丁香园

丁香公开课
CLASS.DXY.CN

临床预测模型构建&机器学习(R语言进阶)

第5章 线性模型中的高级特征选择技术

周支瑞

CONTENT

线性回归的一般形式

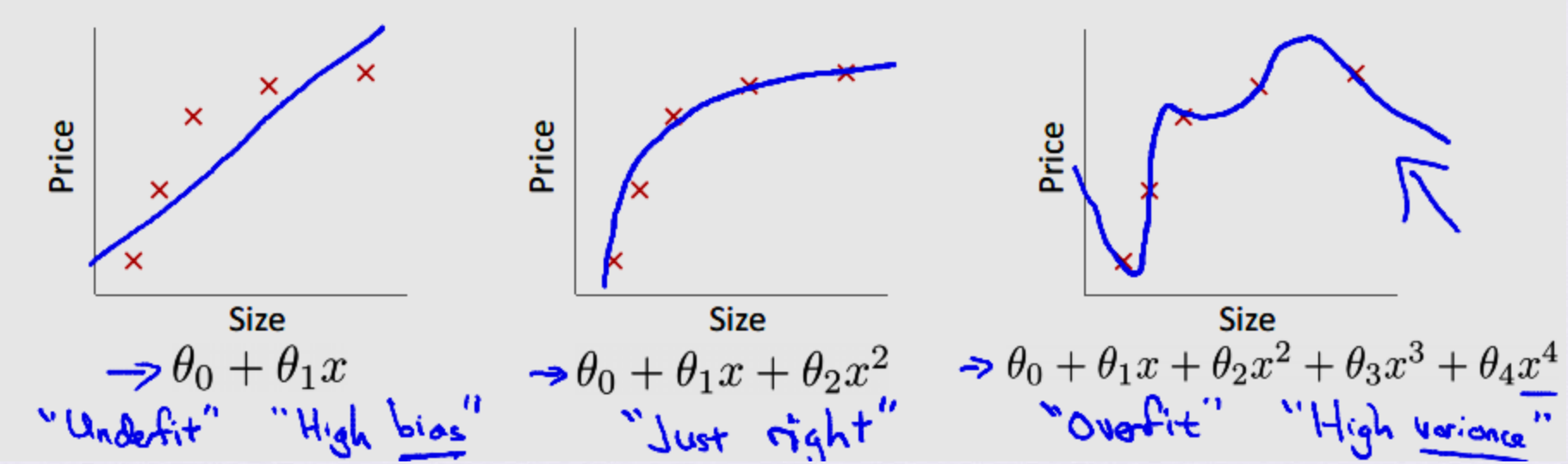
假设函数：$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n$

损失函数：$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2$

目标：$\min J(\theta_0, \theta_1, \ldots, \theta_n)$

进一步理解过拟合的问题

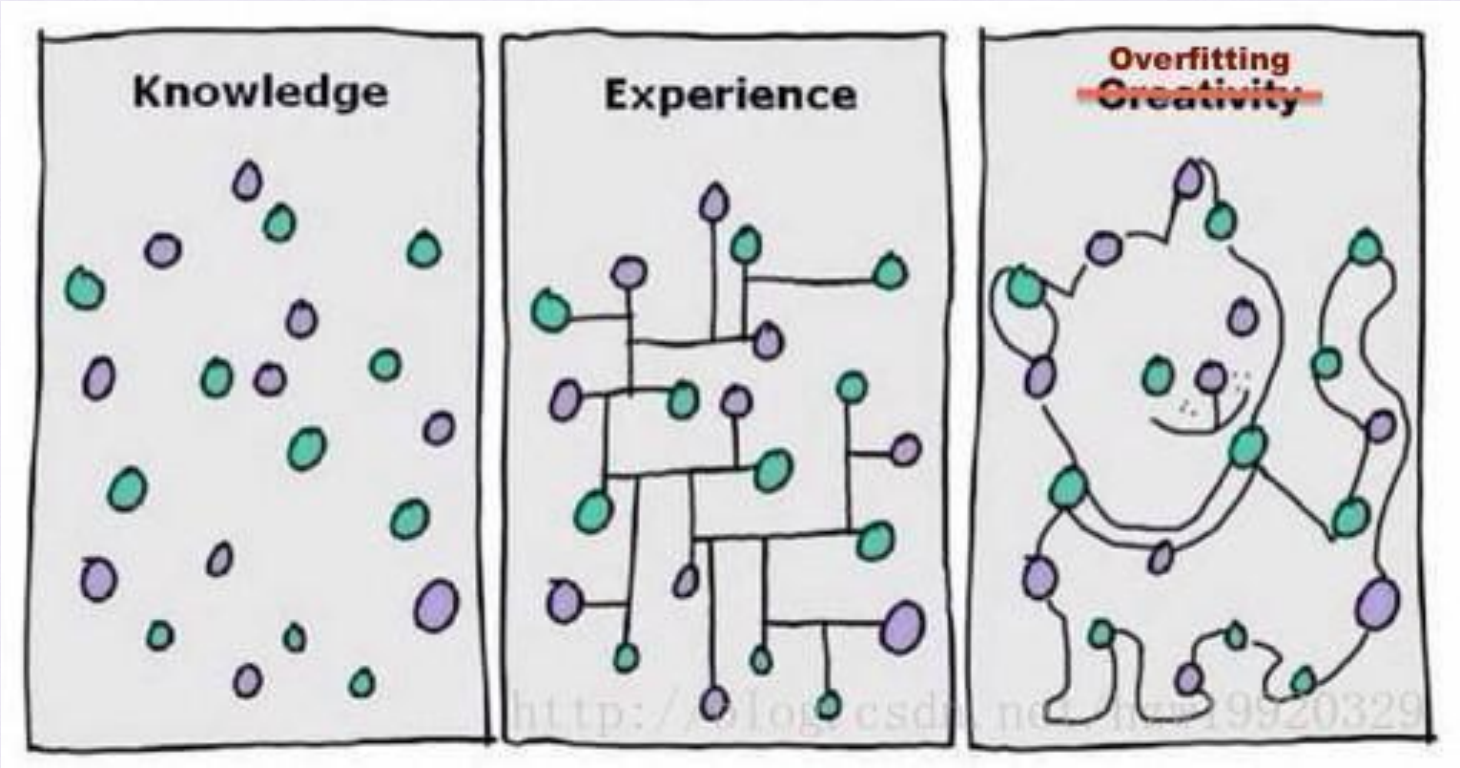
- 由统计学知识知，当训练集数据足够大时，经验风险最小化能够保证得到很好的学习效果。当训练集较小时，则会产生过拟合现象。虽然对训练数据的拟合程度高，但对未知数据的预测精确度低，这样的模型不是适用的模型。



图片引用自https://blog.csdn.net/Byron309/article/details/77716127

线性回归中可能遇到的问题

- 以下面一张图片展示过拟合问题



- 解决方法：(1)：丢弃一些对我们最终预测结果影响不大的特征，具体哪些特征需要丢弃可以通过PCA算法来实现；(2)：使用正则化技术，保留所有特征，但是减少特征前面的参数θ的大小，具体就是修改线性回归中的损失函数形式即可，岭回归以及Lasso回归就是这么做的。

正则化简介

- 线性模型形式为Y=B0+B1X1+…+BnXn+e，最佳拟合试图最小化RSS。RSS是实际值减去估计值的差的平方和，可以表示为e1^2 + e2^2+…+en^2。通过正则化，我们会在RSS的最小化过程中加入一个新参数，称之为收缩惩罚项。这个惩罚项包含了一个希腊字母λ以及对β系数和权重的规范化结果。不同的技术对权重的规范化方法都不尽相同。简言之，我们在模型用RSS+λ（规范化后的系数）代替RSS。我们对λ进行选择，在模型构建过程中，λ被称为调优参数。如果λ = 0，模型就等价于OLS，因为规范化项目都被抵消。

正则化的优势

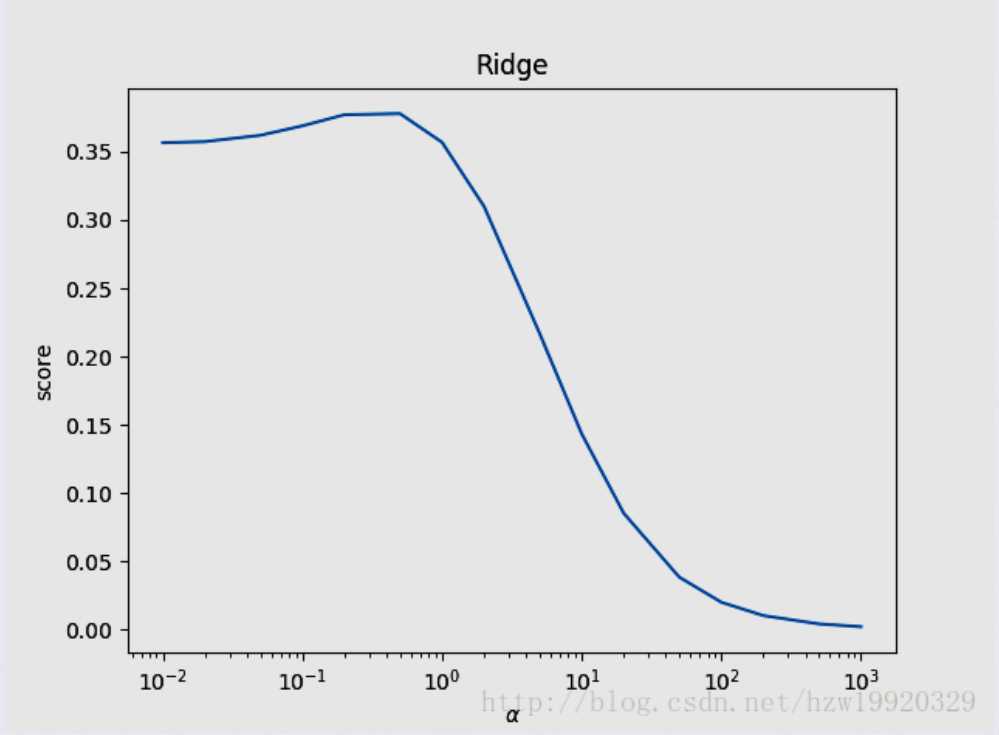
- 首先，正则化方法在计算上非常有效。如果使用最优子集法，我们需要在一个大数据集上测试2^p个模型，这肯定是不可行的。如果使用正则化方法，对于每个λ值，我们只需拟合一个模型，因此效率会有极大提升。
- 其次，是偏差/方差权衡问题。在线性模型中，响应变量和预测变量之间的关系接近于线性，最小二乘估计接近于无偏，但可能有很高的方差。这意味着，训练集中的微小变动会导致最小二乘系数估计结果的巨大变动（James，2013）。正则化通过恰当地选择λ和规范化，可以使偏差/方差权衡达到最优，从而提高模型拟合的效果。
- 最后，系数的正则化还可以用来解决多重共线性引起的过拟合问题。

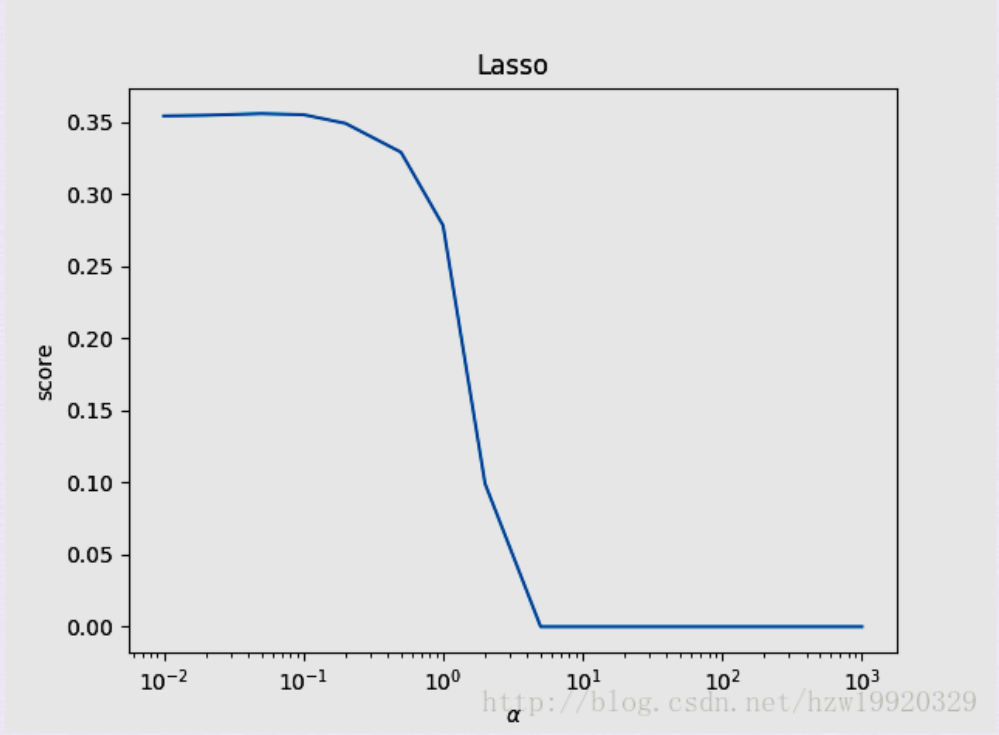
岭回归

- 在岭回归中，规范化项是所有系数的平方和，称为L2-norm（L2范数）。在我们的模型中就是试图最小化RSS+λ(sumβj^2)。当λ增加时，系数会缩小，趋向于0但永远不会为0。岭回归的优点是可以提高预测准确度，但因为它不能使任何一个特征的系数为0，所以在模型解释性上会有些问题。为了解决这个问题，我们使用LASSO回归。

LASSO回归

- 区别于岭回归中的L2-norm，LASSO回归使用L1-norm，即所有特征权重的绝对值之和，也就是要最小化RSS+λ(sum|βj|)。这个收缩惩罚项确实可以使特征权重收缩到0，相对于岭回归，这是一个明显的优势，因为可以极大地提高模型的解释性。
- 如果LASSO这么好，那还要岭回归做什么？当存在高度共线性或高度两两相关的情况下，LASSO回归可能会将某个预测特征强制删除，这会损失模型的预测能力。举例来说，如果特征A和B都应该存在于模型之中，那么LASSO可能会将其中一个的系数缩减到0。可见岭回归与Lasso回归应该是互为补充的关系。





Lasso回归最终会趋于一条直线，原因就在于好多θ值已经均为0，而岭回归却有一定平滑度，因为所有的θ值均存在。

弹性网络

- 弹性网络的优势在于，它既能做到岭回归不能做的特征提取，又能实现LASSO不能做的特征分组。弹性网络包含了一个混合参数α，它和λ同时起作用。α是一个0和1之间的数，λ和前面一样，用来调节惩罚项的大小。
- 请注意，当α等于0时，弹性网络等价于岭回归；当α等于1时，弹性网络等价于LASSO。实质上，我们通过对β系数的二次项引入一个第二调优参数，将L1惩罚项和L2惩罚项混合在一起。通过最小化（RSS+λ[(1-α)(sum|βj|^2)/2+α(sum|βj|)]/N）完成目标。

CONTENT

案例

- 本章案例是一个前列腺癌数据。虽然这个数据集比较小，只有97个观测共9个变量，但通过与传统技术比较，足以让我们掌握正则化技术。斯坦福大学医疗中心提供了97个病人的前列腺特异性抗原（PSA）数据，这些病人均接受前列腺根治切除术。我们的目标是，通过临床检测提供的数据建立一个预测模型预测患者术后PSA水平。对于患者在手术后能够恢复到什么程度，PSA水平可能是一个较为有效的预后指标。手术之后，医生会在各个时间区间检查患者的PSA水平，并通过各种公式确定患者是否康复。术前预测模型和术后数据（这里没有提供）互相配合，就可能提高前列腺癌诊疗水平，改善其预后。

数据包含的变量

- 收集自97位男性的数据集保存在一个含10个变量的数据框中，如下所示：

  lcavol：肿瘤体积的对数值

  lweight：前列腺重量的对数值

  age：患者年龄（以年计）

  lbph：良性前列腺增生（BPH）量的对数值，非癌症性质的前列腺增生。

  svi：精囊是否受侵，一个指标变量，表示癌细胞是否已经透过前列腺壁侵入精囊腺（1=是，0=否）。

  lcp：包膜穿透度的对数值，表示癌细胞扩散到前列腺包膜之外的程度。

  gleason：患者的Gleason评分；由病理学家进行活体检查后给出（2~10），表示癌细胞的变异程度——评分越高，程度越危险。

  pgg45：Gleason评分为4或5所占的百分比。

  lpsa：PSA值的对数值，响应变量。

  train：一个逻辑向量（TRUE或FALSE，用来区分训练数据和测试数据）。

- 这个数据集包含在ElemStatLearn这个R包内。加载所需的程序包和数据框之后，查看变量以及变量之间可能存在的联系，如下所示：

数据准备 代码

```
library(ElemStatLearn) #contains the data
library(car) #package to calculate Variance Inflation Factor
library(corrplot) #correlation plots
library(leaps) #best subsets regression
library(glmnet) #allows ridge regression, LASSO and elastic net
library(caret) #this will help identify the appropriate parameters
data(prostate)
str(prostate)
```

数据可视化 代码


```
plot(prostate)
plot(prostate$gleason, ylab = "Gleason Score")
table(prostate$gleason)
boxplot(prostate$lpsa ~ prostate$gleason, xlab = "Gleason Score",
      ylab = "Log of PSA")
prostate$gleason <- ifelse(prostate$gleason == 6, 0, 1)
table(prostate$gleason)
p.cor = cor(prostate)
corrplot.mixed(p.cor)
```

划分训练集与验证集 代码


```
train <- subset(prostate, train == TRUE)[, 1:9]
str(train)
test = subset(prostate, train==FALSE)[,1:9]
str(test)
```

CONTENT

最优子集建模

- 数据已经准备好，下面我们将开始构建模型。为了进行对比，先用最优子集回归建立一个模型，然后使用正则化技术建立模型。
- 通过regsubsets()命令建立一个最小子集对象，然后指定训练数据集。选择出的特征随后用在测试集上，通过计算均方误差来评价模型。我们建立模型的语法为lpsa~.，使用“~.”说明，要使用数据框中除响应变量之外的所有变量进行预测。代码如下:

最优子集建模 代码


```
subfit <- regsubsets(lpsa ~ ., data = train)
b.sum <- summary(subfit)
which.min(b.sum$bic)
plot(b.sum$bic, type = "l", xlab = "# of Features", ylab = "BIC",
    main = "BIC score by Feature Inclusion")
plot(subfit, scale = "bic", main = "Best Subset Features")

ols <- lm(lpsa ~ lcavol + lweight + gleason, data = train)
plot(ols$fitted.values, train$lpsa, xlab = "Predicted", ylab = "Actual",
    main = "Predicted vs Actual")

pred.subfit = predict(ols, newdata=test)
plot(pred.subfit, test$lpsa , xlab = "Predicted",
    ylab = "Actual", main = "Predicted vs Actual")
resid.subfit = test$lpsa - pred.subfit
mean(resid.subfit^2)
```

CONTENT

岭回归建模

- 在岭回归中，我们的模型会包括全部8个特征，所以岭回归模型与最优子集模型的比较令人期待。我们要使用的程序包glmnet。这个程序包要求输入特征存储在矩阵中，而不是在数据框中。岭回归的命令形式为glmnet(x=输入矩阵,y=响应变量,family=分布函数,alpha=0)。这里的alpha为0时，表示进行岭回归；alpha为1时，表示进行LASSO回归。要准备好供glmnet使用的训练集数据也很容易，使用as.matrix()函数处理输入数据，并建立一个向量作为响应变量，代码如下所示：

岭回归建模 代码

```
x <- as.matrix(train[, 1:8])
y <- train[, 9]
ridge <- glmnet(x, y, family = "gaussian", alpha = 0)
print(ridge)
plot(ridge, label = TRUE)
plot(ridge, xvar = "lambda", label = TRUE)
ridge.coef <- predict(ridge, s=0.1, type = "coefficients")
ridge.coef
plot(ridge, xvar = "dev", label = TRUE)

newx <- as.matrix(test[, 1:8])
ridge.y = predict(ridge, newx = newx, type = "response", s=0.1)
plot(ridge.y, test$lpsa, xlab = "Predicted",
    ylab = "Actual", main = "Ridge Regression")
ridge.resid <- ridge.y - test$lpsa
mean(ridge.resid^2)
```

CONTENT

LASSO回归建模

- 运行LASSO就非常简单了，只要改变岭回归模型的一个参数即可。也就是说，在glmnet()语法中将岭回归中的alpha=0变为alpha=1。

Lasso回归代码


```
lasso <- glmnet(x, y, family = "gaussian", alpha = 1)
print(lasso)
plot(lasso, xvar = "lambda", label = TRUE)
lasso.coef <- predict(lasso, s = 0.045, type = "coefficients")
lasso.coef
lasso.y <- predict(lasso, newx = newx,
                   type = "response", s = 0.045)
plot(lasso.y, test$lpsa, xlab = "Predicted", ylab = "Actual",
    main = "LASSO")
lasso.resid <- lasso.y - test$lpsa
mean(lasso.resid^2)
```

弹性网络建模

- caret包旨在解决分类问题和训练回归模型，它配有一个很牛的网站，帮助人们掌握其功能：http://topepo.github.io/caret/index.html。这个软件包有很多强大功能可以使用。现在我们的目的集中于找到λ和弹性网络混合参数α的最优组合。可以通过下面3个简单的步骤完成。
  （1）使用R基础包中的expand.grid()函数，建立一个向量存储我们要研究的α和λ的所有组合。
  （2）使用caret包中的trainControl()函数确定重取样方法，可使用LOOCV。
  （3）在caret包的train()函数中使用glmnet()训练模型来选择α和λ值。
  （4）选定参数，像在岭回归和LASSO回归那样，在测试数据上验证。

弹性网络代码 代码

```
grid <- expand.grid(.alpha = seq(0,1, by=.2),
                    .lambda = seq(0.00, 0.2, by = 0.02))
table(grid)
head(grid)
control <- trainControl(method = "LOOCV") #selectionFunction="best"
set.seed(701) #our random seed
enet.train = train(lpsa ~ ., data = train,
                   method = "glmnet",
                   trControl = control,
                   tuneGrid = grid)
enet.train
enet <- glmnet(x, y,family = "gaussian",
               alpha = 0,
               lambda = .08)
enet.coef <- coef(enet, s = .08, exact = TRUE)
enet.coef
enet.y <- predict(enet, newx = newx, type = "response",  s= .08)
plot(enet.y, test$lpsa, xlab = "Predicted",
    ylab = "Actual", main = "Elastic Net")
enet.resid <- enet.y - test$lpsa
mean(enet.resid^2)
```

CONTENT

弹性网络

- 使用R基础包中的expand.grid()函数，建立一个向量用于存储我们要研究的α和λ的所有组合。
- 使用caret包中的trainControl()函数确定重取样方法，可使用LOOCV。
- 在caret包的train()函数中使用glmnet()训练模型来选择α和λ。一旦选定参数，我们会像在岭回归和LASSO回归中做的那样，在测试数据上使用它们。

交叉验证

- glmnet包在使用cv.glmnet()估计λ值时，默认使用10折交叉验证。在K折交叉验证中，数据被划分成k个相同的子集（折），每次使用k-1个子集拟合模型，然后使用剩下的那个子集做测试集，最后将k次拟合的结果综合起来（一般取平均数），确定最后的参数。在这个方法中，每个子集只有一次用作测试集。在glmnet包中使用K折交叉验证非常容易，结果包括每次拟合的λ值和响应的MSE。默认设置为α=1，所以如果你想试试岭回归或弹性网络，必须指定α值。因为我们想看看尽可能少的输入特征的情况，所以还是使用默认设置，但由于训练集中数据量的原因，只分3折：

交叉验证 代码


```
set.seed(317)
lasso.cv = cv.glmnet(x, y, nfolds = 3)
plot(lasso.cv)
lasso.cv$lambda.min #minimum
lasso.cv$lambda.1se #one standard error away
coef(lasso.cv, s = "lambda.1se")
lasso.y.cv = predict(lasso.cv, newx=newx, type = "response",
                     s = "lambda.1se")
lasso.cv.resid = lasso.y.cv - test$lpsa
mean(lasso.cv.resid^2)
```

CONTENT

模型选择

通过对数据集的分析，我们得出5个不同模型。下面是这些模型在测试集上的误差。

（1）最优子集模型：0.51
（2）岭回归模型：0.48
（3）LASSO模型：0.44
（4）弹性网络模型：0.48
（5）LASSO交叉验证模型：0.45

模型选择

- 仅看误差的话，7特征LASSO模型表现最好。但是，这个最优模型能解决我们试图回答的问题吗？我们通过交叉验证得到λ值约为0.125的模型，它更简约，也可能更加合适。我更倾向于选择它，因为其解释性更好。说到这里，显然需要来自肿瘤专家、泌尿科专家和病理学家的专业知识，来帮助我们搞清什么是最有意义的。确实如此，但同时也需要更多数据。
- 在本例的样本规模之下，仅改变随机数种子或重新划分训练集和测试集都可能使结果发生大的改变（你可以试试）。

CONTENT

正则化与分类

- 上面使用的正则化技术同样适用于分类问题，二值分类和多值分类皆可。因此，结束本章之前，我们再介绍一下可以用于logistic回归问题的示例代码。更具体地说，是可以用于第三章乳腺癌数据集biopsy的代码。在具有定量型响应变量的回归问题中，正则化是一种处理高维数据集的重要技术。
- 因为Logistic回归函数中有线性的部分，所以可以联合使用L1和L2正则化。和前一章一样，先加载并准备好乳腺癌数据:

正则化与分类 代码


```
library(MASS)
biopsy$ID = NULL
names(biopsy) = c("thick", "u.size", "u.shape", "adhsn",
                  "s.size", "nucl", "chrom", "n.nuc", "mit", "class")
biopsy.v2 <- na.omit(biopsy)
set.seed(123) #random number generator
ind <- sample(2, nrow(biopsy.v2), replace = TRUE, prob = c(0.7, 0.3))
train <- biopsy.v2[ind==1, ] #the training data set
test <- biopsy.v2[ind==2, ] #the test data set
x <- as.matrix(train[, 1:9])
y <- train[, 10]
set.seed(3)
fitCV <- cv.glmnet(x, y, family = "binomial",
                   type.measure = "auc",
                   nfolds = 5)
plot(fitCV)
fitCV$lambda.1se
coef(fitCV, s = "lambda.1se")
```

正则化与分类 代码续


```
library(InformationValue)
predCV <- predict(fitCV, newx = as.matrix(test[, 1:9]),
                  s = "lambda.1se",
                  type = "response")
actuals <- ifelse(test$class == "malignant", 1, 0)
misClassError(actuals, predCV)
plotROC(actuals, predCV)

predCV.min <- predict(fitCV, newx = as.matrix(test[, 1:9]),
                  s = "lambda.min",
                  type = "response")
misClassError(actuals, predCV.min)
```

小结

- 本章的目标是，通过一个数据量较小的prostate数据集介绍如何对线性模型应用高级特征选择技术。数据集的结果变量是定量的，但我们使用的glmnet包也支持定性的结果变量（二值分类和多值分类）。我们介绍了正则化及其包含的3种技术，并应用这些技术构建模型，然后进行了比较。正则化是一项强大的技术，与其他建模技术相比，既可以提高计算效率，还可以提取更有意义的特征。此外，我们还开始使用caret包在训练模型时使多个参数达到最优化。

请在此处输入小标题

感谢观看

THANKS

丁香园特邀讲师 周支瑞