北京大学

学士学位论文

题目 宝洁统计大赛案例分析

姓　　名　占翔

学　　号　00601128

院　　系　数学科学学院

专　　业　概率统计系

研究方向　统计学

指导教师　姚远

2010.6.12

Case Analysis of P& G Statistics Competition

Zhan Xiang

Supervisor: Yao Yuan

School of Mathematical Sciences, Peking University

June, 2010

Submitted in total fulfilment of the requirements for the degree of Bachelor in Probability and Statistics

版权声明

摘　　要

本文以宝洁统计大赛为背景，用各种统计模型和方法来处理和拟合宝洁大赛所给的数据。线性模型在统计学中有了很大的发展，它们不仅简单而且往往对输入如何影响输出提供充分和可解释的描述。均方误差可以分解为方差和偏差平方两部分，高斯-马尔科夫定理告诉我们：在所有的无偏估计中，最小二乘估计具有最最小的方差。一个自然的想法是去掉无偏这一个的限制，牺牲偏差但使得方差大幅度减小，从而降低均方误差。本文主要讨论的就是这些方法，涉及到的主要有：子集选择，岭回归，套索回归。最后也会扩展地讨论一些非线性的方法，如：树和多元自适应回归样条。

关键词： AIC准则，交叉验证，最佳子集选择，岭回归，套索回归，树，多元自适应回归样条

Abstract

This paper is based on P&G Statistics Competition, we use several statistical models to deal with the data given byP&G. Linear models are simple and often provide an adequate and interpretable description of how the inputs affect the outputs. Mean square error can be divided into two parts: variance and the square of the bias. Gauss-Markov theorem tells us, that among all unbiased estimations, ordinary least square estimation has the least mean square error. However if we drop out the restriction of unbiased estimation, that is we make a trade-off between bias and variance, we may get a lower mean square error estimation. And this is what we will mainly focus on in this paper. *Subset selection, Ridge regression* and*Lasso regression* are the three linear models included in this paper. In addition we will consider some non-linear models such as *Tree* and*MARS*.

Keywords: AIC,Cross Validation,Best subset selection, Ridge regression,Lasso,Tree,MARS

目　　录

第一章　处理数据

1.1　最近邻法填补数据

本文以宝洁”亲近生活，美化生活”2010统计创新大赛的课题一为背景，主要是应用一些线性的回归模型来处理课题一，在建立这些统计模型之前，有必要介绍一下宝洁统计大赛课题一的相关背景:洗衣粉是通过其中的化学成分溶于水后改变水溶液的物理化学性质来实现去污的作用的，因此通过测量洗衣产品溶于水后的溶液的一些属性就可以了解产品去污的功效。如果能建立溶液属性和产品功效之间的模型，就可以找出能够最大化产品功效的溶液的属性，根据这些属性和化工技术知识我们就可以找出最优的配方。

现有96个产品的物理属性及功效数据，从中随机选取了10个产品作为验证模型预测精度的数据，请用剩下的86组数据(部分数据缺失)来建立模型。每一个产品的21个属性作为输入变量(PP1-PP21)。产品在18种污渍上的功效作为输出变量(O1-O18)。要求如下：

1) 请根据现有数据拟合出一个统计模型，模型能够基于产品的属性数据对产品的功效做出比较可靠的预测;

2) 考虑所有输入变量的线性项，根据模型的需要选择它们的平方项及交互作用项；

3) 对此数据用多种不同的方法进行分析

4) 选择合适的能够反映模型预测能力的评价准则（可以根据需要提出新的准则），并根据准则选出最优的建模方法和最优模型；

5) 提供数据说明拟合出的模型的预测能力。

先观察数据发现一些总体上的特征，所给的数据一共有86个观测，21个自变量PP1 PP21，18个因变量O1 O18,其中缺失了部分数据。前50个观测的数据是完整的，但是后32个观测的PP2 PP5变量有不同情况的缺失。进一步观察发现PP2和PP3是成对缺失的，PP4和PP5也是成对缺失的。在训练之前，我们用3-最近邻方法来填补缺失数据。由于86个观测的PP1，PP6 PP21这17个分量都是完整的，我们可以利用完整的这17为来定义距离，缺失观测的值用与其最邻近的3个观测的值的平均来估计。这一想法的R代码见附录1。

1.2 观察数据

在回归之前，首先考察一下自变量的共线性问题。称一组自变量$X_1X_2...X_p$是共线性的,如果存在不全为0的常数$C_0C_1...C_p$使得等式$C_1X_1+...+C_pX_p=C_0$成立。当自变量存在共线性时，系数估计的方差就会增大。例如考察只有两个自变量的回归,$Y=\beta_0+\beta_1X_1+\beta_2X_2+\varepsilon$ 假设X_1和X_2 之间样本相关系数为r_{12},定义符号$SX_iX_i=\sum\left(X_i-\overline{X}_i\right)^2$ 则可以证明：

$$\mathrm{var}(\widehat{\beta_i})=\sigma^2(\frac{1}{1-r_{12}^2})\frac{1}{SX_iX_i}(i=1,2)$$

当X_1和X_2之间存在较强的共线性时，r_{12}趋近于1，从上式可以看出$\widehat{\beta_i}$的方差会趋向无穷大。对于$p>2$ 的情形，定义R_k^2为X_k和其他X之间的复相关系数的平方。它是通过做X_k关于其他X的回归计算得到的。如果R_k^2接近1，则可以可能存在共线性问题。当X来自多元正态分布样本时，上面定义的相关系数是共线性问题的一个敏感的诊断统计量。非正态情形时，要度量共线性问题必须从原始的定义出发。当等式$C_1X_1+...+C_pX_p=C_0$近似成立时，存在一个单位向量c使得Xc几乎等于0向量。考察向量的长度，对于这个c，$(Xc)^TXc=c^TX^TXc$充分接近于0。可以证明：对于任意c,c^TX^TXc大于等于X^TX的最小特征值。因此，如果相对于其他特征值而言，最小特征值充分小的话，可以判断出共线性。一个基于特征值的常用量称为条件数k定义为：k=（最大特征值/最小特征值)$^{\frac{1}{2}}$。条件数作为共线性的经验上一种度量，在理论上并没有严格的证明。Berk等人提出如果$k\geq30$。则可以认为存在共线性问题。至此，我们可以根据上面的结果来考察数据的共线性问题。计算易得K $=280.0878>>30$ 故可以认为X有很强的共线性。我们可以通过考察变量之间的散点图来证实。

从图1.1可知自变量PP4与PP5有很强的共线性。处理共线性问题的一个常见

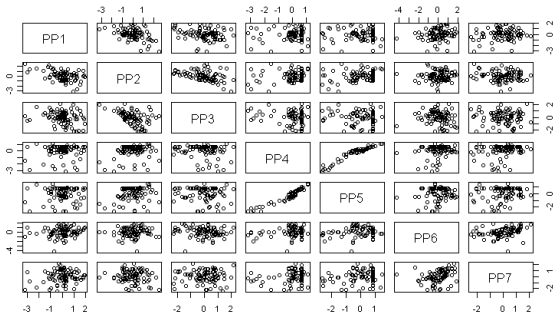

图 1.1: 部分自变量之间散点图

做法是删除部分自变量，如果共线性严格成立，那么删除不会造成信息的丢失。当共线性关系近似成立时，删除那些使模型信息丢失最少的自变量。

第二章　模型的评估与选择

2.1　模型的复杂性

我们有一个目标变量Y，一个输入变量X和一个由已知训练样本估计的预测模型$\widehat{f(x)}$。度量Y和$\widehat{f(x)}$之间误差的损失函数记作$L(Y,\widehat{f(x)})$。典型的损失函数有以下两个：

1）均方误差：$L(Y,\widehat{f(x)})=(Y-\widehat{f(x)})^2$

2）绝对误差：$L(Y,\widehat{f(x)})=\left|Y-\widehat{f(x)}\right|$

对于分类问题框架下的损失函数，典型的我们有：

1）0-1损失：$L(G,G(x))=I(G\neq G(x))$

2）对数似然损失：$L(G,\widehat{P(x)})=-2\log\widehat{P_G}(x)$ 其中$P_k(x)=\Pr(G=k|X)$

为评估模型，我们定义两个误差：

1）检验误差：它是在独立的检验样本上的期望预测误差：$Err=E[L(Y,\widehat{f(x)})]$ 其中X和Y都是从他们的联合分布（总体）中随机抽取的。

2）训练误差：是在训练样本上的平均损失：$err=\frac{1}{N}\sum_1^N L(Y,\widehat{f(x)})$ 下图是一张典型的检验误差和训练误差随模型变化的图:

从图中可以看出：随着模型越来越复杂，它能够适应更复杂的结构（偏倚减少，方差增加）其间存在最佳模型复杂性，它产生最小的检验误差。遗憾的是训练误差不是检验误差的一个很好估计，如果模型的复杂性增加到足够大，典型的

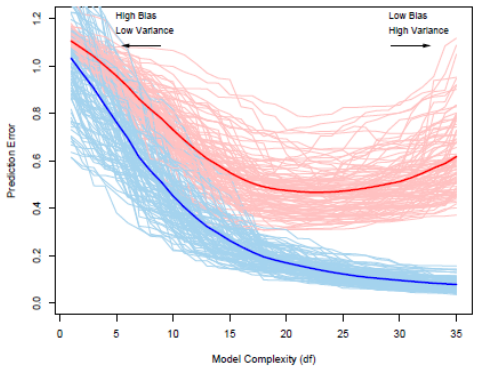

图 2.1: 模型复杂度与预测误差之间关系图，其中红线代表检验误差，蓝线代表训练误差

训练误差会减小到0，而这样过分拟合数据的模型的检验误差通常比较差。

如果我们的数据量足够大，我们可以把数据随机的分为三部分：训练集，验证集和检验集。训练集用来拟合模型，验证集用于模型选择：最终的模型应该在验证集上具有最小的检验误差。检验集用来评估模型在新数据上的预测误差，理想的检验集应该是事先"密闭"的，直到数据分析结束后在拿出来使用。而然，现实是很难给出一种方法来讲数据划分为三部分。本节后面将主要介绍两种方法或解析地或通过有效地样本重用来近似的实现验证。

2.2 AIC准则

典型地，由于同样的数据被用于拟合和评估误差，训练误差

$$err = \frac{1}{N}\sum_{1}^{N} L(y_i, \widehat{f}(x_i))$$

将小于实际误差$Err = E[L(Y, \widehat{f}(x))]$。因此训练误差err是检验误差Err的一个过分乐观的估计，两者的部分差异是由于计算点的不同引起的。Err是一种样本外误差，因为检验集的输入变量与训练集的输入变量不一样。考虑到这一点，可以定义一个样本内误差：

$$Err_{in} = \frac{1}{N}\sum_{i=1}^{N} E_y E_Y L(Y_i, \widehat{f}(x_i))$$

其中Y_i 是在训练点x_i,$i = 1, 2..., N$处观测到的N个新的响应值。定义乐观性为Err_{in}与训练误差err之间的期望差,记为op。即：$op = Err_{in} - E_y(err)$，从上面的讨论可以知：op是正的。事实上我们有一个更强的结论：对于平方损失，0-1损失和其他损失函数，

$$op = \frac{2}{N}\sum_{i=1}^{N} \mathrm{cov}(\widehat{y_i}, y_i)$$

这个结论的简要证明过程可以参见附录3。上述结论给出了乐观性估计op的一个估计，由此可以估计样本内误差：

$$\widehat{Err_{in}} = err + \widehat{op}$$

给定一个由调整参数α标记的模型族$f_\alpha(x)$用$err(\alpha)$和$d(\alpha)$分别表示模型的训练误差和参数个数。对于线性模型$d(\alpha)$为输入或者基函数的个数，对于非线性和其他复杂模型，$d(\alpha)$ 用模型的复杂度的某种度量来代替。然后，对这个模型我们定义：

$$AIC(\alpha) = err(\alpha) + 2\frac{d(\alpha)}{N}\widehat{\sigma_\varepsilon}^2$$

结合附录3的结果以及上面的讨论可知：$AIC(\alpha)$提供了检验误差曲线的一个估计。我们的目标就是寻找使其极小化的调整参数α,我们最终选择的模型就是$f_{\widehat{\alpha}}(x)$。上面所述的模型选择的方法就是所谓的AIC准则。

2.3 交叉验证

与AIC准则估计样本内误差不同，交叉验证直接估计样本外误差。K折交叉验证使用部分数据进行拟合模型，而用剩下的数据去检验模型。我们将数据大致分为K等分。每次取出第k部分作为检验集，剩下的（K-1）部分作为训练集。用训练集拟合出来的结果去预测检验集，并计算拟合模型的预测误差，对$\mathrm{k} = 1, 2, ..., \mathrm{K}$计算K个预测误差，并合并所有的预测误差。下面用数学语言叙述一下交叉验证。映射κ:$\{1, 2, ..., \mathrm{N}\} \to \{1, 2, ..., \mathrm{K}\}$. 用$\widehat{f}^{-k}(x)$表示用除去第$k$组数据剩下的（K-1）组数据拟合出来的结果,那么预测误差的交叉验证估计是:

$$CV = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \widehat{f}^{-\kappa(i)}(x_i))$$

典型的K的取值为5或10。当K=N是就是所谓的留一交叉验证。

给定一个由调整参数α标引的模型$f(x, \alpha)$的集合，则对于这个模型集，我们定义:

$$CV(\alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \widehat{f}^{-\kappa(i)}(x_i, \alpha))$$

函数$CV(\alpha)$提供了检验误差曲线的一个估计，我们寻找它的极小化调整参数$\widehat{\alpha}$，相应的我们得到的最优模型为:$f(x, \widehat{\alpha})$。

对于平方误差损失函数下的线性拟合，广义交叉验证提供了一种对留一交叉验证方便的逼近。对于线性拟合方法, $\widehat{y} = Sy$,其中S是的矩阵，依赖于x_i但不依赖于y_i。$\widehat{y}$和y都是维向量。对于许多线性拟合方法，留一交叉验证的校验误差估计为：

$$CV = \frac{1}{N}\sum_{i=1}^{N} [y_i - \widehat{f}^{-i}(x_i)]^2 = \frac{1}{N}\sum_{i=1}^{N} [\frac{y_i - \widehat{f}(x_i)}{1 - S_{ii}}]^2$$

其中S_{ii}是矩阵S的第i个对角元素。

所谓的广义交叉验证，其实是对留一交叉验证给出了一个近似的估计：

$$GCV(\widehat{f}) = \frac{1}{N}\sum_{i=1}^{N} [\frac{y_i - \widehat{f}(x_i)}{1 - Trace(S)/N}]^2$$

绝大多数情况下，S的迹会比S的对角元素更易于计算，GCV有计算上的优势，同时GCV也可以缓和交叉验证光滑不足的趋势。统计软件R中，经常采用这种方法来进行模型选择。

第三章　最佳子集选择

3.1　子集选择

最小二乘估计的一个弊端是解释性不强，尤其是存在大量预测因子的情况。通常我们希望确定一个表现出对因变量有最强影响的较小子集。为了获得这种更深刻的印象，我们情愿牺牲某些小的偏差。最佳子集回归是指,对每个$k \in \{1, 2, ...p\}$,找出容量为k的子集,使得在所有容量为k的子集回归中，它具有最小的残差平方和。残差平方和与k成反比，因此不能最为选择子集容量k的标准。而应该采用第二章模型选择中的标准。R软件中采用的是AIC准则。

显然当p 比较大时，搜索所有可能的子集是不可取的，但可以寻找一条通过可能子集的好的路径。一种可循的方法称为逐步前向选择（forward stepwise selection）。该法由截距开始，依次将对拟合改进最大的预测子添加到模型中去，可以用F统计量衡量对模型拟合的改进。假设当前模型中有m个自变量，参数估计用$\widehat{\beta}$表示。添加一个自变量后的参数估计为$\widetilde{\beta}$衡量拟合的改进通常基于F统计量:

$$F = \frac{RSS(\widehat{\beta}) - RSS(\widetilde{\beta})}{RSS(\widetilde{\beta})/(N - m - 2)}$$

典型的策略是添加产生最大F值的自变量。给定显著性水平α,当没有一个自变量添加后产生的F统计量大于$F_{1,N-m-2}$分布的$1 - \alpha$分位数时，停止添加自变量。

与逐步前向选择相对应得是逐步后向选择（backward stepwise selection），从包括p个自变量的完整模型开始，依次删除自变量。每次删除产生最小F值的自变量。直到删除模型中的任何一个自变量时产生的F值都大于给定显著性水平的F临界值。后向选择仅当$N > p$时才可以使用。

3.2　宝洁数据例子

我们以$O1$为因变量，用最佳子集，岭回归和Lasso回归拟合数据，同样的过程完全可以在因变量$O2...O18$上重复，出于对篇幅的考虑，我们省略掉这一部分内容。

```
Coefficients:
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) -2.242e-17  8.790e-02 -2.55e-16   1.0000
PP1          1.523e-01  1.582e-01     0.963   0.3393
PP2          1.186e-01  1.661e-01     0.714   0.4779
PP3          3.463e-01  1.497e-01     2.314   0.0239 *
PP4         -2.375e-01  3.707e-01    -0.641   0.5240
PP5          4.708e-02  3.718e-01     0.127   0.8996
PP6         -1.582e-01  1.146e-01    -1.380   0.1725
PP7         -1.141e-01  1.113e-01    -1.025   0.3090
PP8         -1.191e-01  1.402e-01    -0.849   0.3989
PP9          4.034e-01  1.529e-01     2.639   0.0104 *
PP10        -3.733e-01  1.657e-01    -2.253   0.0277 *
PP11         3.763e-01  1.896e-01     1.984   0.0515 .
PP12        -2.805e-01  1.886e-01    -1.487   0.1420
PP13         4.449e-01  1.530e-01     2.908   0.0050 **
PP14         1.247e-01  2.062e-01     0.605   0.5476
PP15        -3.970e-02  1.918e-01    -0.207   0.8367
PP16         5.656e-02  2.370e-01     0.239   0.8121
PP17        -2.573e-01  2.558e-01    -1.006   0.3183
PP18        -5.226e-02  2.480e-01    -0.211   0.8338
PP19        -7.692e-02  2.070e-01    -0.372   0.7115
PP20         3.637e-01  2.567e-01     1.417   0.1614
PP21        -1.358e-02  2.423e-01    -0.056   0.9555
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '

Residual standard error: 0.8152 on 64 degrees of freedom
Multiple R-squared: 0.5055,     Adjusted R-squared: 0.3432
F-statistic: 3.115 on 21 and 64 DF,  p-value: 0.0002483
```

图 3.1: 线性模型拟合宝洁数据，粗略地讲，绝对值大于2的t值在0.05水平下是显著的

首先对训练集使用最小二乘估计，产生的估计值、标准误、t值、p值如图3.1所示：

从图3.1的结果中可以看出，大部分的变量都是不显著的，这既影响到模型的可解释性，又影响到模型的预测精度。因此有必要通过子集选择来删去部分变量。整个删除变量的过程中，采用第二章中的AIC准则，当AIC值最小时，删除过程停止，得到的就是最佳子集。

比较图3.1和图3.2，从模型p值，$\overline{R}^2$和自变量的p值可以看出，最佳子集方法比普通最小二乘拟合效果更好，解释性更强。进一步，按照宝洁统计大赛的要求，我们考虑添加交叉项和二次项之后进行回归。矩阵yyy 是添加了交叉项和二次项之后的数据矩阵（见附录1），为使问题简化，我们只考虑添加那些在最佳子集回归中显著的那些自变量的交叉项和二次项。并对这个添加交叉项和二次项之后完整模型进行子集选择。图3.3是最终的结果：

从模型p值，$\overline{R}^2$和自变量的p值等方面比较上面三个结果，不难看出带有交叉项和二次项的最佳子集回归的效果是最好的。经计算知拟合分残差平方和为：30.40。

```
Coefficients:
              Estimate Std. Error   t value Pr(>|t|)
(Intercept) -1.815e-17  8.395e-02 -2.16e-16  1.00000
PP1          1.607e-01  1.114e-01     1.443  0.15333
PP3          2.466e-01  9.158e-02     2.692  0.00877 **
PP4         -1.550e-01  9.411e-02    -1.647  0.10389
PP6         -1.783e-01  9.557e-02    -1.865  0.06613 .
PP9          3.040e-01  1.170e-01     2.599  0.01127 *
PP10        -3.518e-01  1.412e-01    -2.491  0.01500 *
PP11         4.371e-01  1.417e-01     3.085  0.00287 **
PP12        -3.477e-01  1.543e-01    -2.253  0.02723 *
PP13         4.504e-01  1.329e-01     3.388  0.00113 **
PP17        -3.003e-01  1.204e-01    -2.495  0.01484 *
PP20         2.385e-01  1.203e-01     1.982  0.05122 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7785 on 74 degrees of freedom
Multiple R-squared: 0.4785,     Adjusted R-squared: 0.401
F-statistic: 6.172 on 11 and 74 DF,  p-value: 4.298e-07
```

图 3.2: 最佳子集回归

```
Coefficients:
                          Estimate Std. Error   t value Pr(>|t|)
(Intercept)             -4.006e-18  7.263e-02 -5.52e-17 1.000000
PP3                      3.664e-01  9.050e-02     4.049 0.000136 ***
PP9                      2.833e-01  9.026e-02     3.138 0.002525 **
PP11                     4.469e-01  1.293e-01     3.457 0.000955 ***
PP12                    -3.713e-01  1.510e-01    -2.460 0.016496 *
PP13                     1.438e-01  9.337e-02     1.540 0.128277
PP17                    -2.589e-01  1.151e-01    -2.250 0.027732 *
yyy[, 39 + 2 * 21 + 3]  -1.770e-01  9.681e-02    -1.828 0.071994 .
yyy[, 39 + 9 + 2 * 21]  -1.542e-01  9.913e-02    -1.555 0.124612
yyy[, 39 + 12 + 2 * 21] -7.953e-01  1.451e-01    -5.483 6.86e-07 ***
yyy[, 39 + 17 + 2 * 21]  3.483e-01  1.204e-01     2.893 0.005143 **
yyy[, 39 + 9 + 9 * 21]   3.211e-01  1.546e-01     2.077 0.041655 *
yyy[, 39 + 9 + 12 * 21] -4.342e-01  1.411e-01    -3.077 0.003029 **
yyy[, 39 + 10 + 9 * 21]  2.739e-01  1.001e-01     2.735 0.007967 **
yyy[, 39 + 11 + 11 * 21] -3.221e-01  1.741e-01    -1.850 0.068676 .
yyy[, 39 + 12 + 11 * 21]  2.544e-01  1.477e-01     1.722 0.089597 .
yyy[, 39 + 12 + 16 * 21] -6.447e-01  2.375e-01    -2.714 0.008444 **
yyy[, 39 + 13 + 12 * 21] -2.394e-01  1.077e-01    -2.223 0.029625 *
yyy[, 39 + 17 + 16 * 21]  3.665e-01  1.895e-01     1.934 0.057292 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6736 on 67 degrees of freedom
Multiple R-squared: 0.6466,     Adjusted R-squared: 0.5516
F-statistic: 6.809 on 18 and 67 DF,  p-value: 2.738e-09
```

图 3.3: 包含交叉项和二次项最佳子集回归

第四章　岭回归和Lasso回归

4.1　岭回归

通过保留某些自变量同时删去其余自变量的方法产生的模型的解释性比原模型强，并可能具有比完整模型更低的预测误差。但是最佳子集回归模型是一个离散的过程。每一个变量或者保留或者去掉，这往往表现出高方差。一系列的收缩方法可以弥补这一缺陷。岭回归就是一种重要的收缩方法。

在标准线性回归模型$Y = X\beta + e$,$\mathrm{var}(e) = \sigma^2$中，最小二乘估计$\widehat{\beta} = (X^TX)^{-1}X^TY$是最小方差线性无偏估计，如果扩大我们所考虑的估计类，把有偏估计考虑进来，并且考虑参数估计的均方误差和作为一个新的准则函数,记为$SMSE$。

$$SMSE = \sum_{i=1}^{p}\{\mathrm{var}(\widehat{\beta}_i) + bias(\widehat{\beta}_i)^2\} = \sum_{i=1}^{p}E(\widehat{\beta}_i - \beta_i)^2 = E(\widehat{\beta} - \beta)^T(\widehat{\beta} - \beta)$$

这一准则函数与导致最小二乘估计的准则是不同的，所以按这一准则推到出来的估计完全有可能比最小二乘估计更好。记$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$是矩阵$X^TX$以排好序的特征值。假设$\widehat{\beta}$ 为最小二乘估计的解，带入上式有：

$$SMSE = E(\widehat{\beta} - \beta)^T(\widehat{\beta} - \beta) = \sigma^2 trace(X^TX)^{-1} = \sigma^2\sum_{i=1}^{p}\lambda_i^{-1}$$

整理有：$E\widehat{\beta}^T\widehat{\beta} = \beta^T\beta + \sigma^2\sum_{i=1}^{p}\lambda_i^{-1} \geq \beta^T\beta + \sigma^2\lambda_p^{-1}$ 当最小特征值λ_p较小时，上式告诉我们：即使最小二乘估计$\widehat{\beta}$是β的无偏估计，但从向量长度出发，两者相差很大。个很自然的想法就是压缩最小二乘估计，一般都是朝原点0压缩。Hoerl和Kennard(1970)完成了这一工作，并提出了岭回归的概念。岭回归的一种原始定义是：

$$\widehat{\beta}^{ridge} = \arg\min\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p}x_{ij}\beta_j)^2$$

受限于$\sum_{i=1}^{p}\beta_i^2 \leq s$其中$s$是模型参数，其选取涉及到偏倚和方差的权衡问题。

从长度加罚的角度我们可以给出岭回归的另外一种定义：

$$\widehat{\beta}^{ridge} = \arg\min\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\}$$

其中$\lambda \geq 0$是控制收缩量的复杂度参数，λ越大，收缩量越大。把上述定义式写成向量形式：

$$RSS(\lambda) = (Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta$$

求偏导后，易看出解为：$\widehat{\beta}^{ridge} = (X^TX + \lambda I)^{-1}X^TY$ 其中I是p阶单位阵，λ称为岭参数。

4.2 lasso回归

Lasso也是一种重要的收缩方法，它与岭回归相比有微妙的区别。Lasso估计有下式定义：

$$\widehat{\beta}^{lasso} = \arg\min\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

受限于$\sum_{i=1}^{p}|\beta_i| \leq t$。从定义可以看出Lasso回归用L1罚代替了岭回归中的L2罚。考察该约束的特性，当t充分小时，将导致某些系数恰好为0，这样Lasso起到了某种连续子集选择的作用。当$t \geq \sum_{i=1}^{p}|\widehat{\beta}_i|$ （其中$\widehat{\beta}_i$是相应的最小二乘估计），起不到任何的约束作用，Lasso解即为最小二乘解。为便于解释，定义标准化参数$s = t/\sum_{i=1}^{p}|\widehat{\beta}_i|$称为收缩系数，$s$的选取通常由交叉验证来确定。

4.3 宝洁数据例子（续）

4.3.1 岭回归

先用chemometrics包里面的plotRidge函数考查岭回归的预测均方误差MSEP,得到如下结果（图4.1）：

可以看到最佳的岭回归参数lambda在25左右。可以用MASS包中的select函数进行精确计算，R的输出结果为：smallest value of GCV at 25.1，岭回归参数取为25.1做相应的岭回归，岭回归的系数估计见图4.2

计算的最后的残差平方和为：48.36。

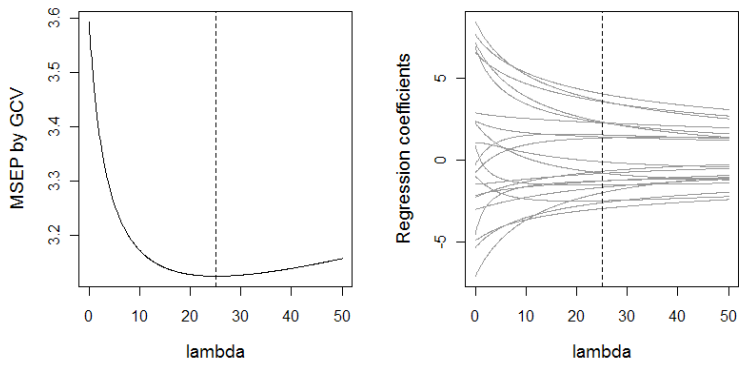

图 4.1: 岭回归的预测误差

```
> fit1
                    PP1           PP2           PP3           PP4
-2.025100e-17  1.201948e-01 -4.178375e-02  1.877257e-01 -6.619815e-02
          PP5           PP6           PP7           PP8           PP9
-7.964345e-02 -8.734495e-02 -6.720733e-02 -3.568313e-02  2.133492e-01
         PP10          PP11          PP12          PP13          PP14
-1.045308e-01  1.224125e-01 -1.367145e-01  1.901528e-01  7.483308e-02
         PP15          PP16          PP17          PP18          PP19
 7.154566e-02 -5.464716e-03 -1.557152e-01 -1.304531e-01 -4.240905e-02
         PP20          PP21
 1.203425e-01  8.149263e-02
```

图 4.2: 岭参数取25.1的岭回归

4.3.2 Lasso回归

首先要确定Lasso回归的收缩系数，采用10折的交叉验证选取。在R中可以用Lars函数包中的cv.lars函数来画10折交叉验证画cv误差图(图4.3)。

从图4.3可以看出最优的收缩系数在0.66附近，进一步地，用lars函数研究回归系数与系数与收缩系数的关系（图4.4)。

从图4.4可以看出，第20步对应的收缩系数约为0.66，取出此时对应的Lasso系数：[1,] 0.10877178 [2,] 0.00000000 [3,] 0.25055661 [4,] -0.13608862 [5,] 0.00000000 [6,] -0.11960692 [7,] -0.08842795 [8,] -0.06482237 [9,] 0.33873160 [10,] -0.26377463 [11,] 0.30062594 [12,] -0.25368455 [13,] 0.35662894 [14,] 0.08657092 [15,] 0.00000000 [16,] 0.00000000 [17,] -0.19482828 [18,] -0.10004874 [19,] 0.00000000 [20,] 0.22931155 [21,] 0.00000000

最后计算回归的残差平方和为44.39。

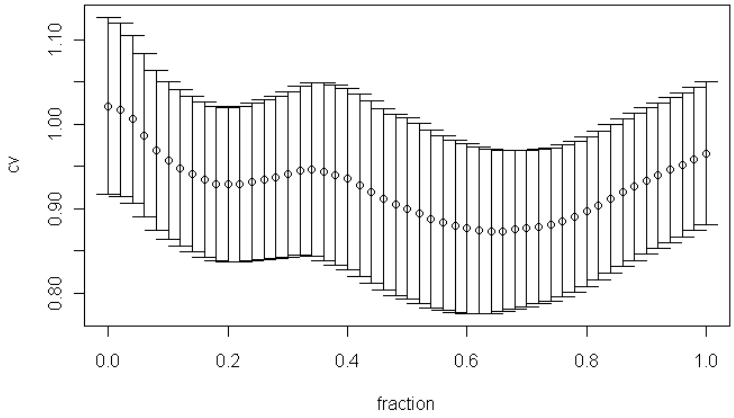

图 4.3: Lasso回归的CV图

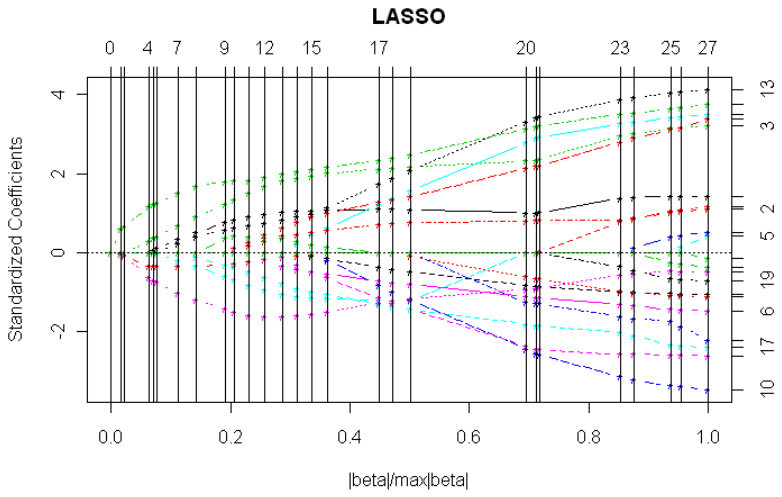

图 4.4: Lasso系数分布图

第五章　非线性模型

5.1　树

基于树的方法把自变量空间（也称特征空间）划分成一系列的矩形区域，然后在每一个矩形区域中拟合一个简单的模型（如常量）。它概念上简单但却很有效。现在假设我们的输入包括p 个自变量和1个因变量,有N次观测。我们的目的是确定分裂变量和分裂点，以及树应该有什么样的拓扑结构。首先假设我们已经将特征空间划分为M 个区域，分别记做$R_1, R_2, ...R_M$,在每个区域内用常数c_m对响应变量Y建模,即：$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$。如果采用平方和$\sum_{i=1}^{N}(y_i - f(x_i))^2$极小化为准则,则容易看出拟合最优的$\widehat{c_m}$为$y_i$在区域$R_m$内的平均值：$\widehat{c_m} = ave(y_i|x_i \in R_m)$然而上述思想在计算上通常是不可行的，我们采用下面的算法处理。考察一个分裂变量X_i和分裂点t ，定义一对半平面区域,

$R_1(i,t) = \{X|X_i \leq t\}$和$R_2(i,t) = \{X|X_i > t\}$

然后收索分裂变量X_i和分裂点t，这等价于求解：

$$\min_{i,t}[\min_{c_1} \sum_{x_j \in R_1(i,t)} (y_j - c_1)^2 + \min_{c_2} \sum_{x_j \in R_2(i,t)} (y_j - c_2)^2]$$

对于任意给定裂变量X_i和分裂点t，上式内部的极小化问题是容易解决的：$\widehat{c_1} = ave(y_j|x_{j\in}R_1(i,t))$,$\widehat{c_2} = ave(y_j|x_{j\in}R_2(i,t))$ 固定分裂变量X_i,计算分裂点t可以很快完成。扫描全部的输入，确定最好的对偶(X_i, t)在计算上是可行的。

找到最好的分裂后，将原有的数据一分为二，然后对每个子区域重复上述分裂过程。进一步对每个结果区域重复上述过程。一个自然的问题是树分裂到什么程度为止。一种想法是当分裂使平方和减小的量小于某个事先给定的阈值时，停止分裂。实际应用中，不同的人对这个问题有不同的观点，现在流行的R软件的控制树的复杂性时采用三个参数：最小节点的大小（默认为5，即每个区域内至少含5个观测），节点的最少个数（默认为10，即至少分裂10次），以及度量节点内方差的量（当方差足够小时，认为数据差异不大，没必要继续分裂）。

5.2 多元自适应回归样条

多元自适应回归样条（MARS）是一个自适应的回归过程，很适合于高维情形和大量输入情形。可以将MARS看做是逐步线性回归的泛化。MARS使用形如$(x-t)_+$和$(t-x)_+$的分段线性基函数展开式，其中符号"+"表示正的部分。即：

$$(x-t)_+ = \begin{cases} x-t & if(x-t)>0 \\ 0 & else \end{cases}$$

;

$$(t-x)_+ = \begin{cases} t-x & if(t-x)>0 \\ 0 & else \end{cases}$$

上述两个函数都是分段线性的，纽结都在t 处，称这两个函数为反演对（reflected pair）。MARS的思想是：每个自变量X_j以其每一个观测值x_{ij}为纽结，形成反演对。这样的所有反演一起构成了基函数集合：

$$C = \{(X_j - t)_+, (t - X_j)_+\}$$

其中$t = x_{1j}, x_{2j}, ...x_{Nj}$,$j = 1, 2, ...p$

如果所有的观测值互不相同，则函数集C 中共有$2Np$ 个基函数。MARS的模型构造策略类似于前向逐步线性回归，但MARS允许使用函数集C中函数的乘积，不仅仅是使用原始的输入。这样，模型具有如下形式：

$$f(x) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(x)$$

其中$h_m(x)$是C 中函数，或者C 中两个或多个函数的乘积,M是给定的模型中包含项数的阈值。用集合H表示已经包含在模型中函数。一开始H 中只有一个常值函数$h_0(x) = 1$。MARS的技巧在于函数$h_m(x)$的构造上，一旦给定模型的函数$h_m(x)$,系数β_m 可以通过极小化残差平方和来估计。

在每个阶段考虑模型集H中的一个函数$h_m(x)$和基函数集C中反演对中一个函数的积，将这样的积看做是一个新的基函数对，每次将如下形式的项添加到模型集H中：

$$\widehat{\beta}_{M+1} h_l(x)(X_j - t)_+ + \widehat{\beta}_{M+2} h_l(x)(t - X_j)_+$$

其中$h_l(x) \in H$

它能最大限度的降低残差。这里$\widehat{\beta}_{M+1}$和$\widehat{\beta}_{M+2}$是用最小二乘估计出来的系数。继

续上述步骤，直到模型集H中的项数达到了预先给定的阈值。添加项的过程中，通常有两个限制，一是：在一个积中，每个自变量只能出现一次，这样可以防止一个自变量的高阶幂形式，因为高阶项在自变量空间的边界上增长或者降低的特别快，没有线性项稳定。另一个限制是交叉项的阶设置上限，一般设置为2，即允许C中分段线性函数两两乘积出现在模型集H 中，但不允许3阶或者更多阶的积，上限为1将导致加法模型。限制交叉项的阶数的一个原因是便于模型最终的解释。

添加项的过程结束后，我们往往会得到一个很大的模型，且多数情形该模型会过分拟合数据，为此MARS还有一个向后删除过程。每一步删除引起残差平方和增加最小的项。至于何时删除停止，大部分软件包中采用的是广义交叉验证准则。

5.3 宝洁数据例子（续）

5.3.1 回归树

R: >o1< −tree(O1 x)

输出结果为：node), split, n, deviance, yval * denotes terminal node

1) root 86 85.0000 -1.507e-16

2) x.PP18 < 0.625741 70 50.6000 2.138e-01

4) x.PP9 < -1.30843 6 10.4900 -9.217e-01 *

5) x.PP9 > -1.30843 64 31.6500 3.203e-01

10) x.PP4 < 0.064627 26 7.3490 6.558e-01

20) x.PP12 < 0.364664 21 5.1530 7.803e-01

40) x.PP1 <-0.53442 7 1.8980 3.549e-01 *

41) x.PP1 > -0.53442 14 1.3540 9.930e-01 *

21) x.PP12 > 0.364664 5 0.5014 1.327e-01 *

11) x.PP4 > 0.064627 38 19.3700 9.069e-02

22) x.PP1 < 0.760289 29 11.1800 2.708e-01

44) x.PP1 < -0.262063 9 2.8980 -2.809e-01 *

45) x.PP1 > -0.262063 20 4.3090 5.191e-01

90) x.PP14 < -0.0717303 5 0.6708 1.146e-02 *

91) x.PP14 > -0.0717303 15 1.9200 6.883e-01 *

23) x.PP1 > 0.760289 9 4.2160 -4.898e-01 *

3) x.PP18 > 0.625741 16 17.2000 -9.354e-01

6) x.PP15 < -0.18024 5 3.5030 -1.975e+00 *

7) x.PP15 > -0.18024 11 5.8420 -4.631e-01

14) x.PP11 < 1.02258 5 0.8747 -3.042e-02 *

15) x.PP11 > 1.02258 6 3.2510 -8.236e-01 *

也可以plot或者text函数来展示回归的结果，图5.1是使用text函数画的树的简图。

最后考察回归树的拟合效果，残差平方和经计算为：31.58。回归树模型拟合数据的效果仅微弱于带交叉项和二次项的最佳子集回归，但是比岭回归和Lasso回归都要好。

5.3.2 多元自适应回归样条

R: > fit< −mars(x,O1,degree=2) *其中degree=2表示交叉项的阶最多为2* 最终的残差为18.10。表明MARS的拟合结果大幅度地好于上述任何模型。MARS对数据的拟合结果好于岭回归，Lasso回归和树是不难解释的，其道理和含交叉项和二次项的最佳子集回归结果由于上述三者一样，都是交叉项对模型的贡献。MARS的拟合结果也比含交叉项的最佳子集也好很多，这是由于MARS所有的交叉项，而在第三章的最佳子集模型中，我们为了计算上的简便，只添加乐人显著自变量的交叉项。

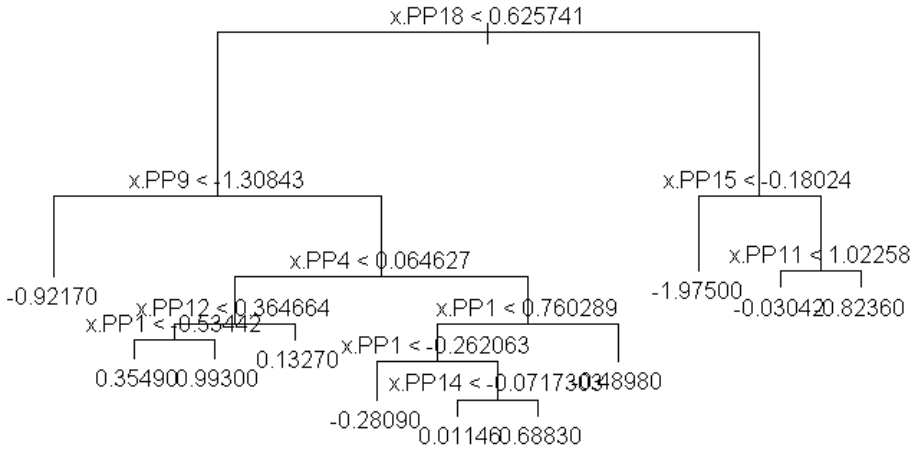
x.PP18 < 0.625741
x.PP9 < -1.30843
x.PP15 < -0.18024
x.PP4 < 0.064627
x.PP11 < 1.02258
-0.92170
x.PP12 < 0.364664
x.PP1 < 0.760289
-1.97500
-0.03042
0.82360
x.PP1 < -0.53442
x.PP1 < -0.262063
0.13270
0.35490
0.99300
x.PP14 < -0.07173
-0.28090
0.01146
0.68830

图 5.1: 回归树的简图

第六章　附录

6.1　用最近邻法补充数据的R代码

R code:

```
x<- read.csv("e:P&G.csv")
z<-x[,2:40]
y<-x[1:50, 2:40]
E<-rep(0,39)
std<-rep(0,39)
for(i in 1:39)
{
E[i]<-mean(y[,i])
std[i]<-sqrt(mean((y[,i]-E[i])^2))
}
yy<-x[,2:40]
for(i in 1:39)
{
yy[,i]<-(yy[,i]-E[i])/std[i]
}
#对有缺失值的数据，选择距离最近的三组数据，用此三组数据的相关的数据的平均填补缺失数据#
#下面距离定义为第1，6：21栏距离差的绝对值的和#
for(r in 51:86){
summ<-rep(0,50)
#计算yy[r,]与前50组数据中的第i组数据的距离，记为summ[i]#
for(i in 1:50)
{
summ[i]<-0 for(j in 5:21)
{
```

```
summ[i]<-abs(yy[i,j]-yy[r,j])+summ[i]
}
summ[i]<-summ[i]+abs(yy[i,1]-yy[r,1])
}
#minorder存放与r行距离最近的三行的行号#
minorder<-order(summ)[1:3] #如果第r行2，3列是缺失值，则用与第r行最近的三行的2，3列的平均数来填补# if(is.na(yy[r,2])) {
z[r,2]<-(z[minorder[1],2]+z[minorder[2],2]+z[minorder[3],2])/3
z[r,3]<-(z[minorder[1],3]+z[minorder[2],3]+z[minorder[3],3])/3
}
#如果第r行4，5栏是缺失值，则用与第r行最近的三行的4，5列的平均数来填补#
if(is.na(yy[r,4])) {
z[r,4]<-(z[minorder[1],4]+z[minorder[2],4]+z[minorder[3],4])/3
z[r,5]<-(z[minorder[1],5]+z[minorder[2],5]+z[minorder[3],5])/3
}
}
yy<-z
#最后把交叉项加入到数据集中去#
yyy<-yy for(i in 1:21)
{
for(j in 1:21)
{
yyy[,39+(i-1)*21+j]<-yy[,i]*yy[,j]
}
}
#充交叉项和二次项，yyy是补充完成后的数据，共有480列，yyy的第（39+(i-1）*21+j)列是PPi和PPj的交叉项#
zz<-yy
zzz<-yyy
E<-rep(0,480)
std−rep(0,480)
for(i in 1:480)
```

```
{
E[i]<-mean(zzz[,i])
std[i]<-sqrt(mean((zzz[,i]-E[i])^2))
}
for(i in 1:480)
{
yyy[,i]<-(yyy[,i]-E[i])/std[i]
}
yy<-yyy[,1:39]
```

#最终得到两个标注化数据集yy和yyy.前者是一个86*39的矩阵，后者是一个86*480的矩阵#

参考文献

[1] Hastie, T., Tibshirani, R., & Friedman, J.(2008). The Elementts of Statistical Learning:Data mining, Inference, and Prediction. Springer Press

[2] Weisberg，S.(1985). Applied Linear Regression. John Wiley & Sons, Inc.

[3] Golub，G.，Heath，M.，& Wahba，G.(1979). Generalized Cross-Validation as Method for choosing a Good Ridge Parameter，*TECHNOMETRICS.*，VOL.21, No.2

[4] Hoerl，A. and Kennard，R.(1976)，Ridge regression: some simulations. *Comm. in Scientist.*,4.105-123

[5] Efron，B.，Johnstone，I.，Hastie，T. and Tibshirani，R.(2004)，Least Angle Regression，*Ann. Statist.*，32. 407-499.

[6] Varmuza, K. and Filzmoser P.(2008)，Comparison of some linear regression methods. *Austrian Research Promotion*，project no.812097/11126.

致　　谢

感谢国家、党和学校对我多年来的培养；感谢姚远教授对本次论文的精心指导；感谢姚远老师讨论班同学的帮助以及所营造的良好的学习和讨论氛围；感谢宝洁公司提供的数据；本文的大部分R代码是由沈致远、唐明宇和我探讨之下共同完成，在此，我对沈致远同学和唐明宇同学表示感谢；由于这是本人第一次使用LaTex写文章，行文过程中，樊楷同学和李康同学给予了很大帮助，对此表示感谢；最后感谢一下长期以来对我一直支持和关注的父母和朋友。

……