线性回归模型处理宝洁统计课题

1 背景和摘要：

洗衣粉去污作用是通过将其溶解于水形成水溶液,通过溶质的物理化学性质来去除污渍.因此测量洗衣产品水溶液的一些属性,利用这些属性的定量数据可以有望了解此产品去污的功效.我们小组经过对数据的观察和研究，寻求建立溶液属性和产品功效之间的回归模型，这样可以帮助了解产品水溶液的成分与去污能力的关系，以及根据这个关系结合化工技术知识找出最优的配方.我们小组通过联合使用多种线性回归模型的方法，对于每一个响应变量 O1~O18，考察已知数据 PP1~PP21 以及它们二次项和交互作用对这些响应变量的影响.由于添加了所有的二次项和交互作用后，总的预测变量维数变得很高，远远超过样本量.为了解决这个问题,我们使用了 Stepwise，主成分分析，Lars(Least Angle Regression)和 Lasso 这些方法来进行数据降维和变量选择，建立合适的预测模型.原数据集里面包含了缺失数据，为了充分地利用信息，我们使用多重填补的方法处理缺失数据，以尽可能减低缺失数据对于我们使用数据拟合模型回归的影响.

2 问题的提出和分析：

2.1 问题的提出

主办方提供了一个含有 86 个样品的样本，要求我们根据现有数据拟合出一个统计模型，这个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.可能是出于商业机密的考虑，数据的实际含义已经被隐藏，取而代之的是代码化的变量名称 PP1~PP21，O1~O18，在这种情况之下我们无法从数据的实际含义入手，只能单纯地根据数据的相互关系建立模型.其中样本中带有 32 个含缺失预测变量数据的样品,这占了样本量的 37.2%，这个比例比较大，若将含有缺失数据的样品随便放弃,则会造成信息的极大浪费，不能做出好的预测.为了能够准确地建立回归模型，我们应该尽可能地考虑各预测变量的相互作用，然而这样一来预测变量的总维数会急剧上升，当添加了全部的二次项与交叉项后，总维数上升至 252 维，远远超过了样本量 86.考虑到上述的两种情况，模型的建立要解决两个主要问题：(1)对缺失数据的处理以及(2)对预测变量的选择.

2.2 问题的分析

2.2.1 缺失值的处理

由于题目所给数据存在缺失，我们需要对数据进行合理的填补并使用填补后的数据拟合回归模型.这样要求填充的数据与完全数据比较接近，即用填补的数据的与观测到的数据服从相同的总体分布，使用经过填充后的全部数据进行模型拟合能得到比较准确的结果.我们对数据的分布进行猜测并检验，使用通过了检验的分布模型对数据进行填补.

2.2.2 线性模型的选用

由于所有的二次项和交叉项都作为可能选用的预测变量，这样就会使预测变量维数会变得很高，对于建立回归模型，需要进行变量选择.线性回归模型在变量选择上有着比较成熟的理论基础和实践经验，且有多种不同的方法可以选用；其他回归方法在预测变量选择上没有很好的方法.如果采用其他的回归方法，如神经网络等，在建立模型时可以仅选用 PP1~PP21 这 21 个变量而不加入二次项

和交互作用项.经过权衡，我们认为数据中已经包含充足的预测变量，如果能充分利用，线性模型可以提供比较准确的预测，于是就集中研究对所给的问题寻找合适的线性回归模型.

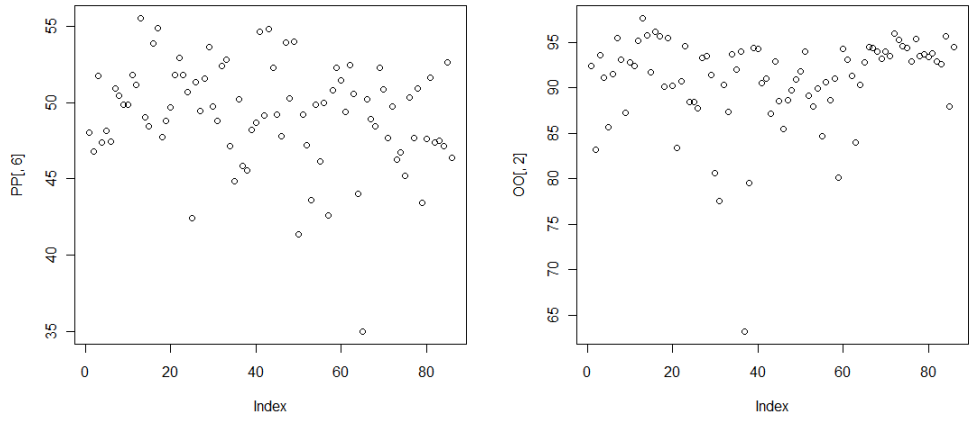
2.2.3 共线性处理和变量选择

我们首先在矩阵图中观察各个变量的分布以及相互关系，发现部分预测变量存在着较大的共线性关系.于是我们采用主成分分析的方法对数据进行初步的降维，这样能在不损失太多信息的前提下减少预测变量的个数.线性回归中经常使用 Stepwise 方法来选择预测变量，在实际中有着不错的效果.Lars 和 Lasso 因其惩罚项的设置而使得其估计的回归系数具有稀疏的性质，这个特点可以获得一个含预测变量并不是很多的回归模型，对于我们要解决的问题也是一个比较有用的工具.

3 方法：

3.1 异常值识别

在进行所有的数据处理之前，我们先看数据是否存在异常值.我们观察 PP1~PP21，O1~O18 的散点图，看是否有明显的离群值.散点图中可以看出 PP6 的第 65 个样品，PP8 的第 4 个样品，PP20 和 PP21 的第 38 个样品，O2 和 O7 的第 37 个样品，O9 的第 85 个样品，O15 的第 1 个样品在散点图上显示为比较明显的离群值.然而这些含有离群值的样品仅仅在某一两个分量上出现离群现象，在其他维度上其数据并未表现出异常情况.鉴于上述情况，我们不能认定上述离群值是错误数据，因而不能舍弃该数据.

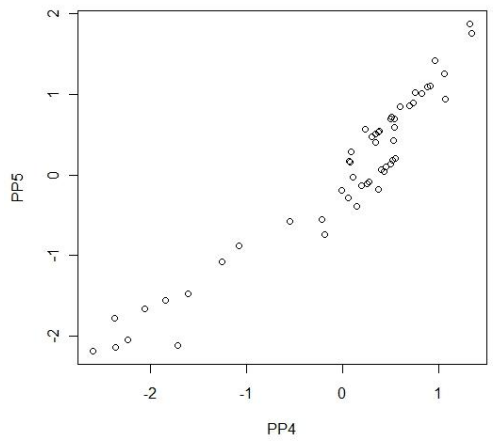


图表 1 部分包含疑似异常值的散点图

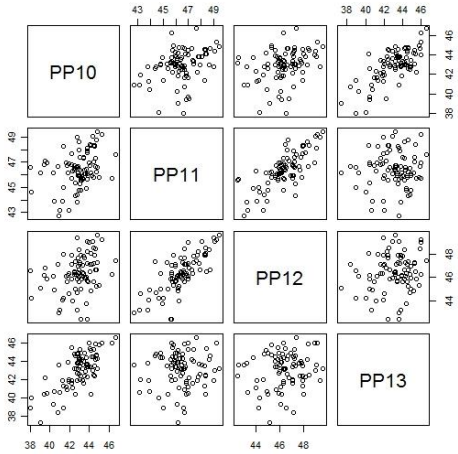
3.2 主成分分析作数据变换

通过观察自变量的矩阵图，我们发现有若干预测变量有着很强的共线性性:PP4 与 PP5 都是缺失了数据的变量，它们的缺失数据成对出现，仅观察其非缺失数据，可以发现其线性相关性很显著，相关系数达到 0.9527；PP10 至 PP13 这四个变量内部有较强的共线性性；PP14 至 PP21 这 8 个变量存在着多重共线性关系，其内部 PP14 与 PP15，PP16 与 PP17，PP18~PP21 这三个小组变量中能够发现明显的共线性关系.对这 14 个变量分为上述三组提取主成分，能够有效地提取有用信息，降低数据维数.

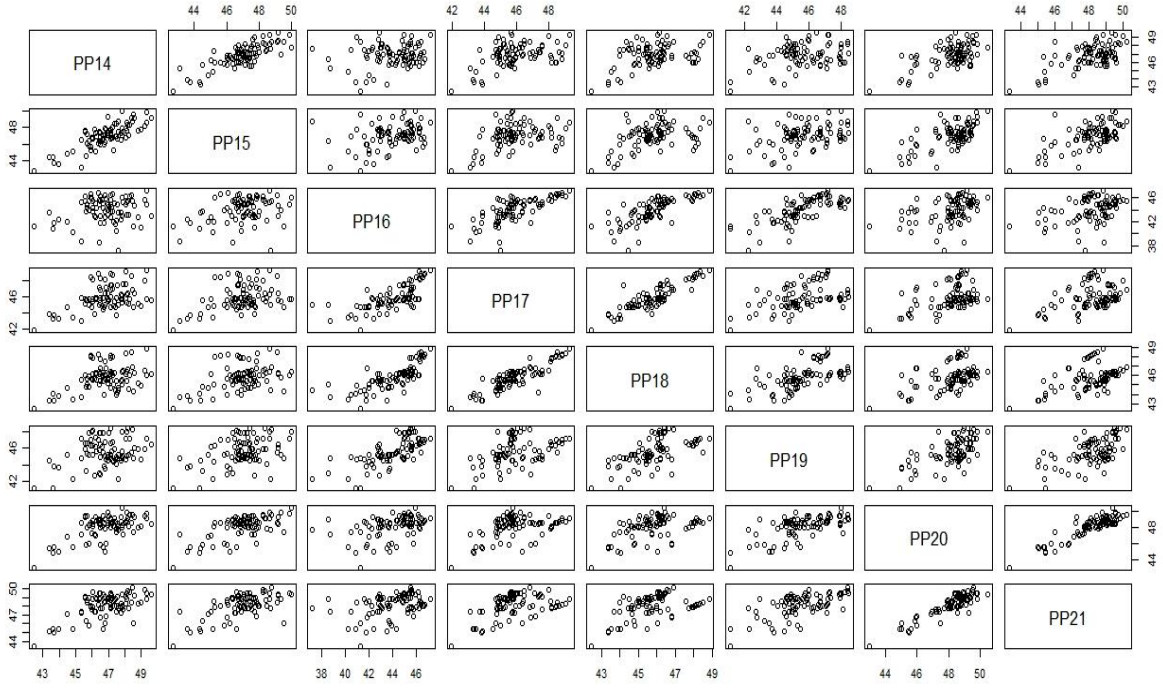
PP4，PP5 的第一主成分方差贡献为 0.9798，选用第一主成分代表原来两个变量；PP10~PP13 的前两个主成分的贡献率分别为 0.5546，0.3263,累计贡献率为 0.8809，选用前两个主成分代表原来的四个变量；PP14~PP21 的前三个主成分的贡献率分别为 0.6026，0.1795 和 0.0981，累计贡献率为 0.8802，因而选用前三主成分.经过提取主成分以后，数据变量从原来的 21 维降低至 13 维，包含所有一次项、二次项和交叉项的数据从原来的 252 维下降为后来的 104 维.



图表 2.1 PP4 与 PP5 非缺失数据的散点图



图表 2.2 PP10~PP13 的矩阵图



图表 2.3 PP14~PP21 的矩阵图

提取主成分之后的数据可以视为另一个用来的用来预测 O1~O18 的数据样本，我们对这个新的样本用其他线性回归方法进行系数估计.根据主成分的特性，在 PP1~PP21 里原来共线性较强的变量都变成相互正交的主成分，于是新的变量

间的共线性性会比原来的变量大为削弱.对于拟合线性模型，使用新的变量效果会比用原始的 PP1~PP21 有优势.我们对初始预测变量 PP1~PP21 以及提取主成分之后的数据这两种样本都进行模型拟合，比较用两者拟合的模型在交叉检验之下的误差,选择较优的模型.

3.3Stepwise(逐步回归)选择变量与回归

Stepwise 是线性回归中常用的变量选择方法，它属于一种贪心算法：每一步先尝试向子集模型添加一个新的预测变量，这个新加入的变量能够最大程度低改进前一步的子集模型，然后看能否从现模型中剔出删去后对子集模型影响不大的变量，不断重复直到不能再对模型中的变量添加或剔出，得出的子集模型就是结果.Stepwise 的优点是快速，添加和删除变量的计算和判断比较简单，在较短的时间内能选择出一个较优解.

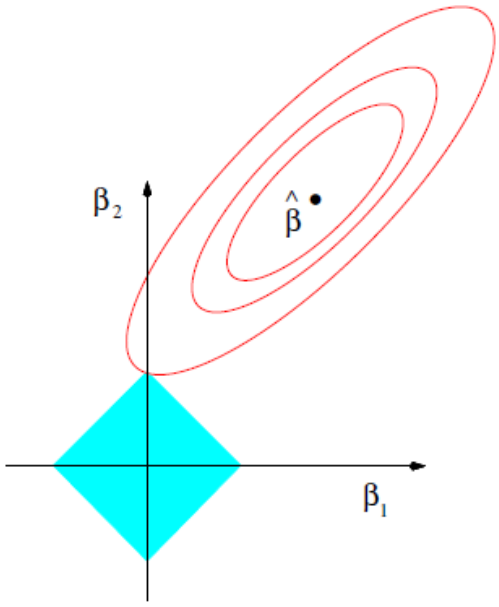
我们将使用 Stepwise 来选择变量的过程分为两个步骤.首先根据题目的提示，我们首先挑选一些可能对模型拟合有重要作用的二次项与交叉项.每一次将一个二次项或交叉项和所有一次项对响应变量进行回归，根据拟合结果挑选出使得残差平方和最小的 40 个.在第一步的选择中我们只使用没有缺失的样品数据.接下来，我们将这 40 个变量连同一次项合共 61 个变量使用 Stepwise 进行选择，并计算最优子集模型的回归系数.第二个步骤我们使用经过填补后的整个样本数据.我们使用软件 R 中的 step 函数进行 Stepwise 变量选择，step 函数使用 AIC 准则寻找最佳子集模型.

3.4Lars 和 Lasso

Lars 与 Lasso 是两种对线性模型加了限制的回归方法，Lars 对传统的 Stagewise 变量选择方法作了修改.对于一个预测变量，Stagewise 方法仅有"添加"或"不添加"这两种状态，相比之下 Lars 通过压缩回归系数的方式实现对预测变量的"部分添加"，这样对改善变量选择的结果比较有帮助.因此 Lars 方法被认为能够有效地选择出对线性模型有显著作用的预测变量[1].Lasso 则是通过在平方距离损失函数的基础上添加系数L_1范数惩罚项达到压缩系数的目的(公式 1).由于 Lasso 的L_1范数惩罚项特性，用 Lasso 得到的回归系数经常是一个稀疏的结果(图表 3)[2]，这样对于解决预测变量多于样本量的问题，Lasso 应该是比较有用的.在使用 Lasso 的时候我们注意到会存在着 bias-variance tradeoff，过分强调惩罚项可能会使得回归后的模型存在很大的误差.我们通过交叉检验的方法选择最优的回归系数估计.

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

公式 1 Lasso 的定义



图表 3, 可行域是原点关于L_1范数的邻域,回归系数通常会落在这个区域的尖角上,使得回归系数具有稀疏的性质

R 里面的程序包 Lars 中包含了使用 Lars，Lasso 作回归以及为 Lars 模型作交叉检验的函数，我们有关 Lars 和 Lasso 的计算都是直接使用或者修改这个程序包里面的函数而实现的.

3.4.1Lars_Lasso 拟合回归模型

首先使用 Lars 初步选择变量，再使用 Lasso 作系数估计，这样配合使用拟合回归模型.对每一个响应变量（O1~O18）使用 Lars 进行回归计算，将所有备选变量中最先加入到模型的 40 个变量作为第一步变量选择的结果.所有备选变量指包括了所有的一次项、二次项与交叉项的数据，对于初始数据 PP1~PP21 是总共 252 个变量，对于提取了主成分后的数据则是 104 个变量.在应用 Lars 作变量选择时，我们仅使用不含有缺失值的样品.接下来是用 Lasso 将选出来的 40 个变量对响应变量进行系数估计，通过交叉检验的效果来决定惩罚项系数λ，这样同时也在 Lasso 的解路径（solution path）中确定了回归系数.在应用 Lasso 作回归时我们采用经过填补后的全部数据.

虽然根据其定义，Lars 和 Lasso 看起来是有很大区别的，但是从其算法和 solution path 的关系看来，两者却有着紧密的联系， Lars 的算法只要稍作修改，即可成为 Lasso 的算法.这种相似性似乎意味着我们上述过程所分开的两个步骤可能其实是在重复做同一个事情.但是从结果上看来，事实并非如此.在很多的情形下，由 Lars 挑选出来最早进入模型（也意味着与响应变量最相关）的变量，Lasso 对其回归系数的估计是 0;通过 Lasso 估计得到非零回归系数的预测变量，他们在第一步用 Lars 选择变量时进入模型的顺序也不是完全连续的一片，而是比较无规则地分布在第一个到第四十个当中.

进入模型 / 项目 / 顺序	1	2	3	4	5	6	7	8
变量	PP1PP4	PP1PP12	PP13PP20	PP7PP13	PP1	PP6PP13	PP2	PP17PP18
回归系数	0	0	-3.893545	0	3.4012567	0	0	-4.264953

进入模型 / 项目 / 顺序	9	10	11	12	13	14	15	16
变量	PP3PP15	PP5PP7	PP13PP15	PP3	PP6PP7	PP6PP18	PP5PP13	PP13PP14
回归系数	-0.276304	0	0	3.5711655	-3.218719	0.7589089	-1.509367	0
进入模型 / 项目 / 顺序	17	18	19	20	21	22	23	24
变量	PP15PP16	PP11	PP19	PP6PP8	PP9PP21	PP18^2	PP3PP8	PP1PP11
回归系数	0	0	0	-0.169073	0.1363425	0	-1.7303	0
进入模型 / 项目 / 顺序	25	26	27	28	29	30	31	32
变量	PP8	PP10PP16	PP16PP17	PP3PP6	PP2PP5	PP5PP19	PP3PP17	PP14
回归系数	0	0	0	0	0.980471	0.0827092	0.150872	0
进入模型 / 项目 / 顺序	33	34	35	36	37	38	39	40
变量	PP10PP19	PP2PP16	PP7^2	PP9PP11	PP3PP19	PP5PP12	PP7PP12	PP13^2
回归系数	3.575602	0	0.923191	0	0	1.2031262	0	0

图表 4 使用 PP1~PP21 以及其二次项,交叉项对 O1 作 Lars_Lasso 选择变量以及系数估计的结果

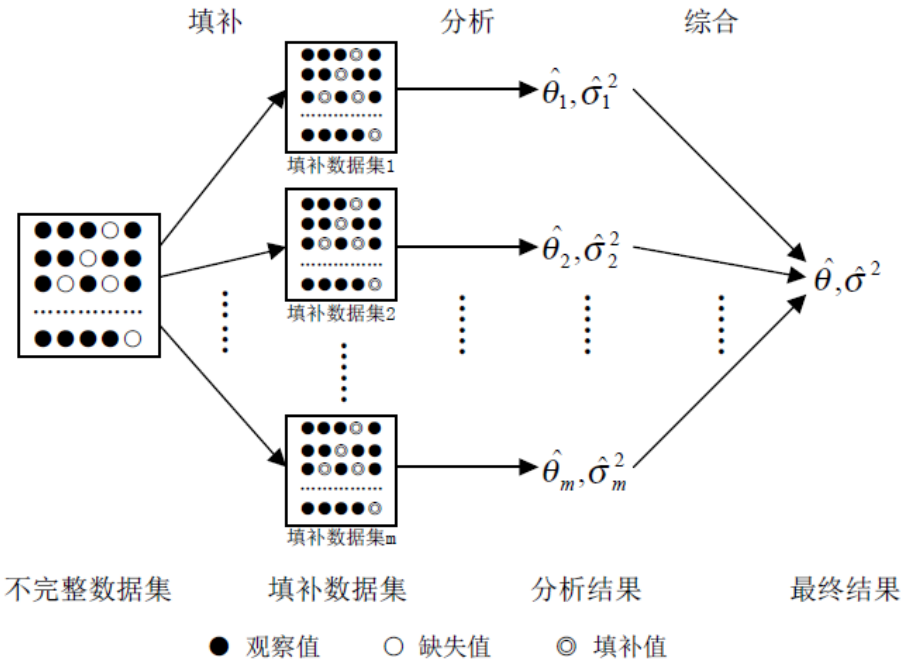
3.4.2Lars_LSE 拟合回归模型

直接使用 Lars 作回归拟合，其系数的估计并不是很准确，但是 Lars 可以作为变量选择的方法，其选择变量的效果与 Stagewise 相比有所改进[1].另一方面，如果实际模型确实为线性模型,则 LSE 是最佳线性无偏估计,有着许多优良的性质.根据这两点,我们综合使用两种方法.首先使用 Lars 做变量选择：将所有备选变量使用 Lars 进行回归，在交叉检验之下选出残差平房和最小的参数估计，将回归系数估计值非零的变量作为子集模型所包含的预测变量，然后对于这个子集模型用 LSE 重新估计回归系数.

3.5 多重填补(MI,Multiple Imputation)

针对数据中的缺失情况.我们使用多重填补的方法进行处理，多重填补是处理缺失数据的有效方法.

Rubin(1987)[3]证明了如果数据是随机缺失的(Missing at Random)，且进行填补和分析的模型是恰当的(proper)，那么使用多重填补进行数据处理得到的估计量将是渐进无偏的并且相应的置信区间是有效的(Statistically Valid).



图表 5 多重填补步骤及其推断原理(来自于http://stat.smmu.edu.cn/ARTICLE/sasinmed/sasandMI.pdf)

所有的数据都是连续型变量，我们使用推广的 Shapiro-Wilk 检验对原始数据的正态性进行检验，发现不能否定其正态性.于是假定所有的原始数据(X_{obs},X_{mis})服从联合正态分布$N_k(\mu,\Sigma)$，使用数据扩充算法对数据进行填充.(X_{obs}是观测数据，X_{mis}是缺失数据)

数据扩充算法是 Gibbs 采样器的一种，它使用迭代的方法对缺失数据和未知参数进行填补[4].每一次迭代中包含以下步骤：

1.从$N_k(\mu^t,\Sigma^t)$的条件分布中抽取缺失数据X_{mis}^t；

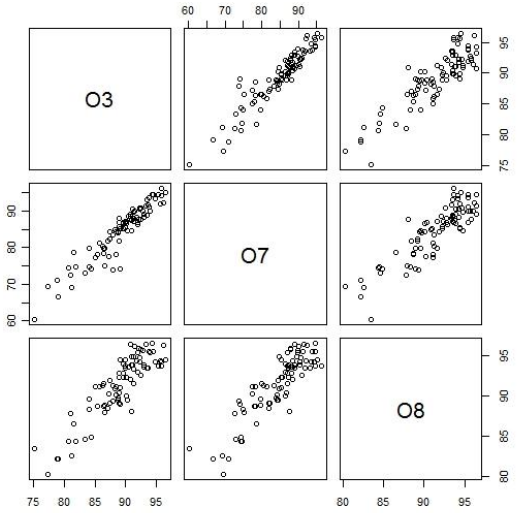
2.从(X_{obs},X_{mis}^t)中产生(μ^{t+1},Σ^{t+1});

3.重复迭代这两步直到填补的数据分布达到稳定;

我们独立的进行了 5 次数据扩充，获得了 5 个扩充的完整数据集.在这些数据集上都使用不同的回归方法，考察哪一种方法能够在这 5 个数据集上拟合的平均效果最好.选定了合适的拟合方法以后，对 5 个完整的数据集使用该方法进行回归，将得到的结果进行平均以获得最终结果.

3.6 其他方法——整组变量最小二乘

除了对预测变量,我们也观察了响应变量之间的关系.从 O1~O18 的散点图可以得知,一些响应变量之间也是具有比较强的线性相关性.例如(O3,O7,O8)两两之间就有着明显的线性相关性(图表 6.1,图表 6.2).



图表 6.2 O3，O7，O8 的矩阵图

	03	07	08
03	1	0.9390	0.8832
07	0.9390	1	0.8750
08	0.8832	0.8750	1

图表 6.1 O3，O7，O8 的相关阵

考虑以下参数模型 A：

$$y_1 = f(x, \theta) + \varepsilon_1$$

$$y_2 = \alpha_2 f(x, \theta) + \beta_2 + \varepsilon_2$$

$$\cdots\cdots$$

$$y_n = \alpha_n f(x, \theta) + \beta_n + \varepsilon_n$$

$\varepsilon_i (i = 1,2, \cdots\cdots, n)$相互独立，服从$N(0，\ \sigma^2)$，$f(x, \theta)$是已知参数模型.在我们的研究中沿用线性模型，就是$f(x, \theta) = x^T\theta + c$.对于模型 A，我们希望能够充分利用响应变量之间的相关性，很自然地，我们考虑最小化在这些相关的响应变量的残差平方和之和，即

$$\min \sum_{k=1}^{N} \sum_{i=1}^{n} (y_{i,k} - \alpha_i f(x_k, \theta) - \beta_i)^2 \qquad (*)$$

其中$(\alpha_1, \beta_1) = (1,0)$

(1)如果模型 A 的参数$(\alpha_2, \beta_2, \cdots\cdots, \alpha_n, \beta_n)$已知，$(*)$式中只需要估计$\theta$，即

$$\underset{\theta}{\operatorname{arcmin}} \sum_{k=1}^{N} \sum_{i=1}^{n} (y_{i,k} - \alpha_i f(x_k, \theta) - \beta_i)^2$$

这样可以化为一个普通的最小二乘解计算问题.

(2)但是如果模型 A 的参数$(\alpha_2, \beta_2, \dots\dots, \alpha_n, \beta_n)$未知，$(*)$式中除了需要估计$\theta$，还需要估计模型 A 的参数，即

$$\underset{\alpha_2, \beta_2, \dots\dots, \alpha_n, \beta_n, \theta}{\arcmin} \sum_{k=1}^{N} \sum_{i=1}^{n} (y_{i,k} - \alpha_i f(x_k, \theta) - \beta_i)^2 \qquad (I)$$

即使在$f(x_k, \theta)$是线性函数的假设下，直接处理(I)式是比较困难的.用求导数零点的方法会产生多元二次方程组，消元后出现高次方程.并且由于二次方程解的多值性，即使求出所有数值解，仍需要对所有的导数零点进行检验以选出(I)式的解.为此，我们想到了一个稍微妥协一点的办法，在响应变量之间做线性回归，用模型 A 参数的估计值替代真实值，即转为求解(II)式

$$\underset{\theta}{\arcmin} \sum_{k=1}^{N} \sum_{i=1}^{n} (y_{i,k} - \hat{\alpha}_i f(x_k, \theta) - \hat{\beta}_i)^2 \qquad (II)$$

$(\hat{\alpha}_i, \hat{\beta}_i)$是$y_i$对$y_1$的回归系数.经过这样的修改以后计算会变得简便，但是其偏差也会增大.

我们将这个整组处理的方法的结果与对每个响应变量单独作回归作比较，选择效果更好，更合适我们要处理的数据的方法.

4 结果

4.1 随机模拟的结果

4.1.1 整组变量最小二乘回归的效果

我们通过随机模拟来测试我们提出的方法对系数估计的准确程度.

我们用伪随机数生成机产生零均值，方差为 4 的样本量为 100 的相互独立预测变量$x_i (i = 1,2,3,4,5)$,然后如下生成响应变量$y_i (i = 1,2,3)$

$$y_1 = (x_1 + 3x_2 + 7x_3) + \varepsilon_1$$

$$y_2 = 2(x_1 + 3x_2 + 7x_3) + 4 + \varepsilon_2$$

$$y_3 = 5(x_1 + 3x_2 + 7x_3) + 3 + \varepsilon_3$$

$\varepsilon_i (i = 1,2,3)$相互独立，服从$N(0, \sigma^2)$.

我们用三种方法进行比较：(1)在 3.6 中已知模型 A 的参数

$(\alpha_2, \beta_2, \ldots\ldots, \alpha_n, \beta_n)$ 的情况，估计参数；(2) 在 3.6 中未知模型 A 的参数 $(\alpha_2, \beta_2, \ldots\ldots, \alpha_n, \beta_n)$的情况，用(II)式估计参数 ；(3)对每个$y_1$， y_2和y_3分别作最小二乘回归.我们考察不同信噪比下三种方法的效果，σ从 1 逐渐递增到 10. 在高信噪比之下，三种方法相差无几，都能做出比较准确的估计.但是随着信噪比的减小，三种方法的结果就会差别：方法(1)对系数的估计相当准确而且稳定；方法(3)对系数的估计也比较准确，但是偶尔会出现较大的偏差；方法(2)则稳定地存在较大的误差，其准确程度甚至比不上方法(2)，即对每个响应变量分别作最小二乘回归.很遗憾的是在我们的问题中，这些线性相关性很强的响应变量之间的真实关系并不清楚，即模型 A 的参数未知，从我们模拟的结果可知，对每个响应变量分别考虑已经是最优策略. (输出文本 test1.txt，test2.txt，test3.txt 分别记录了随机模拟过程中上面对应的三种方法估计系数的结果.)

4.1.2Lars_LSE 的效果

用和 4.1.2 中相同预测变量$x_i (i = 1,2,3,4,5)$以及因变量y_1，同样在不同的信噪比之下察看这个方法作变量选择、系数估计的效果.整体上看，用 Lars_LSE 对系数的估计比较准确.Lars 选择变量的效果没有随信噪比的下降而有明显变坏(整个过程信噪比从 236 下降至 2.36)，在信噪比接近 3 的时候依然可以保留正确的预测变量，剔除无关的预测变量，甚至作出完全正确的变量选择.于是我们对实际的洗衣粉溶液数据也采用 Lars_LSE 方法，并与其他方法比较，择其最优者. (输出文本 Lars_LSE.txt 记录了随机模拟过程中 Lars_LSE 的变量选择--回归拟合的结果.)

4.2 模型比较的标准

考虑到我们使用了比较多的回归方法，而且有时还先后联合使用多种方式进行变量选择和系数估计，对于最后模型自由度的估计比较困难；并且根据主办方的要求，他们的最终判断标准是要我们的预测对新的 10 个样品的平方距离尽可能小.因此，我们选用十折交叉检验(10-fold cross-validation)的效果作为比较我们所有回归模型的标准.十折交叉检验不仅是比较常用的检验方法，将样本分为 10 份以后，每一份含有的样品个数为 8 或 9，比较接近最终的检验样品数量 10.

我们填充缺失值得到了 5 个完整的数据集，将回归方法对每一个数据集都使用一次，计算其交叉检验下的误差(cv_err)，然后我们把 5 个 cv_err 求和，选择使其最小的回归模型作为结果.

4.3 各种模型在交叉检验之下的结果及比较

根据交叉检验获得的误差的平均值的大小(图表 7)，我们为每一个响应变量选定了合适的系数估计方法.O1,O4,O6,O7,O9,O10,O12 和 O14 使用 Stepwise 的效果最好;对于 O2,O3,O8,O13 和 O17，使用 Lars_Lasso 的效果最好;O5,O11,O16 以及 O18 则是用提取了主成分的数据，再进行 Lars_Lasso 方法的效果最好;最后 O15 是先以 Lars 选择变量，然后用最小二乘回归的效果最好.

cv_err 平均	01	02	03	04	05	06	07
stepwise	188.1203	18.01172	9.448084	108.0097	315.0749	176.7641	18.24483
Lars_Lasso	261.6194	14.47453	8.276216	138.2991	297.1828	299.6858	19.9139
prin_Lars_Lasso	274.6381	15.33043	9.30653	169.1199	290.0226	288.3497	27.52229
Lars_LSE	369.8098	15.82597	9.499236	222.6292	305.4745	338.1971	26.74241

prin_Lars_LSE	377.277	15.60126	9.841976	186.4071	304.6714	282.7691	27.69122

cv_err 平均	08	09	010	011	012	013	014
stepwise	6.642727	177.7814	140.6715	272.9709	178.2875	193.4196	96.22112
Lars_Lasso	6.451931	235.0725	212.8872	222.1262	247.2081	173.3307	134.9911
prin_Lars_Lasso	8.099151	229.7026	165.4967	215.7944	184.6614	181.5974	132.242
Lars_LSE	8.42806	271.9547	269.9262	275.1025	250.155	200.4752	125.1116
prin_Lars_LSE	8.152103	264.581	197.8964	237.642	191.777	215.3666	169.5413

cv_err 平均	015	016	017	018
stepwise	9.340332	26.02488	36.55292	35.04812
Lars_Lasso	7.233059	24.83422	21.81999	31.38749
prin_Lars_Lasso	7.548557	23.97691	26.83969	30.56954
Lars_LSE	7.199682	27.37437	25.00447	36.62078
prin_Lars_LSE	7.371947	25.84137	28.95863	38.15203

图表 7 各种模型在交叉检验之下的误差的平均值

4.4 对结果的讨论

	01	02	03	04	05	06	07	08	09
Variance	361.17	25.66	20.222	209.76	389.8	358.86	53.538	14.205	274.72
	010	011	012	013	014	015	016	017	018
Variance	345.43	395.43	401.73	408.16	326.04	9.5298	29.528	53.837	50.328

图表 8 各响应变量的样本方差

结合图表 7 和图表 8 可以看出，Stepwise 方法效果最好的响应变量 O1,O6,O7,O9,O10,O12 和 O14 的方差都比较大（普遍大于 300），在中心化以后数据的绝对值比较大，于是对于标准化的预测变量，要求回归系数的绝对值也要相应地较大.Lasso 存在对回归系数的L_1范数的惩罚(但是不包括常数项)，这种惩罚使得回归系数绝对值偏小，导致回归模型偏差较大.Lars 的计算方法与 Lasso 类似,因而也会出现这样的情况.这样也解释了为什么在这些响应变量中 Stepwise 在这些响应变量中(除了 O7)其准确程度表现出了明显的优势.相反地，使用 Lar_Lasso 或者先提取主成分，再使用 Lars_Lasso(prin_Lars_Lasso)能够取得最好效果的响应变量，即 O2,O3,O5,O8,O11,O13,O16,O17 和 O18，它们中的大多数方差都比较小，其中方差在 30 以下的有 4 个，方差处于 50 到 55 之间的有两个.此时出现了一个相反的效应，当响应变量的数值较小时，尽管 Lasso 对回归系数有L_1范数惩罚，但是此时系数的绝对值较小，惩罚项没有使得系数的估计偏差太大，此时 Lasso 减小模型方差的作用在 Bias-Variance tradeoff 中起了主要作用，减小了拟合模型的均方误差.

引用文献:

[1] Bradley Efron et al.(2006), *Least Angle Regression*

[2] Trevor Hastie et al., *Elements of. Statistical Learning, 2nd Edition,* page 71

[3] Rubin, D. B. (1987), *Multiple imputation for nonresponse in surveys*

[4] Roderick JA Little, Donald B. Rubin, *Statistical Analysis with Missing Data, 2nd Edition*

致谢

指导老师姚远对本文的选题和研究的方法提供了宝贵意见.同组成员王筑艺同学和付潇鹏同学分别完成了缺失数据的处理和使用 Lars_LSE 方法的进行回归拟合的工作，我们共同努力完成了对宝洁这个课题的研究.

附录 1：4.1.1 随机模拟 R 代码

```
x1<-1:100
x2<-round(runif(100,0,500))
x3<-round(rnorm(100,50,80))
x4<-rnorm(100,0,2)
x5<-rnorm(100,0,2)
x1<-2*scale(x1)[1:100]
x2<-2*scale(x2)[1:100]
x3<-2*scale(x3)[1:100]
x4<-2*scale(x4)[1:100]
x5<-2*scale(x5)[1:100]
for(i in seq(from=1,to=10,by=0.2)){
    y1<-x1+3*x2+7*x3+rnorm(100,sd=i)
    y2<-2*(x1+3*x2+7*x3)+4+rnorm(100,sd=i)
    y3<-5*(x1+3*x2+7*x3)+3+rnorm(100,sd=i)
    Y<-(y1+2*(y2-4)+5*(y3-3))/(1+2^2+5^2)
    myfit1<-lm(Y~x1+x2+x3+x4+x5)
    cat("sd =",i,"\n",file="test1.txt",append=(i!=1))
    sink("test1.txt",append=TRUE)
    print(myfit1$coef)
    print(myfit1$coef*2+c(4,0,0,0,0,0))
    print(myfit1$coef*5+c(3,0,0,0,0,0))
    cat("\\\\\n",file="test1.txt",append=TRUE)
    sink()
    fit1<-lm(y2~y1)
    fit2<-lm(y3~y1)
    Y.<-(y1+fit1$coef[2]*(y2-fit1$coef[1])+fit2$coef[2]*(y3-fit2$coef[1]))/(1+(fit1$
coef[2])^2+(fit2$coef[2])^2)
    myfit2<-lm(Y.~x1+x2+x3+x4+x5)
    cat("sd =",i,"\n",file="test2.txt",append=(i!=1))
    sink("test2.txt",append=TRUE)
    print(myfit2$coef)
    print(myfit2$coef*fit1$coef[2]+c(fit1$coef[1],0,0,0,0,0))
    print(myfit1$coef*fit2$coef[2]+c(fit2$coef[1],0,0,0,0,0))
    cat("\\\\\n",file="test2.txt",append=TRUE)
    sink()
    cat("sd =",i,"\n",file="test3.txt",append=(i!=1))
    sink("test3.txt",append=TRUE)
    print(lm(y1~x1+x2+x3+x4+x5)$coef)
    print(lm(y2~x1+x2+x3+x4+x5)$coef)
    print(lm(y3~x1+x2+x3+x4+x5)$coef)
    cat("\\\\\n",file="test3.txt",append=TRUE)
    sink()
}
```

附录 2：4.1.2 随机模拟 R 代码(需要加载 lars 程序包)

```
x1<-1:100
x2<-round(runif(100,0,500))
x3<-round(rnorm(100,50,80))
x4<-rnorm(100,0,2)
x5<-rnorm(100,0,2)
x1<-2*scale(x1)[1:100]
x2<-2*scale(x2)[1:100]
x3<-2*scale(x3)[1:100]
x4<-2*scale(x4)[1:100]
x5<-2*scale(x5)[1:100]
XX<-cbind(x1,x2,x3,x4,x5)
for(i in seq(from=1,to=10,by=0.2)){
    y1<-x1+3*x2+7*x3+rnorm(100,sd=i)
    cv.lars(XX,y1)->cv.Reg;
    lars(XX,y1)->Reg
    beta1<-predict.lars(Reg,type="coefficients",mode="fraction",s=cv.Reg$fraction[
cv.Reg$cv==min(cv.Reg$cv)])
    VAR<-seq(1,length(beta1$coef))[beta1$coef!=0]
    sink("lars_lm.txt",append=(i!=1))
    print(i)
    print(VAR)
    print(lm(y1~XX[,VAR]))
    sink()
}
```