分类问题

王宁宁

2015 年 11 月 27 日

需要安装的包：ISLR MASS class e1071 party nnet

```
rm(list = ls(all = TRUE))
options(digits = 4  )
```

数据来自标普 500 在 2001 年和 2005 年的 1250 个观测。

目的：利用前四天的涨跌情况来预测明天的涨跌。

模型评估：预测的准确率

```
library(ISLR)
names(Smarket)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag
5"
## [7] "Volume"    "Today"     "Direction"
```

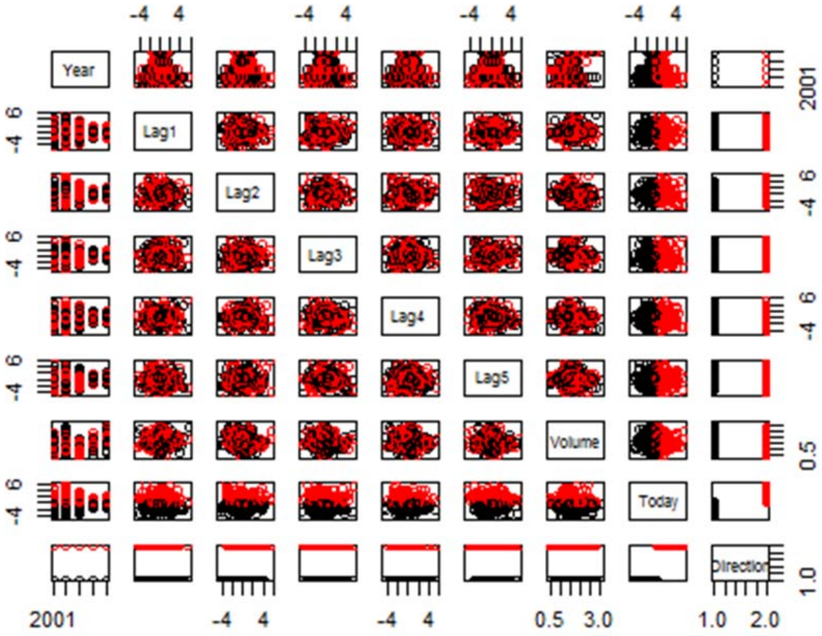
```
head(Smarket)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5 Volume  Today Direction
## 1 2001  0.381 -0.192 -2.624 -1.055  5.010  1.191  0.959        Up
## 2 2001  0.959  0.381 -0.192 -2.624 -1.055  1.296  1.032        Up
## 3 2001  1.032  0.959  0.381 -0.192 -2.624  1.411 -0.623      Down
## 4 2001 -0.623  1.032  0.959  0.381 -0.192  1.276  0.614        Up
## 5 2001  0.614 -0.623  1.032  0.959  0.381  1.206  0.213        Up
## 6 2001  0.213  0.614 -0.623  1.032  0.959  1.349  1.392        Up
```

```
summary(Smarket)
```

```
##       Year           Lag1             Lag2             Lag3       
##  Min.   :2001   Min.   :-4.922   Min.   :-4.922   Min.   :-4.922  
##  1st Qu.:2002   1st Qu.:-0.640   1st Qu.:-0.640   1st Qu.:-0.640  
##  Median :2003   Median : 0.039   Median : 0.039   Median : 0.038  
##  Mean   :2003   Mean   : 0.004   Mean   : 0.004   Mean   : 0.002  
##  3rd Qu.:2004   3rd Qu.: 0.597   3rd Qu.: 0.597   3rd Qu.: 0.597  
##  Max.   :2005   Max.   : 5.733   Max.   : 5.733   Max.   : 5.733  
##       Lag4             Lag5            Volume          Today       
##  Min.   :-4.922   Min.   :-4.922   Min.   :0.356   Min.   :-4.922  
##  1st Qu.:-0.640   1st Qu.:-0.640   1st Qu.:1.257   1st Qu.:-0.640  
##  Median : 0.038   Median : 0.038   Median :1.423   Median : 0.038  
##  Mean   : 0.002   Mean   : 0.006   Mean   :1.478   Mean   : 0.003  
##  3rd Qu.: 0.597   3rd Qu.: 0.597   3rd Qu.:1.642   3rd Qu.: 0.597  
##  Max.   : 5.733   Max.   : 5.733   Max.   :3.152   Max.   : 5.733  
##  Direction 
##  Down:602  
##  Up  :648  
##            
##            
##            
## 
```

```
#?Smarket
pairs(Smarket,col=Smarket$Direction)
```



Logistic regression

利用 Logistic regression

模型拟合：

```
glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Smarket,fam
ily=binomial)
summary(glm.fit)
```

```
## 
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + 
##     Volume, family = binomial, data = Smarket)
## 
```

```
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -1.45   -1.20    1.07    1.15    1.33
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.12600    0.24074   -0.52     0.60
## Lag1        -0.07307    0.05017   -1.46     0.15
## Lag2        -0.04230    0.05009   -0.84     0.40
## Lag3         0.01109    0.04994    0.22     0.82
## Lag4         0.00936    0.04997    0.19     0.85
## Lag5         0.01031    0.04951    0.21     0.83
## Volume       0.13544    0.15836    0.86     0.39
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.2  on 1249  degrees of freedom
## Residual deviance: 1727.6  on 1243  degrees of freedom
## AIC: 1742
##
## Number of Fisher Scoring iterations: 3
```

```
glm.probs=predict(glm.fit,type="response")
glm.probs[1:5]
```

```
##      1      2      3      4      5
## 0.5071 0.4815 0.4811 0.5152 0.5108
```

```
glm.pred=ifelse(glm.probs>0.5,"Up","Down")
attach(Smarket)
table(glm.pred,Direction)
```

```
##         Direction
## glm.pred Down  Up
##     Down  145 141
##     Up    457 507
```

```
mean(glm.pred==Direction)
```

```
## [1] 0.5216
```

预测精度为 0.52

现在我们用 2005 年以前的数据作为训练集合，把 2005 年的数据作为测试集合：

```
train = Year<2005
glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
            data=Smarket,family=binomial, subset=train)
glm.probs=predict(glm.fit,newdata=Smarket[!train,],type="response")
glm.pred=ifelse(glm.probs >0.5,"Up","Down")
Direction.2005=Smarket$Direction[!train]
table(glm.pred,Direction.2005)
```

```
##         Direction.2005
## glm.pred Down Up
##     Down   77 97
##     Up     34 44
```

```
mean(glm.pred==Direction.2005)
```

```
## [1] 0.4802
```

用 2005 年以前的数据，建立 logistic 模型来预测 2005 年的数据，精度为 0.48

下面考虑只用最近两天来预测

```
glm.fit=glm(Direction~Lag1+Lag2,
            data=Smarket,family=binomial, subset=train)
glm.probs=predict(glm.fit,newdata=Smarket[!train,],type="response")
glm.pred=ifelse(glm.probs >0.5,"Up","Down")
table(glm.pred,Direction.2005)
```

```
##         Direction.2005
## glm.pred Down  Up
##     Down   35  35
##     Up     76 106
```

```
mean(glm.pred==Direction.2005)
```

```
## [1] 0.5595
```

```
106/(76+106)
```

```
## [1] 0.5824
```

```
(35+106)/length(Direction.2005)
```

```
## [1] 0.5595
```

精度为 0.56

线性判别分析

下面考虑使用线性判别分析：（使用最近两天来预测）

```
library(MASS)
## Linear Discriminant Analysis
lda.fit=lda(Direction~Lag1+Lag2,data=Smarket, subset=Year<2005)
lda.fit
```

```
## Call:
## lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = Year <
##     2005)
##
## Prior probabilities of groups:
##  Down    Up
## 0.492 0.508
##
## Group means:
##          Lag1     Lag2
## Down  0.04279  0.03389
## Up   -0.03955 -0.03133
##
## Coefficients of linear discriminants:
##          LD1
## Lag1 -0.6420
## Lag2 -0.5135
```

```
Smarket.2005=subset(Smarket,Year==2005)
lda.pred=predict(lda.fit,Smarket.2005)
class(lda.pred)
```

```
## [1] "list"
```

```
data.frame(lda.pred)[1:5,]
```

```
##      class posterior.Down posterior.Up      LD1
## 999     Up         0.4902       0.5098  0.08293
## 1000    Up         0.4792       0.5208  0.59114
## 1001    Up         0.4668       0.5332  1.16723
## 1002    Up         0.4740       0.5260  0.83335
## 1003    Up         0.4928       0.5072 -0.03793
```

```
table(lda.pred$class,Smarket.2005$Direction)
```

```
##
##         Down  Up
##   Down    35  35
##   Up      76 106
```

```
mean(lda.pred$class==Direction.2005)
```

```
## [1] 0.5595
```

精度为 0.56

二次判别函数

```
qda.fit=qda(Direction~Lag1+Lag2 ,data=Smarket ,subset =train)
qda.fit
```

```
## Call:
## qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
##
## Prior probabilities of groups:
##  Down    Up
## 0.492 0.508
##
## Group means:
```

```
##          Lag1     Lag2
## Down  0.04279  0.03389
## Up   -0.03955 -0.03133
```

```
qda.class =predict(qda.fit,Smarket.2005)$class
table(qda.class,Direction.2005)
```

```
##          Direction.2005
## qda.class Down  Up
##      Down   30  20
##      Up     81 121
```

```
mean(qda.class == Direction.2005)
```

```
## [1] 0.5992
```

精度为 0.6

KNN 算法

```
library(class)
#?knn
attach(Smarket)
```

```
## The following objects are masked from Smarket (pos = 5):
##
##     Direction, Lag1, Lag2, Lag3, Lag4, Lag5, Today, Volume, Year
```

```
train.X=cbind(Lag1 ,Lag2)[train ,]
test.X=cbind (Lag1 ,Lag2)[!train ,]
Xlag=cbind(Lag1,Lag2)
train.Direction =Direction[train]
train=Year<2005
knn.pred=knn(Xlag[train,],Xlag[!train,],Direction[train],k=2)
table(knn.pred,Direction[!train])
```

```
##
## knn.pred Down Up
##     Down   48 56
##     Up     63 85
```

```
mean(knn.pred==Direction[!train])
```

```
## [1] 0.5278
```

```
knn.pred2=knn(train.X,test.X,train.Direction,k=3)
table(knn.pred2,Direction[!train])
```

```
##
## knn.pred2 Down Up
##      Down   48 55
##      Up     63 86
```

```
mean(knn.pred2==Direction.2005)

## [1] 0.5317
```

0.54

支持向量机（svm）

考虑一种径向基函数核的支持向量机：

```
library("e1071")
#library(ISLR)
train = Year<2005
model <- svm(Direction~Lag1+Lag2, data = Smarket[train,], method = "C-c
lassification", kernel = "radial",cost = 10, gamma = 0.1)
pred <- predict(model,Smarket[!train,])
Direction.2005=Smarket$Direction[!train]
table(Direction.2005)

## Direction.2005
## Down   Up
##  111  141

table(pred)

## pred
## Down   Up
##   27  225

table(pred,Direction.2005)

##       Direction.2005
## pred   Down  Up
##   Down   18   9
##   Up     93 132

mean(pred==Direction.2005)

## [1] 0.5952
```

精度为 0.6

人工神经网络

使用一个中间层三个节点的神经网络

```
library(nnet)
set.seed(123)
model_nnet <- nnet(Direction~Lag1+Lag2, data = Smarket[train,],size =3,
 rang = 0.3,decay = 5e-4, maxit = 200)

## # weights:  13
## initial  value 693.467865
## iter  10 value 687.840415
## iter  20 value 683.220238
## iter  30 value 682.670126
## iter  40 value 682.510102
## iter  50 value 682.421389
## final  value 682.418903
## converged
```

```
summary(model_nnet)
```

```
## a 2-3-1 network with 13 weights
## options were - entropy fitting  decay=5e-04
##  b->h1 i1->h1 i2->h1
##   1.74   0.11  -4.14
##  b->h2 i1->h2 i2->h2
##  -1.20   3.20   0.23
##  b->h3 i1->h3 i2->h3
##   1.18  -2.69  -0.49
##  b->o h1->o h2->o h3->o
##  4.47  0.81 -4.91 -4.90
```

```
pred <- predict(model_nnet,Smarket[!train,])
prednn <- ifelse(pred>0.5,"Up","Down")
table(prednn)
```

```
## prednn
## Down   Up
##  100  152
```

```
table(Direction.2005)
```

```
## Direction.2005
## Down   Up
##  111  141
```

```
table(prednn,Direction.2005)
```

```
##       Direction.2005
## prednn Down Up
##   Down   47 53
##   Up     64 88
```

```
mean(prednn==Direction.2005)
```

```
## [1] 0.5357
```

精度为 0.54