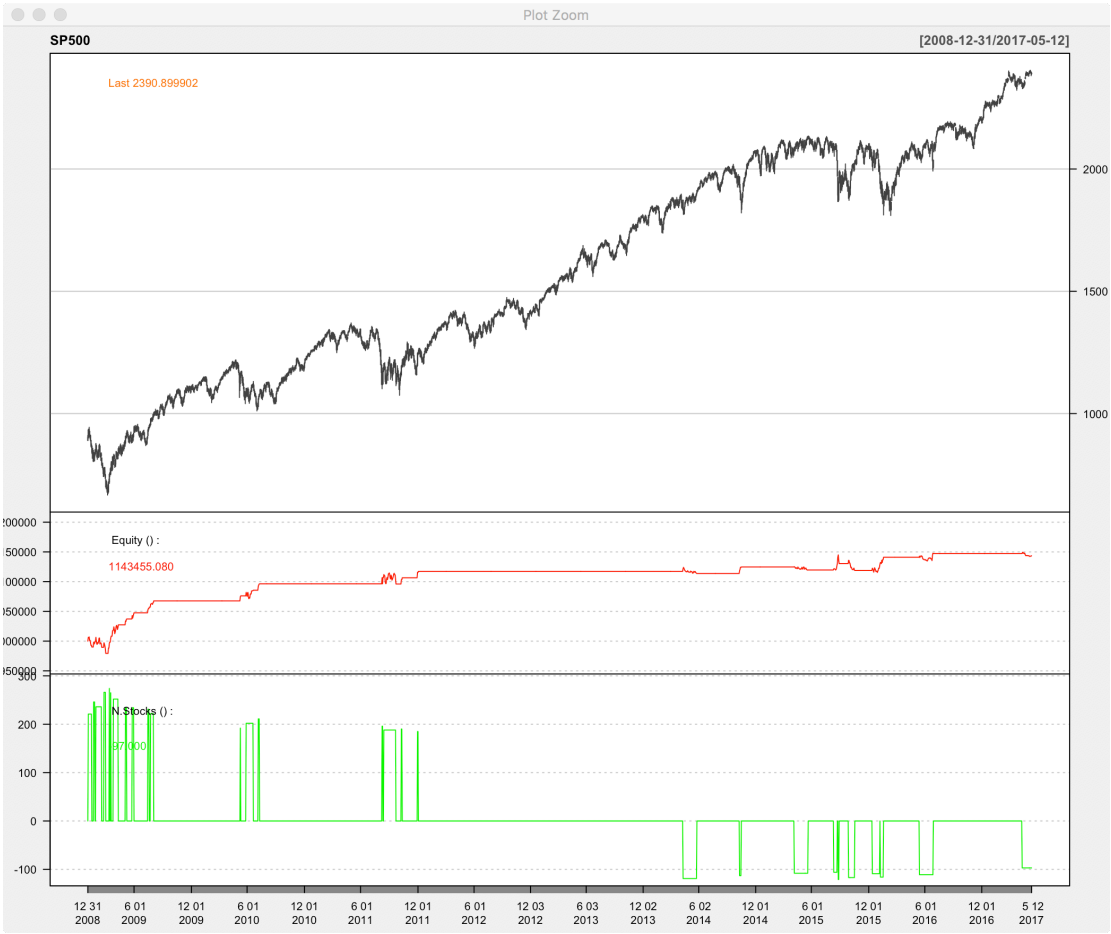
```
MC.svmR <- function(form,train,test,b.t=0.1,s.t=-0.1,...) {
  require(e1071)
  t <- svm(form,train,...)
  p <- predict(t,test)
  trading.signals(p,b.t,s.t)
}
MC.svmC <- function(form,train,test,b.t=0.1,s.t=-0.1,...) {
  require(e1071)
  tgtName <- all.vars(form)[1]
  train[,tgtName] <- trading.signals(train[,tgtName],b.t,s.t)
  t <- svm(form,train,...)
  p <- predict(t,test)
  factor(p,levels=c('s','h','b'))
}
MC.nnetR <- function(form,train,test,b.t=0.1,s.t=-0.1,...) {
  require(nnet)
  t <- nnet(form,train,...)
  p <- predict(t,test)
  trading.signals(p,b.t,s.t)
}
MC.nnetC <- function(form,train,test,b.t=0.1,s.t=-0.1,...) {
  require(nnet)
  tgtName <- all.vars(form)[1]
  train[,tgtName] <- trading.signals(train[,tgtName],b.t,s.t)
  t <- nnet(form,train,...)
  p <- predict(t,test,type='class')
  factor(p,levels=c('s','h','b'))
}
MC.earth <- function(form,train,test,b.t=0.1,s.t=-0.1,...) {
  require(earth)
  t <- earth(form,train,...)
  p <- predict(t,test)
  trading.signals(p,b.t,s.t)
}
```

…（详细代码参看源代码文件）

五、 主要结论

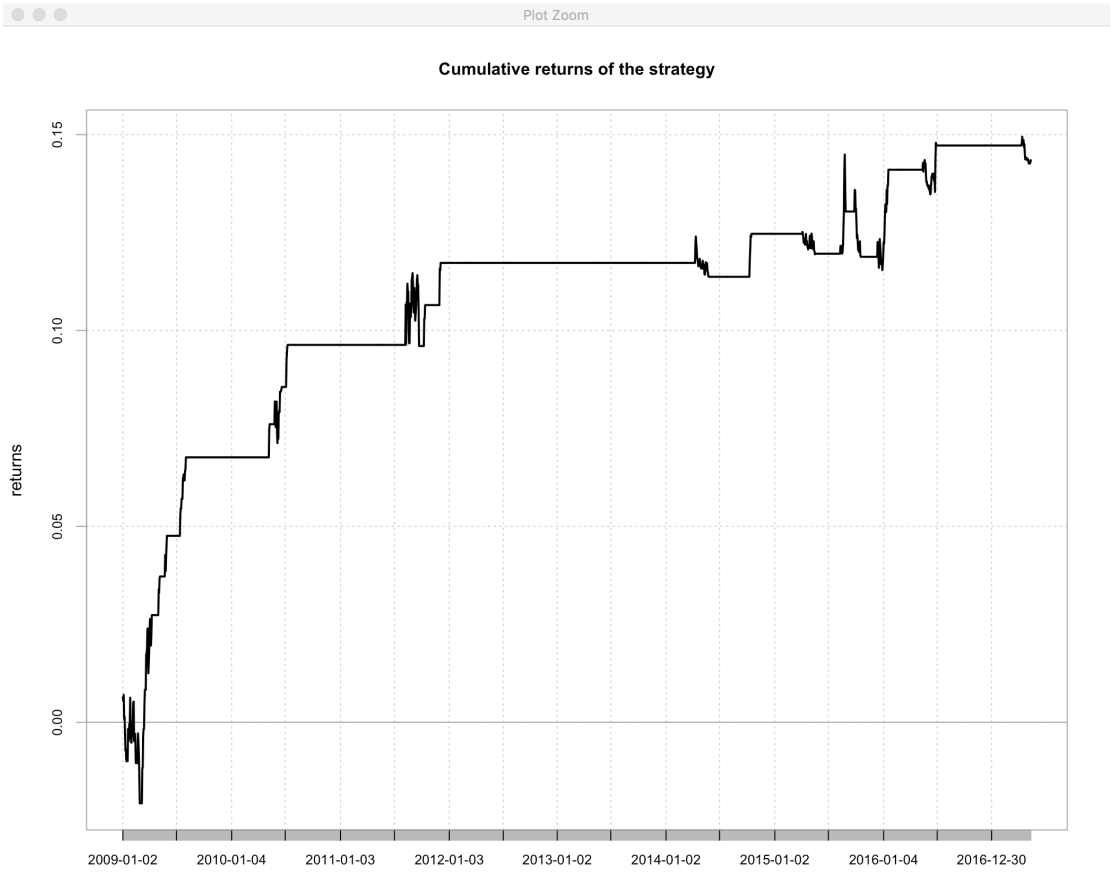
利用上文所述的方法（不包含蒙特卡洛估计筛选最优模型），基于 2008 年 12

月 31 日后的数据进行测试，得到的结果如下：




```
> tradingEvaluation(t1)
    NTrades       NProf    PercProf          PL         Ret    RetOverBH
      31.00       22.00       70.97   143455.08       14.35      -150.35
      MaxDD SharpeRatio     AvgProf     AvgLoss       AvgPL     MaxProf
   29511.39        0.05        4.75       -3.70        2.30        5.42
     MaxLoss
      -5.17
```

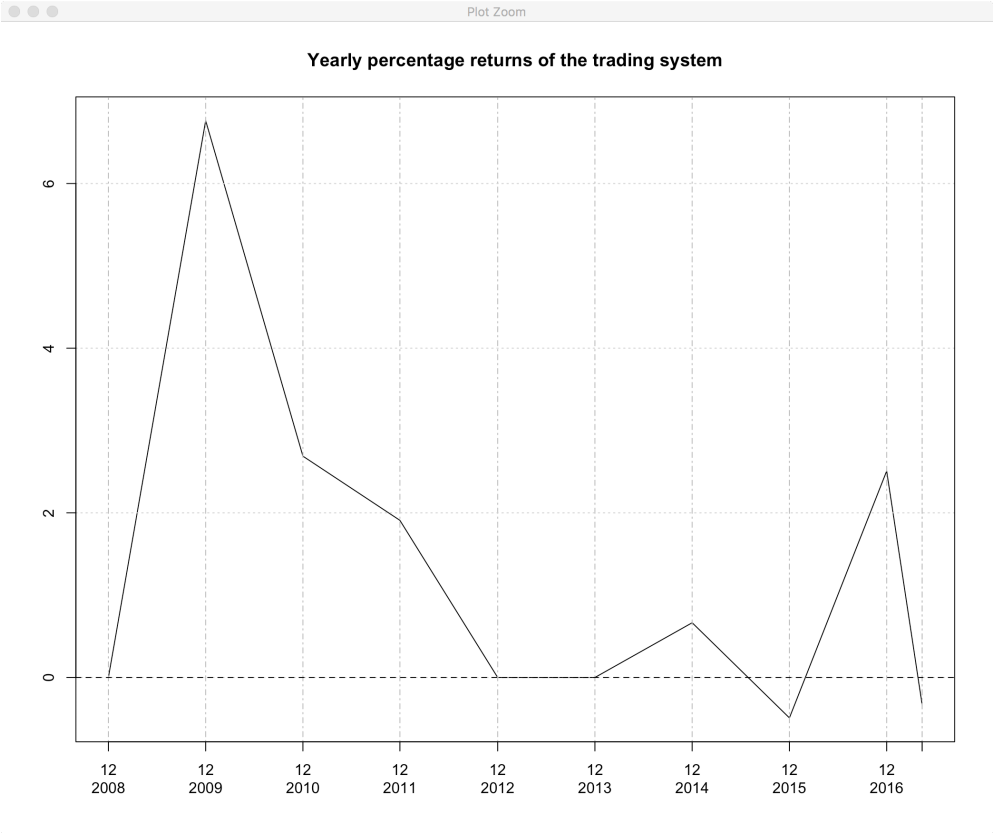
从上图中可以看到，自 08 年金融危机以来，S&P500 总体走势上涨，而使用该策略我们只实现了 14.35%的回报率，并不是令人满意的结果，再看总体回报率的走势：



可以看到，除了在 2009 年第一季度出现亏损之外，整体的收益率均在零以上浮动，并且整体呈现上升趋势，在 2011 年末突破 10%。到了近期，回报率不断再创新高的同时波动也较大，紧逼 15%。

```
> yearlyReturn(t1TradeEquity)
           yearly.returns
2008-12-31    0.000000000
2009-12-31    0.067622421
2010-12-31    0.026884017
2011-12-30    0.019089044
2012-12-31    0.000000000
2013-12-31    0.000000000
2014-12-31    0.006649910
2015-12-31   -0.004908204
2016-12-30    0.025069815
2017-05-12   -0.003280752
```

查看每一年的回报率，从曲线中可以发现较高的收益率出现在 2008 年金融危机之后，2009 年最高，而那时金融危机带来的影响尚未结束，市场仍然不是很景气。这在一定的程度上说明我们建立的模型在市场行情相对低迷的情况下取得的效果较好。



总体而言，通过数据挖掘所得到这一预测股票回报的模型，虽然回报率不尽如人意，但至少能够跑赢一些券商机构推出的理财产品，以蚂蚁金服的余额宝为例，其年化回报率常在 1.5%左右浮动，而凭借我们的策略最高单年能取得 7%左右的回报率。

不过，这一模型不适用于我中国股票市场，因为在定义的交易策略中存在卖空的情况，可是目前的中国股市是不允许卖空的，所以这一模型仅适用于存在卖空机制的市场。同时，这一模型仍然还有很多优化的空间，上文提到的蒙特卡洛方法，此处并没有使用到，如果利用该随机模拟方法，可以更为严谨的设置模型的参数，优化模型，从而使模型达到更高的精确度和召回率，以取得更为可观的回报率。

参考文献：

[1] Chapman, Data Mining with R[M],

[2]李体委. Fama-French 三因子模型的改进和对中国股市收益率的检验[D].山东大学,2011.