基于多种回归模型研究的洗衣粉的物理属性对去污效果的影响

邵智轩

学号：*1400012141*[*]

(日期：2018 年 6 月 16 日)

洗涤溶液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，因此通过测量洗涤产品溶于水后的溶液的一些属性就可以了解产品去污的效果。如果能建立溶液属性和产品效果之间的模型，就可以找出能够最大化产品效果的溶液的属性，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。

首先，我们对数据进行初步分析，包括缺失情况、各变量两两之间的相关性等等。通过响应变量的特点，将它分为两大类，利用因子分析，提取代表这两大类的两个因子作为响应变量。这样就对响应变量实现了降维处理。

其次我们将数据分为训练集和测试集，并简要探讨了缺失数据的处理问题，并决定采用 knn 填补来解决这一问题。

建模部分，本文既包含 OLS 在内的普通线性回归方法，也包括 SVM，elastic net 等流行的统计学习方法，通过 10-fold CV 评价模型性能优劣；并尝试添加变量的二次项与交互项，看模型是否有改进。

最终我们用这些模型预测之前预留的前 10 个样本，通过 RMSE 评价模型预测能力的好坏。此外，通过预测较好的模型中变量的系数，我们也可以以此探究哪些变量，即洗衣粉的哪些物理属性对去污效能的影响最显著？

关键词：洗衣粉，建模，缺失数据，PCA，Cross-Validation，线性回归，lasso，elastic net，SVM，

[*] shaozhixuansh@pku.edu.cn; (86)13381350619

I. 问题重述

A. 问题研究的背景

洗涤溶液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，因此通过测量洗涤产品溶于水后的溶液的一些属性就可以了解产品去污的效果。如果能建立溶液属性和产品效果之间的模型，就可以找出能够最大化产品效果的溶液的属性，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。

B. 试验设计及试验数据

为了研究洗涤溶液的物理属性对去污效果的影响，我们分别测量了 86 个不同产品溶液的物理属性和它们的去污效果的数据。（附件一）

a. 现有 86 个产品的物理属性及效果数据，取前 10 个产品作为验证模型预测精度的数据，用剩下的 76 组数据来建立模型。

b. 每一个产品的 21 个属性作为输入变量 (PP1—PP21)。

c. 产品在 18 种污渍上的功效作为输出变量 (O1—O18)。

II. 问题分析与模型假设

A. 模型假设

a. 样本的序号是随机化的，前 10 个样本与剩余样本来源于同一个模型。

b. 数据的缺失是 MAR 的，即缺失与未观测到的变量无关。

c. 18 类污渍可以分为两大类，应该将响应变量降至 2 维。

d. 去污能力的主要部分由自变量的线性部分描述（或者有平方项或交互项）。

e. 自变量中只有部分对响应变量有影响，即模型需要进行变量选择。

B. 数据初步分析

1. 数据缺失情况

86 个案例的缺失情况如表I所示，有 54 个完整案例，32 个缺失案例，其中有缺失的为 PP2，PP3，PP4，PP5 这 4 个变量。

我们抽取的测试集中，没有变量有缺失。

表 I: 数据缺失情况[a]

	Treatment	PP1	PP6	...	O18	PP2	PP3	PP4	PP5	
54	1	1	1	...	1	1	1	1	1	0
2	1	1	1	...	1	0	0	1	1	2
26	1	1	1	...	1	1	1	0	0	2
4	1	1	1	...	1	0	0	0	0	4
	0	0	0	...	0	6	6	30	30	72

[a] 通过“mice::md.pattern()”查看

2. 变量之间的相关性

所有变量两两之间的相关系数如图 (1) 所示。自变量和响应变量很明显地分为两块，这说明许多自变量之间相关性很高（物理属性比较相近），而许多响应变量之间相关性也很高（污渍类别比较接近）；但从反对角块来看，大部分自变量与因变量之间的相关性并不强，这或许也暗示用含这些自变量的线性模型来预测去污效果，结果可能差强人意。

III. 数据预处理

A. 标准化

许多统计方法（如 PCA）都需要先将数据做标准化。所以我们一上来就先将数据做了标准化（使其均值为 0，标准差为 1）。

B. 响应变量的降维

数据提供了 18 个响应变量，分别为洗衣粉对不同污渍的去污效果。当然，我们可以逐一拟合模型，但未免使得整个问题过于复杂，而且有碍模型的解释性。所以考虑对 18 个响应变量做降维处理。

1. 相关矩阵图

响应变量的相关矩阵图如图 (2a) 所示，从图上很明显地看出两个块，即响应变量似乎聚为两大类。后续分析会印证我们的想法。

2. 主成分分析

对 18 个响应变量作主成分分析，结果如图 (3) 所示。（数据经过标准化处理）

从方差比例图 (3a) 上，前两个主成分分别占了 53.88%，22.79%（合计 76.7%）。双标图 (3b) 仍然指出响应变量分为两大类。

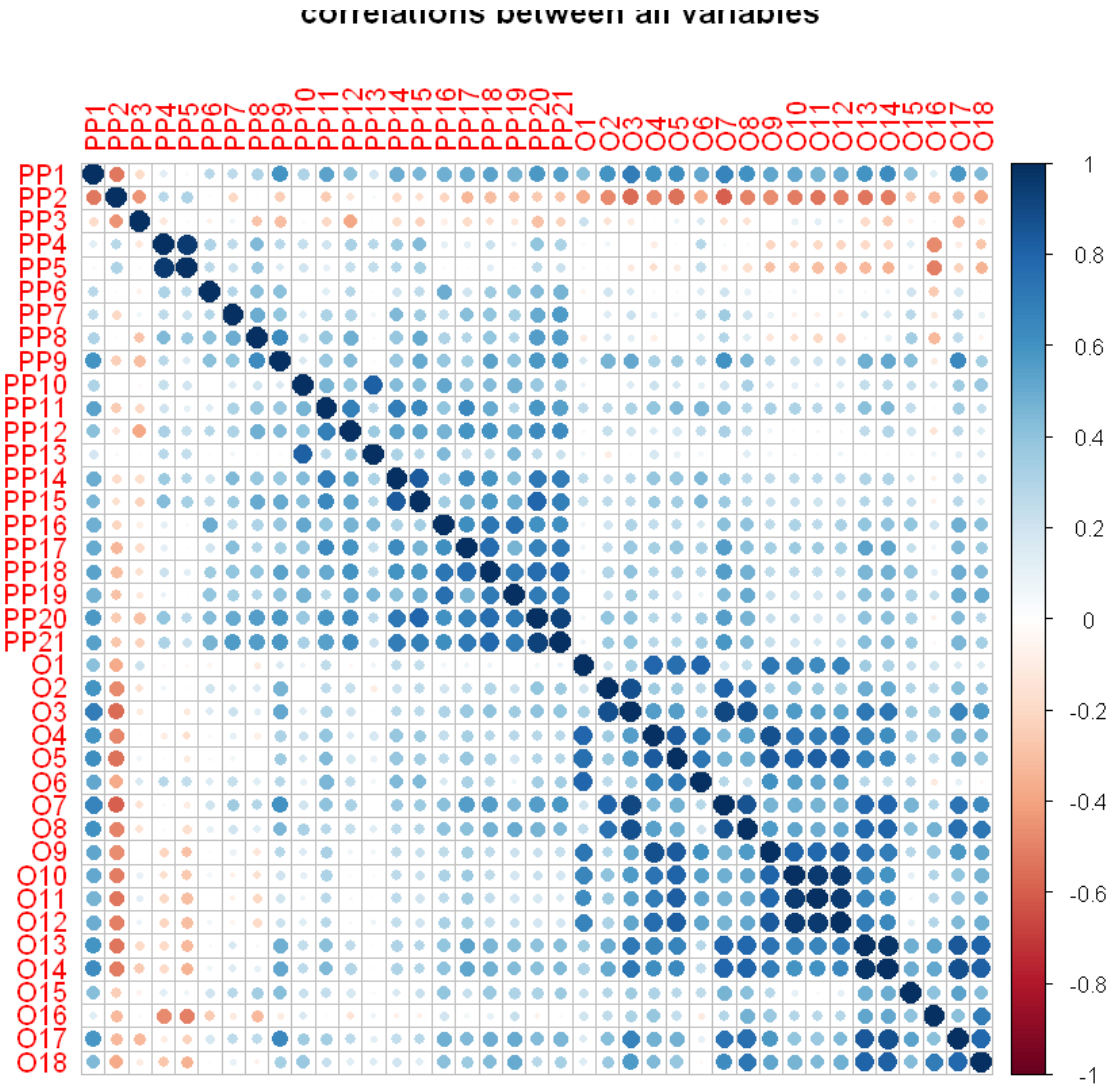


图 1： 所有变量之间的相关矩阵图[a]

[a] corrplot::corrplot()，由于自变量有缺失值，这里作图只使用完全数据，use = "complete"

3. 因子分析（*factor analysis*）

取因子个数为 2，因子分析结果如图 (4) 所示。

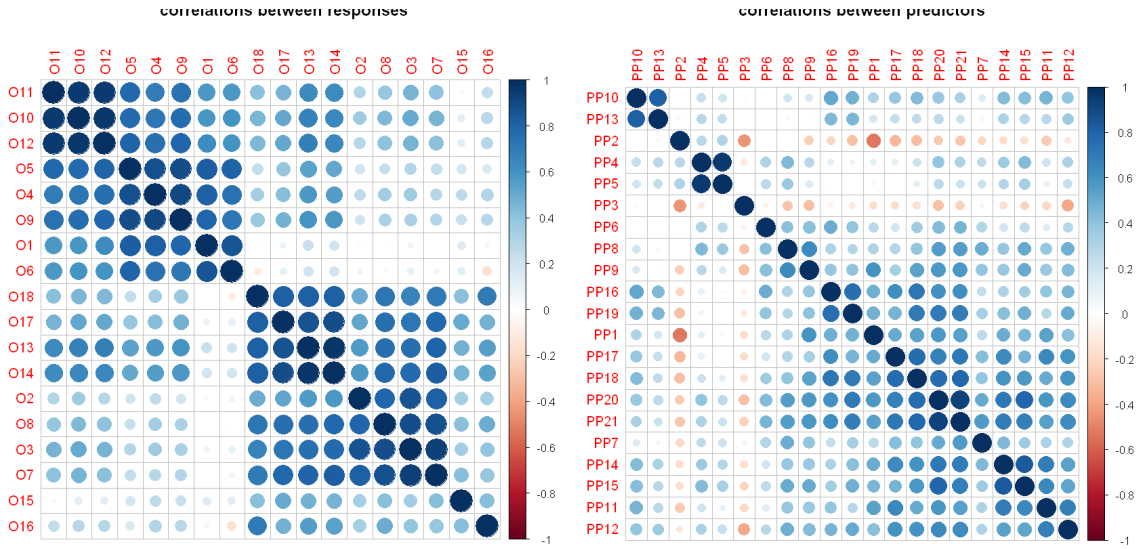
与主成分分析相似，因子分析仍然明显地将响应变量聚为两类，这分别对应着两类污渍（比如水溶，非水溶）：

污渍 1：[O2, O3, O7, O8, O13, O14, O15, O16, O17, O18]

污渍 2：[O1, O4, O5, O6, O9, O10, O11, O12]

计算给出前两个因子分别解释了 38.5% 和 34.5% 地方差（合计 73.0%）。

此外，在图 (4b) 上，我们还将有缺失的案例用红色标出，我们看到缺失似乎与 factor1 体现出一定的正相关，即我们有理由怀疑这不是完全随机缺失（MCAR）。



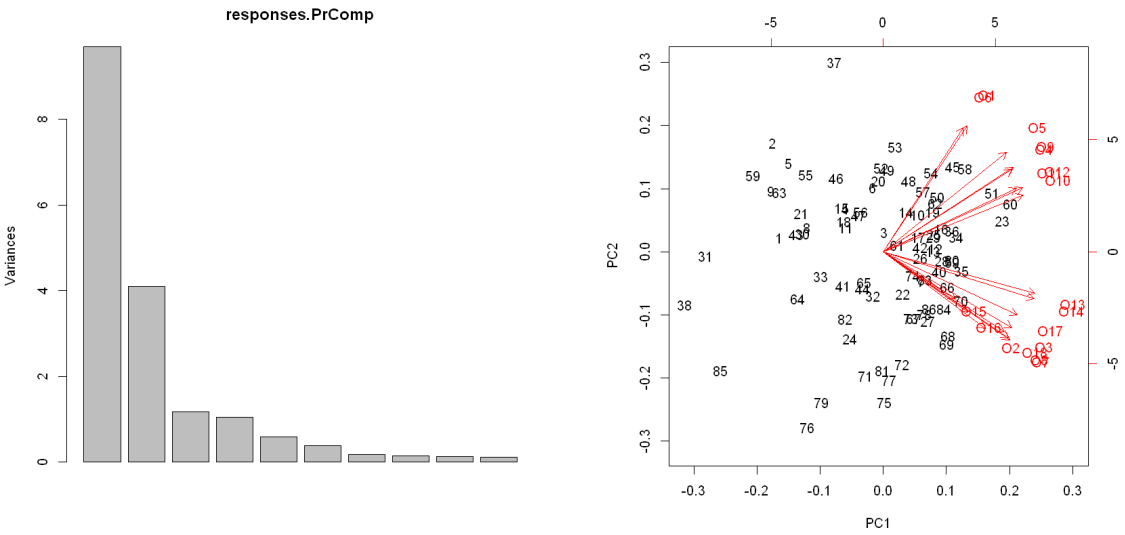
(a) 响应变量的相关矩阵图[a]

[a] 使用了全部数据，use = "all.obs"

(b) 自变量的相关矩阵图[a]

[a] 只使用完全数据，use = "complete"

图 2: 自变量和因变量的相关矩阵图



(a) 各主成分所占方差比例

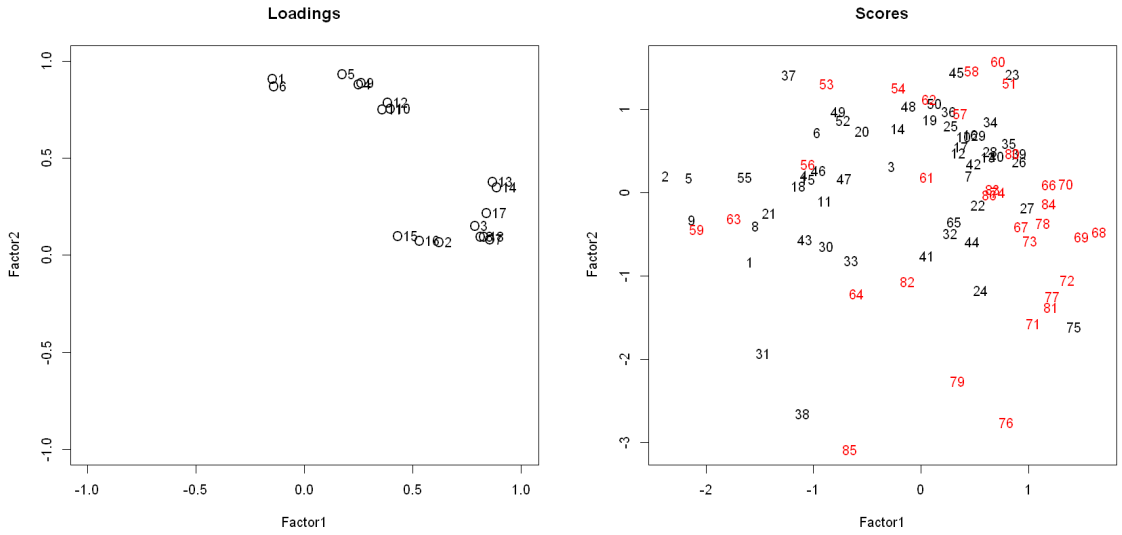
(b) 主成分分析的双标图

图 3: 对 *18* 个响应变量的主成分分析

4. 降维处理

通过以上分析，我们看到对 18 个响应变量取前两个主成分，或因子分析所得的 2 个因子变量，都是不错的选择，可以将响应变量个数降为 2 个。

在这两个方法中，笔者选择了因子分析，这是因为因子分析所取的变量方向，解



(a) 响应变量的因子载荷图　　**(b)** 样本因子得分

图 4: 对 *18* 个响应变量的因子分析（取因子个数为 *2*）

释起来更加直观。从图 (4a) 上我们看到，污渍 1 大多聚集在 Factor1 的正向，污渍 2 大多聚集在 Factor2 的正向，这样解释起来非常直观：基本可以粗略地认为，factor1 代表了对污渍 1 的去污效果，factor2 代表了对污渍 2 的去污效果。

C. 划分训练集与测试集

将数据标准化后，基于假设 (1)，选取前 10 个样本作为测试集。
这之后的建模全部基于训练集，最终通过测试集上的预测评价模型优劣。

D. 对自变量的初步分析

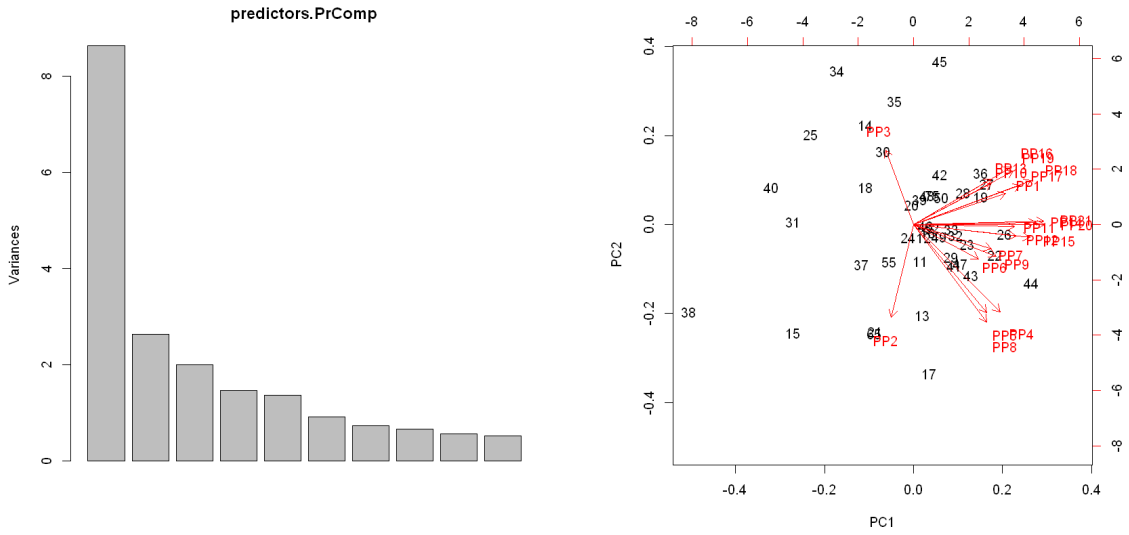
由于自变量中有缺失，先基于完全案例做分析。

1. 相关矩阵图

21 个自变量两两之间的相关系数如图 (2b) 所示。

2. 主成分分析

主成分分析的结果如图 (5) 所示。前五个主成分分别占了方差的：0.411, 0.125, 0.095, 0.070, 0.065。



(a) 各主成分所占方差比例　　**(b)** 主成分分析的双标图

图 5: 对 *21* 个自变量的主成分分析

表 II: 自变量的 *VIF*

	VIF
PP4	27.77
PP5	26.01
PP20	22.92
PP21	20.46
PP14	11.75
PP19	10.68
PP15	10.25

从相关矩阵图 (2b) 和双标图 (5b) 中我们都看到，与其他自变量相比，PP2 和 PP3 分别代表一种很不同（甚至负相关）的属性；PP4 和 PP5 很接近（相关系数非常高），与其他属性关系较小；很多属性比较相似。

3. 自变量共线性

在拟合模型之前，需要考虑自变量的共线性问题。可以看看设计矩阵的条件数 κ，以及共线性的指标“VIF”（方差膨胀因子）：

$$\kappa = 32.15$$

表II是 VIF 较高的几个变量：这也符合我们通过相关矩阵图 (2b) 得预期：颜色很深的非对角元的变量 VIF 往往很大。之后的模型拟合中，需要考虑多重共线性问题的处理，应进行降维、惩罚或变量选择。

E. 缺失数据填补

在进一步拟合模型之前，还有一个重要的问题——缺失数据的处理。将缺失数据所在行直接扔掉的做法是难以接受的：即使 MCAR（缺失与结果无关）的假设成立，在样本量已经如此小的情况下也会导致丢失太多信息。所以考虑对缺失数据作填补（imputation）。

众所周知，填补的方法不胜枚举，分别有不同的假设，适应不同的场合。这里主要考虑的方法有：

a. 均值填补（Unconditional mean imputation）：对于已经标准化的数据来说，即直接用 0 填补。这一做法的好处是简单方便，而缺点也是毫无疑问的：对缺失值的估计是有偏的。

b. 条件均值填补 (Conditional mean imputation)：假设所有变量为联合正态分布，对某一含有缺失值的变量 X_p，取完全案例对其余的变量做线性回归，用拟合得的模型对缺失值做预测，用预测值来填补。

c. knn 填补（k-nearest neighbors），算法大意是：假设第 i 个案例中第 p 个变量值 X_p 缺失，取 k 个离 i 的欧式距离最近的，且变量 X_p 非缺失的案例，填补值就取为这 k 个案例的 X_p 的均值。

在 MAR 缺失，以及变量为多元正态分布的假设下，理论证明条件均值填补给出无偏估计，所以它应该更合理。我们会在后续建模中分别采用这三个填补方法得到的数据，并采用预测效果好的那一组。我们会看到，条件均值填补的预测效果往往是最好的，knn 填补的效果也不错，而直接均值填补的效果相对较差。

IV. 预测模型的建立（只包含 21 个线性项）

A. 样本内的模型评价标准——10-fold CV

由于现阶段拟合模型**只用训练集的数据**，笔者采用 **10-fold cross-validation** 来评价预测能力。在建模以前，我将训练集（76 例）随机分成等大小（至多差 1 个）的 10 组，每次用其中的 9 组数据拟合模型，在剩余的 1 组上做预测，并与真值对比，偏差小的模型预测能力好。这里采用的指标为 RMSE （均方误差开根），越小模型表现越好。

对于有超参数的模型（如 lasso 的惩罚参数 λ，SVM 的 Cost 参数 C），根据 CV 的结果调整模型的超参数，使 RMSE 最小。这一调参过程在 R 程序中可以很方便的用 caret 库中的 train() 函数实现。

为公正地比较各模型，交叉验证的分组在每个模型下都是相同的。以下每个模型我们都利用III E提到的三种方法填补得到的数据，最终比较哪个填补方法产生的预测效果最好。

B. 最小二乘模型

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e} \tag{1}$$

$$(\boldsymbol{e}|X) \sim \mathrm{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}) \tag{2}$$

其中 $\boldsymbol{X}$ 包含全部 21 个自变量和 1 个截距项。

C. 逐步回归（Stepwise Regression）

本质上是选择一个最好的变量子集（Best Subset），由于 p 很大时，枚举法选择最佳子集的复杂度是 2^p。逐步回归是一种 Best-Subset Selection 的计算上可行的方法，它是一种贪心算法，并不保证选到最优解。在每一步，在现有的变量子集中添加或删去一个变量，使得模型的 AIC（等价于 C_p 统计量）最小；也可以选择其他信息准则，如 BIC。

以上算法在 R 程序中很方便地通过 step() 函数实现。

D. Lasso[3] 和 Elastic Net[4]

这一部分属于 **Shrinkage Methods** 或称为 **Regularized Estimation Methods**。这里选取的两个方法（不同于 Ridge Regression 等）不仅压缩回归系数，还将一些系数严格严格缩减为 0，这样即可以实现变量选择。这主要源于其惩罚项在零点不可微。

在IV B中提到的普通最小二乘回归，其损失函数为残差平方和，而 Lasso 加上了对系数的 L1-norm 的惩罚项：

$$\hat{\boldsymbol{\beta}}_{\lambda}^{lasso} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2n} \sum (y_i - \boldsymbol{x}_i' \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{3}$$

有理论证明，lasso 等价于对系数假设了一个均值为 0，参数为 $1/\lambda$ 的 Laplace 先验分布。

对 lasso 的一个改进是 Elastic Net，其惩罚项的形式为：

$$p_{\lambda}^{enet} = \lambda \sum_{j=1}^{p} (\alpha \beta_j^2 + (1-\alpha)|\beta_j|) \tag{4}$$

显然，这是 ridge 和 lasso 的一个“折中版本”，它既可以像 lasso 一样做变量选择，也可以像 ridge 一样对共线性项进行同步的压缩。

lasso 和 elastic net 在 R 程序中都可以通过 enet 库来实现。

E. 偏最小二乘回归（Partial Least Square）

与主成分回归（Principle Components Regression）类似，偏最小二乘回归构造一系列相互正交的 $\boldsymbol{X}$ 的线性组合 $\boldsymbol{X}\alpha$，但与 PCR 不同的是，这一过程利用到了响应变量 $\boldsymbol{y}$。具体的算法详见 [5]

不同于主成分回归寻找 $\boldsymbol{X}$ 的线性组合单使得其与之前的线性组合无关且方差最大，偏最小二乘还要求这一线性组合与 $\boldsymbol{y}$ 高度相关。详细的理论解释参考 [6]。

F. SVM

SVM（Support Vector Machines）是一类有力的，高度灵活的建模方法，起初应用于分类问题中，后来也拓展到回归问题中。它具有以下几个的优点：

a. 稳健性：残差在一个界限 ϵ 以下的数据点对损失函数贡献为 0，大于 ϵ 的项对 Loss 贡献为线性的，所以 outliers 对回归方程的影响不像最小二乘法中那么大。

   换句话说，模型中拟合得好的点在模型中不起作用，决定模型预测的是那些残差超过 ϵ 的数据点，称为支持向量（Support Vectors）。因为以上原因，该方法又称为“ϵ-insensitive regression”。

b. 核函数（Kernel）：对样本 $\boldsymbol{x}_i$ 的预测

$$\hat{y}_i = f(\boldsymbol{u}) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{u}) \tag{5}$$

   可以通过核函数 $K(\cdot)$ 来引入非线性，这样如果真实的模型中有非线性因素，非线性的核函数往往能更好地拟合模型。

笔者分别采用了 Linear Kernel：

$$K(\boldsymbol{x}_i, \boldsymbol{u}) = \boldsymbol{x}_i' \boldsymbol{u} \tag{6}$$

以及 Radial Kernel：

$$K(\boldsymbol{x}_i, \boldsymbol{u}) = \exp(-\sigma \|\boldsymbol{x} - \boldsymbol{u}\|^2) \tag{7}$$

并比较使用两个核函数的 SVM 模型的预测能力。

V. 模型求解与计算

下面取响应变量因子分析得到的 factor1 作为 $\boldsymbol{y}$；之后用类似的方法分析 factor2。

factor1 在训练集内的样本标准差为 0.8901，可供参考。

注：下面给出的结果为条件均值填补的数据的分析结果，其他填补方法的预测结果会在VI给出。

表 III: *方差分析表：模型 1 只含有 5 个再模型 2 中显著的变量，模型 2 包含全部 21 个自变量*

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
70	24.44319	NA	NA	NA	NA
54	17.01346	16	7.429737	1.473854	0.1443544

A. 普通最小二乘

最小二乘的 10-fold CV RMSE 为 0.6578，并不令人满意。

在含有全部 21 个自变量的模型在训练集（$n = 76$）上的拟合结果中，大部分系数并不显著，如果我们选取其中显著的几个自变量：PP2，PP3，PP9，PP10，PP13 重新拟合一个模型（记为 OLS_restricted），CV-RMSE 降为 0.5867：

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03734    0.06944   0.538    0.592
PP2         -0.61588    0.08841  -6.966 1.44e-09 ***
PP3         -0.55709    0.09384  -5.936 1.01e-07 ***
PP9          0.12629    0.08016   1.575    0.120
PP10         0.45000    0.10380   4.335 4.78e-05 ***
PP13        -0.26903    0.10425  -2.580    0.012 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5909 on 70 degrees of freedom
Multiple R-squared:  0.5886,Adjusted R-squared:  0.5592
F-statistic: 20.03 on 5 and 70 DF,  p-value: 2.409e-12
```

对以上两个模型做方差分析（ANOVA），结果表明没有两个模型没有显著的差别，见表III。

B. 逐步回归

以最小 AIC 进行逐步回归，得到的 RMSE= 0.6476；以最小 BIC 进行逐步回归，得到的 RMSE= 0.6645。

我们看到，逐步回归选出的模型，其预测效果并不比普通最小二乘回归强。

表 IV: *lasso 回归的系数中绝对值最大的 6 个，fraction=0.544*

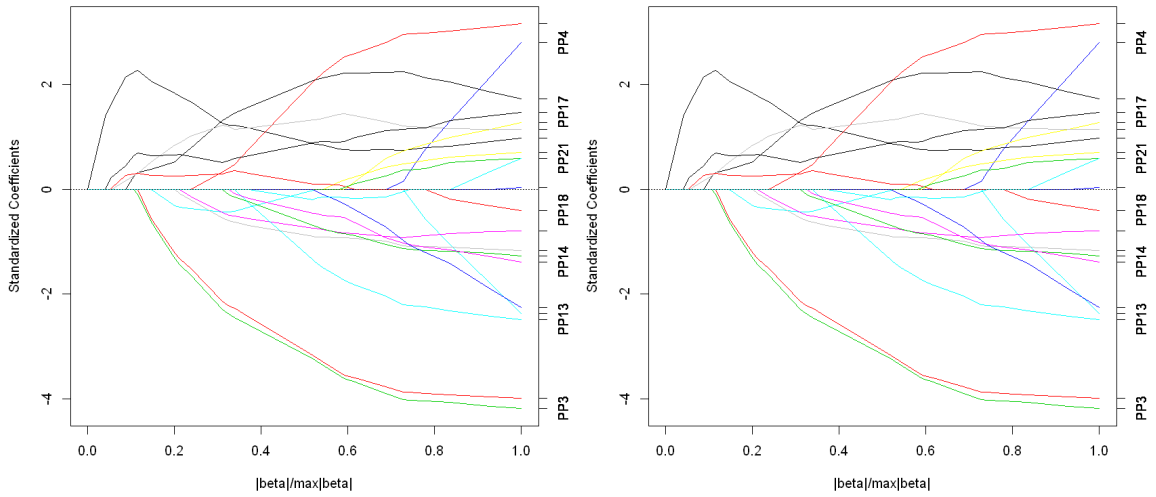
PP3	-0.3977
PP2	-0.3788
PP9	0.2601
PP10	0.2554
PP13	-0.1715
PP16	0.1575

C. Lasso 和 Elastic Net

1. *lasso*

根据 10-fold CV 选出的 fraction=0.5444（定义为 $\|\hat{\beta}_\lambda^L\|_1/\|\hat{\beta}\|$），RMSE=0.6118。我们看到，由于 lasso 有变量筛选功能，他的预测效果比完全模型的线性回归要好，但是仍不如我们手动选取的子集的预测效果。

在 21 个自变量中，lasso 将 6 个严格设为 0。我们可以看一下 lasso 回归的系数中绝对值最大的几个（由于自变量都归一化，所以系数的大小可比），见表IV。或许并不意外，lasso 选出的变量与 OLS 选出的显著变量几乎是完全相同的。



(a) *lasso 的变量选择*　　**(b)** *elastic net 的变量选择*

lasso 和 elastic net 的变量选择机制如图 (6a) 和 (6b)。

2. *Elastic Net*

Elastic Net 给出的结果，无论是变量选择，还是模型系数 $\hat{\beta}^{enet}$ 都与 lasso 非常类似，就不在此列举了。它给出的 RMSE 为 0.6115

表 V: 通过 *10-fold CV RMSE* 比较各模型的预测能力

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
OLS	0.4245	0.5606	0.6383	0.6578	0.7554	1.0080	0
OLS_mean	0.4801	0.5416	0.6432	0.6931	0.8325	1.0360	0
OLS_knn	0.4973	0.5362	0.6238	0.6777	0.7807	1.0040	0
OLS.restricted	0.3115	0.4192	0.5976	0.5867	0.7260	0.9356	0
lmStepAIC	0.3949	0.5870	0.6264	0.6476	0.6745	1.0080	0
lmStepBIC	0.3870	0.6013	0.7014	0.6645	0.7308	1.0240	0
svmLinear	0.3933	0.5001	0.6226	0.6343	0.7592	0.9078	0
svmRadial	0.3619	0.5497	0.6041	0.6087	0.6389	0.9385	0
svmRadial_mean	0.3616	0.5962	0.6203	0.6398	0.6694	1.0000	0
svmRadial_knn	0.3651	0.5194	0.6299	0.6199	0.6700	0.9584	0
lasso	0.3790	0.5219	0.6052	0.6118	0.6986	0.9280	0
pls	0.3513	0.5131	0.5623	0.6170	0.7319	0.9749	0
enet	0.3796	0.5222	0.6036	0.6115	0.6987	0.9296	0
enet_mean	0.3856	0.4883	0.6194	0.6288	0.7078	0.9463	0
enet_knn	0.3764	0.5040	0.6123	0.6174	0.6888	0.9118	0

D. 偏最小二乘

偏最小二乘通过 10-fold CV 选出了 ncomp=3，即用前三个线性组合做回归，给出的 RMSE 为 0.6170。

E. SVM

1. *Linear Kernel*

10-fold CV 选出的 $C = 0.02$，RMSE=0.6343。带 Linear Kernel 的 SVM 在此数据中似乎并不比其他方法有优势。

2. *Radial Kernel*

10-fold CV 选出的 $C = 0.02$，$\sigma = 0.0308$，RMSE=0.6087。它在 CV 上的预测表现优于比其他方法，但仍不如我们选出的子集上的线性回归。

F. 10-fold CV 结果总结

通过 10-fold CV RMSE 比较各模型的预测能力，如表V所示。我们看到我们在选取的变量子集上的最小二乘回归，表现是最好的。

另外，我们也看到在三种填补方法中，knn 和回归填补法表现都不错，而直接均值填补由于有较大偏差，表现较差。

表 VI: 测试集上的预测

id	ytrue	OLS	OLS_mean	OLS_knn	OLS.restricted	pls	lasso	enet
1	-1.5855122	-1.235586547	-1.2702002	-1.33806520	-1.2735787	-1.26228901	-1.0794214	-1.0855943
2	-2.3714088	-1.150952322	-1.0650698	-1.11977439	-1.0020097	-1.08654185	-1.0878594	-1.0935166
3	-0.2747798	0.001373642	0.1986472	0.23418961	0.5773681	0.03459549	0.3132947	0.3164747
4	-1.0845762	-1.090128847	-0.9983714	-1.07490514	-1.9429371	-1.00317991	-0.9603596	-0.9660995
5	-2.1562664	-2.210344863	-2.1558682	-2.18362914	-2.6699820	-2.14980363	-1.9677296	-1.9784179
6	-0.9666176	0.003946920	-0.0435570	-0.06115095	-0.5695284	-0.10905434	-0.2429749	-0.2455679
7	0.4459457	0.577643914	0.6286399	0.76970469	0.7959296	0.71360882	0.7914424	0.7923066
8	-1.5368518	0.411712896	0.5612333	0.55505744	0.2084053	0.60938670	0.6076258	0.6090758
9	-2.1313065	-0.941475656	-0.9143817	-0.92239325	-0.8548945	-0.59622142	-0.5532731	-0.5574353
10	0.3943648	0.570936897	0.5388808	0.52756880	1.0405667	0.64252885	0.7463014	0.7421322

表 VII: *factor1* 测试集 *RMSE* 对比（样本标准差为 *1.02942*）

OLS	0.888498574010219
OLS_knn	0.931004784256062
OLS_mean	0.939504003531406
svmRadial	0.949728593254381
OLS.restricted	0.952485959846832
pls	0.984337636871385
enet	1.00603535571914
lasso	1.00778440569384
svmRadial_mean	1.01721789658544
svmRadial_knn	1.01815496024335
svmLinear	1.09452607343546

G. 测试集上的预测

测试集的样本标准差为 1.02942。在样本集上的预测值如表VI，RMSE 如表VII。

我们看到第 8 个案例在所有方法下都预测反了，如果我们剔除这一例，RMSE 如表VIII：

出乎意料的是，在测试集上的预测与 CV 结果差异很大；虽然预测结果都很差，普通最小二乘方法给出了最接近真实值的预测，CV 表现好的 svmRadial 反而在测试集上表现最差。

当然，测试集的选取（作业要求的前 10 个数据作为测试集）是有随机性的，通过 Cross Validation 数据我们也看到，不同的 fold 下，RMSE 差异非常大。

H. 对 factor2 的预测

factor2 的训练集样本标准差为 0.9153，测试集的样本标准差为 0.4830。最小二乘中的系数显著项为 PP9，PP11 和 PP13。

各模型的计算过程与 factor1 完全类似。在测试集上的 RMSE 如表IX所示。响应

表 VIII: *factor1* 测试集 *RMSE* 对比（剔除第 *8* 个数据）

OLS	0.674733926652801
OLS_knn	0.690540440692826
OLS_mean	0.701167007630739
pls	0.751507093025621
enet	0.782876900008961
lasso	0.78581237882019
OLS.restricted	0.818289044552123
svmRadial	0.852869860317964
svmRadial_knn	0.928511456151497
svmLinear	0.96231367621451
svmRadial_mean	0.975247790456759

表 IX: *factor2* 测试集 *RMSE* 对比（样本标准差为 *0.4830*）

svmLinear	0.44742744377387
svmRadial	0.498684304643301
svmRadial_knn	0.498749548902298
svmRadial_mean	0.534421413978976
pls	0.682598273440063
enet	0.692149791161709
lasso	0.69401776152888
OLS	0.736472916650397
OLS.restricted	0.776687086453052
OLS_mean	0.84208621968754
OLS_knn	0.910802613883242

变量 factor2 很大程度上不能被预测值解释。

I. 总结

无论是对 factor1 还是 factor2，我们单纯使用线性项做预测的结果都是令人失望的——响应变量有很大一部分没有被解释。这也说明我们的假设4，即“响应变量绝大部分可以由自变量的线性项解释”是靠不住的。我们考虑在模型中加入非线性项——平方项和交互项。

VI. 模型的改进——加入非线性项

通过V G和V H中的数据汇总，我们看到，单纯使用自变量的线性项并不能对洗衣粉的去污功效做出很好的预测。下面我们考虑在模型中加入平方项和交互项。

可能的交互项和平方项非常多，平方项 21 个，交互项多达 $\binom{21}{2} = 210$ 个，总共的变量个数多达 252 个。变量选择将成为一个关键的问题。

表 X: *factor1*（样本标准差为 *1.0294*）测试集 *RMSE* 对比，其中有“*aug*”模型中加入了平方项和交互项，没有“*aug*”的只有线性项[a]

pls.aug	0.7630	pls.aug	0.5503
svmLinear.aug	0.7979	svmLinear.aug	0.5880
lasso.aug	0.8327	lasso.aug	0.5985
OLS	0.8885	OLS	0.6747
svmRadial	0.9497	pls	0.7515
pls	0.9843	enet	0.7829
svmRadial.aug	0.9965	lasso	0.7858
enet	1.006	svmRadial	0.8528
lasso	1.008	svmRadial.aug	0.8938
svmLinear	1.095	svmLinear	0.9623

[a] 右侧两列的 RMSE 剔除了第 8 组数据

A. 加入交互项后对 factor1 的预测结果

我们先不考虑变量选择问题，尝试直接将新的含 252 个变量的设计矩阵“喂”给之前提到的算法，得到的结果见表X。

表X的结果令人惊讶，虽然我们暴力地将所有交互项、平方项都直接不加选择地“喂”给这些算法，几乎所有方法的预测效果竟然都有较大的提升。值得注意的是，原本就带有非线性成分地 SVM with Radial Kernel 不升反降，预测能力远不如其他方法。这说明这一 Kernel 在这一数据下并不适用。相反，偏最小二乘法、SVM with Linear Kernel 这些原本只含 X 的线性项的方法有显著提升，这说明在真实模型中或许确实存在起重要作用的平方项或交互项，加入模型后会对预测能力有显著提升，我们会在VI A 2里看到这一点。

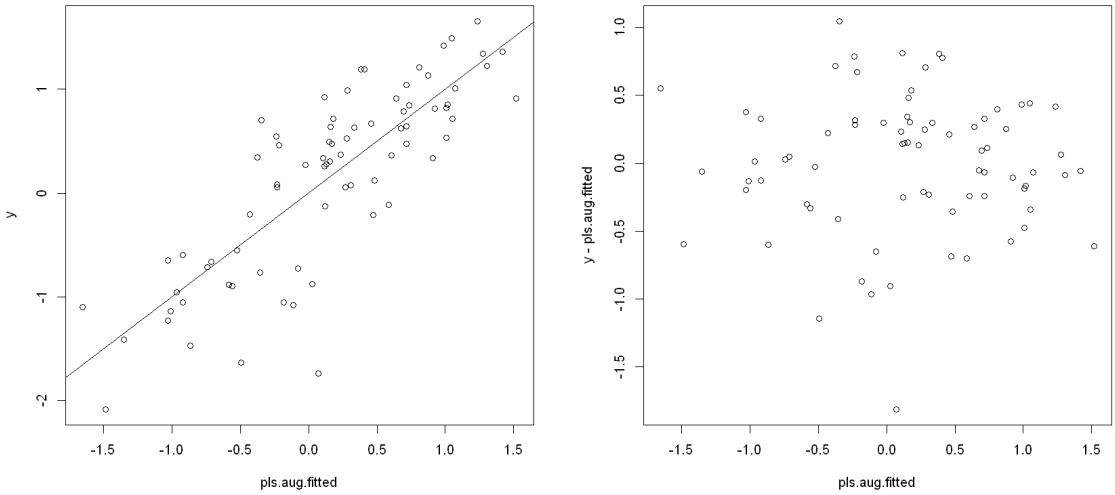
1. 残差诊断

每一个模型拟合完，都应做残差诊断；笔者以预测效果最好的偏最小二乘为例，作残差诊断，如图 (7) 所示。残差图中并没有看出明显的问题。

2. lasso 的变量选择结果

通过 lasso 的变量选择功能，由于自变量的都是归一化的，我们可以根据系数绝对值大小看哪些变量对响应变量有较大的影响，见表XI。lasso 的变量选择结果，当 fraction=0.03（CV 的结果）时，基本只留下了交互项和平方项。

这也指出，单个物理属性很大程度上难以解释污渍 1 的去污效果，而交互项，即多个物理属性的综合效应，会对污渍 1 的去污效果有较大的影响。



(a) y-$\hat{y}$，图中直线为 $y = \hat{y}$，$\mathrm{cor}(y, \hat{y}) = 0.8226$

(b) $\hat{e}$-$\hat{y}$，残差图接近 *null-plot*

图 7: 含平方项和非线性项的偏最小二乘回归，训练集内的 *RMSE* 为 *0.5028*

表 XI: 含平方项和交互项的 *lasso* 回归的系数中绝对值最大的 *10* 个，*fraction=0.03*

PP_3_8	-0.4402
PP_4_15	-0.2838
PP_1_2	0.2497
PP_11_17	0.1962
PP_8_10	0.1697
PP_12_15	-0.1610
PP_11_16	0.1559
PP9_sqrd	0.1507
PP_9_14	-0.1473
PP_1_17	0.1410
PP_1_6	-0.1167

B. factor2 的结果

对于代表第二类污渍的 factor2，我们用上一节的方法，加入平方项和交互项，新的 RMSE 如表XII所示。

不同于 factor1，加入平方向和交互项后的预测效果没有明显提升，有些方法效果反而下降了。

这说明，不同于 factor1，交互项在 factor2 中起的作用并不大。以现有的自变量数据，我们对第二类污渍 factor2 的预测难以令人满意。

表 XII: *factor2*（样本标准差为 *0.4830*）测试集 *RMSE* 对比，其中有“*aug*”模型中加入了平方项和交互项，没有“*aug*”的只有线性项

svmLinear	0.4474
svmRadial.aug	0.4927
svmRadial	0.4987
pls	0.6826
enet	0.6921
lasso	0.6940
OLS	0.7365
lasso.aug	0.7962
pls.aug	0.9672
svmLinear.aug	0.9998

VII. 模型优缺点分析与改进方案

模型对 factor1 的预测能力尚可，对 factor2 的预测效果却不好。这或许也无可奈何——变量太多而样本量太小了（76 个），而真正对响应变量有显著影响的变量应该很少，所以这是一个很大的挑战。

但如果能在拟合模型前通过更有效的方法进行变量选择，尤其是从 210 个交互项中选出有用的变量，应该能更好地还原出真实的模型，达到更好的预测效果。

此外，在变量选择后并拟合模型后，如何对系数做出合理的统计推断？在 [6] 中提到，变量选择后得模型会夸大显著性（Subset Selection Overstates Significance）。

VIII. 总结

本文尝试了统计学习中多种回归问题的方法，尝试通过洗衣粉的 21 个物理属性，预测由因子分析得到的代表两类污渍去污效果的响应变量。

对 factor1 的预测效果在加入交互项和平方项后有了显著提升，而对 factor2 的预测效果始终不理想，进一步的改进重点在于如何更有效地进行变量选择。

[1] Kuhn M, Johnson K. Applied predictive modeling[M]. New York: Springer, 2013.

[2] Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties[J]. Journal of the American statistical Association, 2001, 96(456): 1348-1360.

[3] Tibshirani R. Regression shrinkage and selection via the lasso[J]. Journal of the Royal Statistical Society. Series B (Methodological), 1996: 267-288.

[4] Zou H, Hastie T. Regularization and variable selection via the elastic net[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2005, 67(2): 301-320.

[5] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning[M]. New York: Springer series in statistics, 2001.

[6] Weisberg S. Applied linear regression[M]. John Wiley & Sons, 2005.