计算统计 HW5

朱强强 17064001

2020 年 5 月 20 日

1. 使用以上模型判断测试图片是否是 3。

```
> setwd("E:/Courses/Computational Statistics/Homework/HW5")
>
> azip <- as.matrix(read.table("azip.csv"))
> dzip <- as.matrix(read.table("dzip.csv"))
>
> testzip <- as.matrix(read.table("testzip.csv"))
> dtest <- as.matrix(read.table("dtest.csv"))
>
> dim(azip)
> dim(dzip)
> dim(testzip)
> dim(dtest)
```

```
[1] 1707  256
[1] 1707    1
[1] 2007  256
[1] 2007    1
```

训练数据集分成两个部分：

- azip：1707 张灰度图，被记录成 256×1707 的矩阵，每张图片被纵向拉伸成一个 256 维的向量。
- dzip：1707 张灰度图对应的数字（0-9），被记录成 1×1707 的向量。

我们的目标是利用以上 1707 张图（称为训练数据）预测新图片是否为数字 3，并且把预测结果和真实结果比较。测试数据分为以下部分：

- testzip：2007 张灰度图，256×2007 的矩阵。
- dtest：2007 张灰度图对应的真实数字（0-9），1×2007 的向量。

构建主成分分析函数

```
> pcr <- function(x_train, y_train, x_test, y_test, cut, k=15) {
+   x_train <- as.matrix(x_train)
+   y_train <- as.matrix(y_train)
+   x_test <- as.matrix(x_test)
+   y_test <- as.matrix(y_test)
+
+   res_train <- svd(x_train)
+   U_train <- res_train$u
+   V_train <- res_train$v
+   d_train <- res_train$d
+   Z_train <- U_train[, 1:k] %*% diag(d_train[1:k], nrow=k)
+   train_data <- data.frame(cbind(Z_train, y_train))
+   colnames(train_data) <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7",
+                             "x8", "x9", "x10", "x11", "x12", "x13",
+                             "x14", "x15", "y")
+   m <- lm(y~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15,
+           data=train_data)
+   res_test <- svd(x_test)
+   U_test <- res_test$u
+   V_test <- res_test$v
+   d_test <- res_test$d
+   Z_test <- U_test[, 1:k] %*% diag(d_test[1:k], nrow=k)
+   test_data <- data.frame(Z_test)
+   colnames(test_data) <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7",
+                            "x8", "x9", "x10", "x11", "x12", "x13",
+                            "x14", "x15")
+
+   train_fit <- fitted.values(m, data.frame(Z_train))
+   p <- as.matrix(predict.lm(m, newdata=test_data) > cut)
+
+   n1 <- length(y_test)
+   n2 <- sum(y_test)
+   n3 <- n1 - n2
+
+   i <- 0
+   for (j in 1:length(y_test)) {
+     if (p[j,] == y_test[j,]) {
+       i <- i + 1
+     }
```

```
+   }
+   accurate1 <- i/n1
+
+   i <- 0
+   for (j in 1:length(y_test)) {
+     if (p[j,] == TRUE && y_test[j,] == TRUE) {
+       i <- i + 1
+     }
+   }
+   accurate2 <- i/n2
+
+   i <- 0
+   for (j in 1:length(y_test)) {
+     if (p[j,] == FALSE && y_test[j,] == FALSE) {
+       i <- i + 1
+     }
+   }
+   accurate3 <- i/n3
+
+   return(c(accurate1, accurate2, accurate3))
+ }
> res <- pcr(azip, dzip==3, testzip, dtest==3, cut=0.3)
> res
```

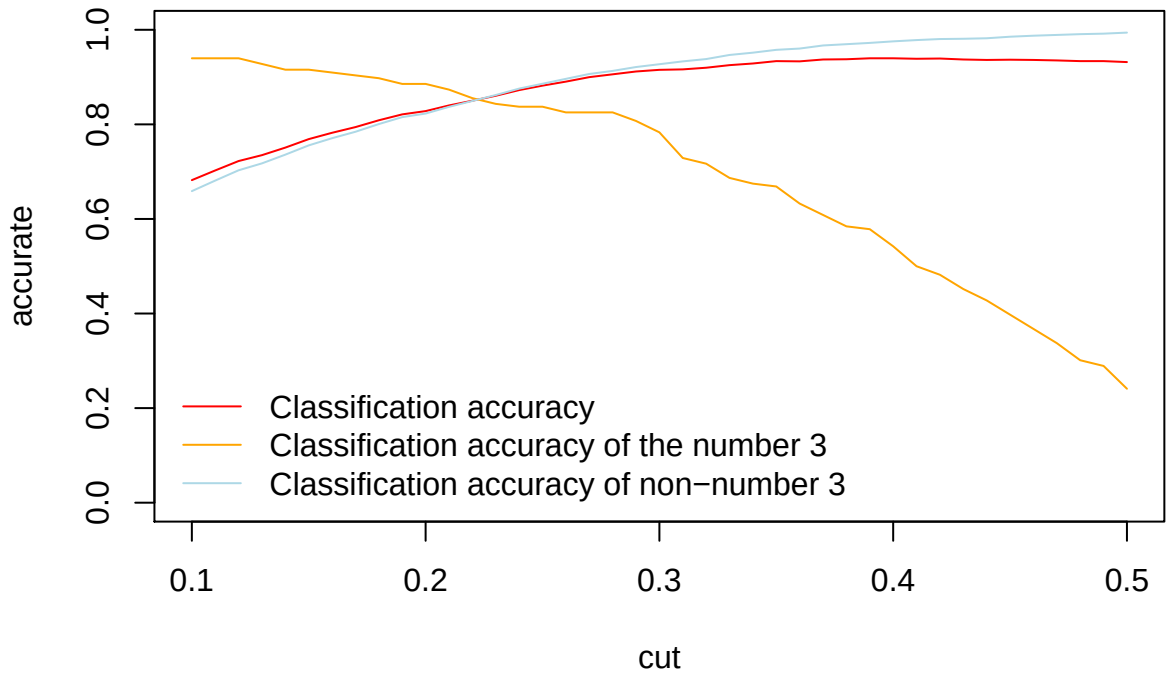
```
[1] 0.9152965 0.7831325 0.9272135
```

我们用训练集中模型拟合结果的 0.3 作为我们的分类截断点（cut）。对于测试集，分类准确度约为 91.53%, 数字 3 的分类准确度为 78.31%，非数字 3 的分类准确度为 92.72%。

2. 选取不同的分类截断点，记录分类的准确度，画一个折线图。

```
> c1 <- c()
> c2 <- c()
> c3 <- c()
> i <- 1
> for (cut in seq(0.1, 0.5, 0.01)) {
+   res.new <- pcr(azip, dzip==3, testzip, dtest==3, cut=cut)
+   c1[i] <- res.new[1]
+   c2[i] <- res.new[2]
+   c3[i] <- res.new[3]
+   i <- i + 1
```

```
+ }
>
> plot(seq(0.1, 0.5, 0.01), c1, type="l", lwd=1, xlab="cut", col="red",
+      ylab="accurate", main="", ylim=c(0, 1))
> lines(seq(0.1, 0.5, 0.01), c2, col="orange", lwd=1)
> lines(seq(0.1, 0.5, 0.01), c3, col="lightblue", lwd=1)
> legend("bottomleft", c("Classification accuracy",
+                        "Classification accuracy of the number 3",
+                        "Classification accuracy of non-number 3"),
+        col=c("red", "orange", "lightblue"), lwd=1, bty="n")
```



我们从图中可以发现随着分类间断点的增大，分类准确度和非数字 3 的分类准确度逐渐升高，而数字 3 的分类准确度会降低。当分类间断点的取值在 0.2 和 0.25 之间时，这些分类准确度会得到一个较好的平衡。

3. 从测试集中画出一部分分类不准确的手写数字 3。

plot.digit 函数包含了一个将 testzip 中的每一列转化为图片的函数：

```
> plot.digit <- function(a) {
+   z <- matrix(a, ncol=16)[, 16:1]
+   op <- par()
+   par(mar=c(0.5, 0.5, 0.5, 0.5), ann=F)
+   image(z=z, col=grey(255:0/255), main="", axes=F)
+   par(mar=op$mar, ann=op$ann)
+ }
>
> x_train <- as.matrix(azip)
> y_train <- as.matrix(dzip == 3)
> x_test <- as.matrix(testzip)
> y_test <- as.matrix(dtest == 3)
> cut <- 0.3
> k <- 15
>
> res_train <- svd(x_train)
> U_train <- res_train$u
> V_train <- res_train$v
> d_train <- res_train$d
> Z_train <- U_train[, 1:k] %*% diag(d_train[1:k], nrow=k)
> train_data <- data.frame(cbind(Z_train, y_train))
> colnames(train_data) <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7",
+                           "x8", "x9", "x10", "x11", "x12", "x13",
+                           "x14", "x15", "y")
> m <- lm(y~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15,
+         data=train_data)
> res_test <- svd(x_test)
> U_test <- res_test$u
> V_test <- res_test$v
> d_test <- res_test$d
> Z_test <- U_test[, 1:k] %*% diag(d_test[1:k], nrow=k)
> test_data <- data.frame(Z_test)
> colnames(test_data) <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7",
+                          "x8", "x9", "x10", "x11", "x12", "x13",
+                          "x14", "x15")
> p <- as.matrix(predict.lm(m, newdata=test_data) > cut)
>
> c <- c()
> i <- 1
```

```
> for (j in 1:length(y_test)) {
+   if (p[j,] == FALSE && y_test[j,] == TRUE) {
+     c[i] <- j
+     i <- i + 1
+   }
+ }
>
> par(mfrow=c(1, 5))
> plot.digit(testzip[c[1],])
> plot.digit(testzip[c[2],])
> plot.digit(testzip[c[3],])
> plot.digit(testzip[c[4],])
> plot.digit(testzip[c[5],])
```