数据科学导论

1 问题定义

原问题描述：假设 (Y, X) 服从线性模型:

$$Y = X\beta + \epsilon, \epsilon \sim (0, \sigma^2 I_N)$$

其中，Y 为 p 维响应变量，X 为 N 行 p 列的协变量，β 为 p 维未知参数。考虑：现在我们收集到了 $\{X_i\}_{i=1}^N$，而相应的 $\{Y_i\}_{i=1}^N$ 由于成本约束，只允许给定其中一些 X_i 的情况下，获得 n 个响应观测，即 $\{Y_l^*, X_l^*\}_{l=1}^n$，$n \ll N$.

目标：估计 β，且最小化 $MSE(\hat{\beta})$.

为了方便建立模型，我们加强了对误差的假设，认为误差服从正态分布：$\epsilon \sim N(0, \sigma^2 I_N)$. 此外，我们假设 X 各列之间不存在线性关系。

2 数据集介绍

为了令数据集能较好地符合我们的问题假设，我们采用人造数据。我们用 R 生成了 500 行 4 列的数作为 X，并根据我们假设的 β 和正态分布的 ϵ 生成对应的 500 行 y。

```
N <- 500
#设定 各分量值
b1 <- 1
b2 <- 2
b3 <- 3
b4 <- 4
b <- c(0,b1, b2, b3, b4)
#随意生成x各分量值
x1 <- c(rnorm(N / 2, mean = 10, sd = 4), rnorm(N / 2, mean = 20, sd
    = 5))
x2 <- c(rnorm(N / 2, mean = 20, sd = 7), rnorm(N / 2, mean = 30, sd
    = 2))
x3 <- c(runif(N / 2, 0, 50), rnorm(N / 2, mean = 25, sd = 10))
x4 <- c(rnorm(N / 2, mean = 30, sd = 1), rnorm(N / 2, mean = 0))
X <- t(cbind(x1, x2, x3, x4))
```

```r
#误差服从正态分布
e <- rnorm(N, sd = 1)
y <- b1 * x1 + b2 * x2 + b3 * x3 + b4 * x4 + e
```

我们生成了 X 各列的直方图，分别如下：

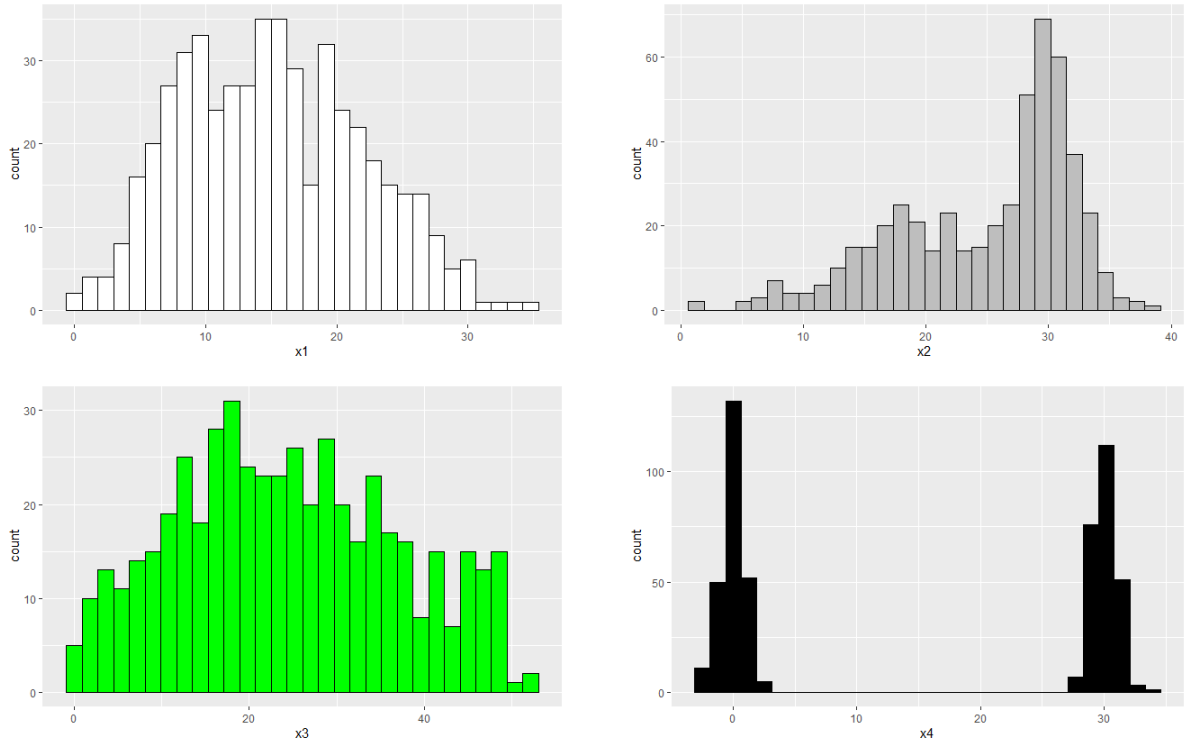

图 1: x_i 的分布

3 模型的设计与实现

3.1 原理介绍

我们选取的数据集的各分量没有依赖关系，复共线性很弱，因此可以采用最小二乘法进行参数估计。我们希望找到一种抽样方法，使得 $MSE(\hat{\beta})$ 最小化。

不妨将抽取的 n 个数据记为 X，其中 X 有 n 行 p 列. 首先，将 X 进行中心化、标准化后，可以证明 [1]:

$$MSE(\hat{\beta}) = \sigma^2 \sum_i \frac{1}{\lambda_i} \tag{$\star$}$$

因此，我们的目标变成最小化 ($\star$)。

又由于 X 已经过中心化、标准化，可计算得 $(X'X)_{ii} = n-1$, 故 $tr(X'X) = p(n-1)$，即 $\sum_i \lambda_i = p(n-1)$，这是一个定值。因此，当 λ_i 各分量相同时，$\sum_i \frac{1}{\lambda_i}$ 取得最小值。又由于 λ_i 度量了第 i 个主

成分取值变动大小 [2], 因此，若希望 λ_i 各分量相同，应使抽取的 X 在各分量方向上分布是相同的。所以，我们的抽样目标是使得样本分布成球对称。

3.2 模型设计

针对上述分析，我们提出了以下三种方法：

- 为直接最小化 ($\star$) 式，提出方法 1：重复进行 m 次简单随机抽样，选取其中令 ($\star$) 结果最小的 X.
- 基于球对称抽样的理念，由于难以直接实现球对称抽样，我们提出了下面两种简单的近似抽样方法。由图一可以看出，我们数据集的 x_1, x_2, x_3 在中部都分布得较为均匀，且 range 相近，因此若剔除分布在两端的数据，剩余数据可认为符合“各分量分布相同”的预期。基于上述分析，我们提出了方法 2：去除各分量中处于“异常”位置的数据，在剩余数据中进行简单随机抽样。虽然 x_4 的分布不符合上述预期，但后续结果表明该方法还是合理的。
- 为了使方法更具有普适性，而不依赖于数据集的选取，我们提出了方法 3：先计算出 X 的经验密度函数，不妨记为 $f(X)$. 选定一个合适的半径 r，对每个样本 x，若 $\|x\|_2^2 < r^2$，其被取样的概率设定为 $c/f(x)$，c 为一常数；否则，为 0。理论上，这样的采样能较为均匀的分布在半径为 r 的球内。但由于经验密度函数难以计算，我们没有对方法 3 进行实验。

3.3 模型的实现

- 简单随机抽样

  我们的模型结果要和简单随机抽样的结果进行比较，我们先写出了简单随机抽样的 R 语言代码：

```
N <- 500
n <- 50
n1 <- sample(1:100, size = n, replace = FALSE)
x1_1 <- x1[n1]
x2_1 <- x2[n1]
x3_1 <- x3[n1]
x4_1 <- x4[n1]
y_1 <- y[n1]
model_1 <- lm(y_1 ~ x1_1 + x2_1 + x3_1 + x4_1)
mse1 <- sum((model_1$coefficients - b) ^ 2)
```

- 方法 1

  我们随机生成 20 组 X，选取 $\sum 1/\lambda$ 最小的 x：

```
#使用提出的方法1, 数值近似解
best_verse <- Inf    #最小的特征值的倒数和
best_n <- numeric(n)#最小的特征值的倒数和对应的采样index
for (i in 1:20) {    #循环20次, 寻找最小倒数和
```

```
    n2 <- sample(1:N, size = n, replace = TRUE)
    x1_2 <- x1[n2]
    x2_2 <- x2[n2]
    x3_2 <- x3[n2]
    x4_2 <- x4[n2]
    y_2 <- y[n2]
    X_2 <- cbind(x1_2, x2_2,x3_2,x4_2)
    X_2 <- scale(X_2, center = T, scale = T)
    value <- eigen(t(X_2) %*% X_2)$value
    verse_value <- sum(1 / value)
    if (verse_value <= best_verse) {
      best_n <- n2
      best_verse <- verse_value
    }
  }
  n2 <- best_n
  y_2 <- y[n2]
  x1_2 <- x1[n2]
  x2_2 <- x2[n2]
  x3_2 <- x3[n2]
  x4_2 <- x4[n2]
  model_2 <- lm(y_2~x1_2+x2_2+x3_2+x4_2)
  mse <- sum((model_2$coefficients-b)^2)
```

- 方法 2

  去除“异常”数据，随机抽样：

```
mse3 <- numeric(itr) #1000次实现方法2
#首先，找出X各分量的分位数
summary(X)
##        x1                  x2                  x3                  x4
##  Min.   : 0.3179   Min.   :-2.299   Min.   :-2.754   Min.   :-2.7197
##  1st Qu.: 9.7081   1st Qu.:19.738   1st Qu.:15.790   1st Qu.: 0.1084
##  Median :14.6301   Median :27.961   Median :24.575   Median :14.7027
##  Mean   :15.1724   Mean   :25.135   Mean   :24.618   Mean   :15.0078
##  3rd Qu.:20.0607   3rd Qu.:30.560   3rd Qu.:33.610   3rd Qu
```

```
      .:30.0199
## Max.   :34.2037   Max.   :38.796   Max.   :51.126   Max.
      :32.4785
#根据summary的分位数进行赋值
Q11 <- 9.77180        # x1四分之一分位数
Q13 <- 20.05183       # x1四分之三分位数
IQR1 <- Q13 - Q11
lower_bound1 <- Q11 - 1.5 * IQR1  # 计算判定x1下界
upper_bound1 <- Q13 + 1.5 * IQR1  # 计算判定x1上界

Q21 <- 19.747         # x2四分之一分位数
Q23 <- 30.016         # x2四分之三分位数
IQR2 <- Q23 - Q21
lower_bound2 <- Q21 - 1.5 * IQR2  # 计算判定x2下界
upper_bound2 <- Q23 + 1.5 * IQR2  # 计算判定x2上界

Q31 <- 17.00534       # x3四分之一分位数
Q33 <- 33.65391       # x3四分之三分位数
IQR3 <- Q33 - Q31
lower_bound3 <- Q31 - 1.5 * IQR3  # 计算判定x3下界
upper_bound3 <- Q33 + 1.5 * IQR3  # 计算判定x3上界

Q41 <- -0.02544       # x4四分之一分位数
Q43 <- 30.15599       # x4四分之三分位数
IQR4 <- Q43 - Q41
lower_bound4 <- Q41 - 1.5 * IQR4  # 计算判定x4下界
upper_bound4 <- Q43 + 1.5 * IQR4  # 计算判定x4上界
#筛选出在两个分位数之间的X和X对应的y
Xandy<-cbind(X,y)
Xandy<-subset(Xandy, Xandy[,1] >= lower_bound1 & Xandy[,1] <=
    upper_bound1)
Xandy<-subset(Xandy, Xandy[,2] >= lower_bound2 & Xandy[,2] <=
    upper_bound2)
Xandy<-subset(Xandy, Xandy[,3] >= lower_bound3 & Xandy[,3] <=
    upper_bound3)
Xandy<-subset(Xandy, Xandy[,4] >= lower_bound4 & Xandy[,4] <=
    upper_bound4)

len <- length(Xandy[, 1])
```

```
n3 <- sample(1:len, size = n, replace = FALSE)
x1_3 <- Xandy[n3, 1]
x2_3 <- Xandy[n3, 2]
x3_3 <- Xandy[n3, 3]
x4_3 <- Xandy[n3, 4]
y_3 <- Xandy[n3, 5]
model_3 <- lm(y_3 ~ x1_3 + x2_3 + x3_3 + x4_3)
mse3 <- sum((model_3$coefficients - b) ^ 2)
```

4 实验结果评价

由于单次实验的不确定性较大，我们将以上代码中简单随机抽样、方法 1 和方法 2 分别运行 1000 次，绘制它们 1000 次实现的 MSE 图像进行比较。结果如图 2 所示。

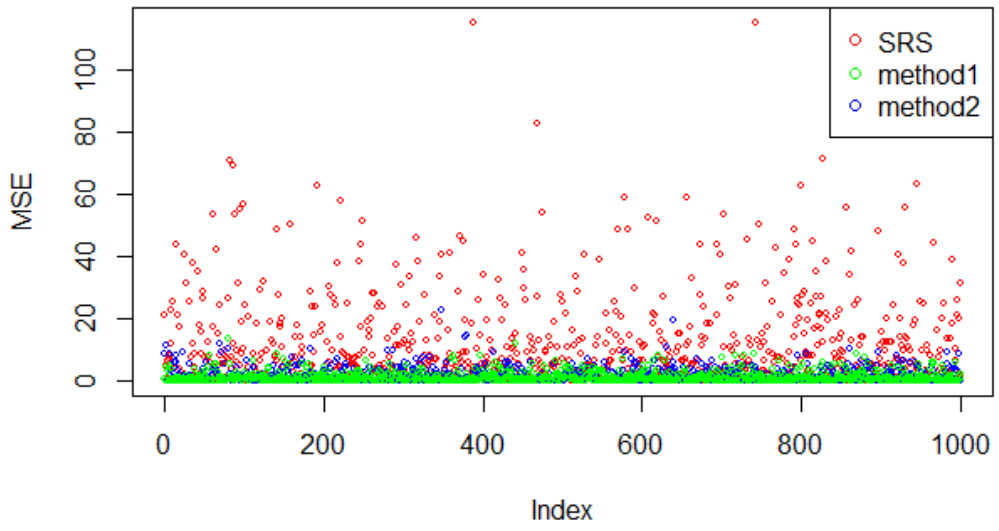

图 2: method1,method2,SRS 比较

如图，红色表示每次模拟下简单随机抽样得到 $\hat{\beta}$ 的 MSE，绿色表示方法 1 得到 $\hat{\beta}$ 的 MSE，蓝色表示方法 2 得到 $\hat{\beta}$ 的 MSE。可以看出，方法 1 和方法 2 在结果上要明显优于简单随机抽样，而方法 1 略好于方法 2.

我们又分别计算了每个方法 1000 次的 $\hat{\beta}$ 的 MSE 取均值，结果如下：

同样容易看出简单随机抽样的均方误差比方法 1 和方法 2 都大。

经过进一步分析发现，随机取样的 MSE 大主要源于截距项 β_0 的误差过大。为了突出 $\beta_i(i = 1, 2, .., p)$ 的估计好坏，我们去掉 β_0, 对 β 剩余分量重新计算了 MSE。

简单随机抽样 (mse1)	10.01089
方法 1(mse2)	1.229089
方法 2(mse3)	1.642618

表 1: method1,method2,SRS 比较

同样，我们绘制三种方法 1000 次实现的 MSE 图像并且分别计算 MSE 的均值进行比较。

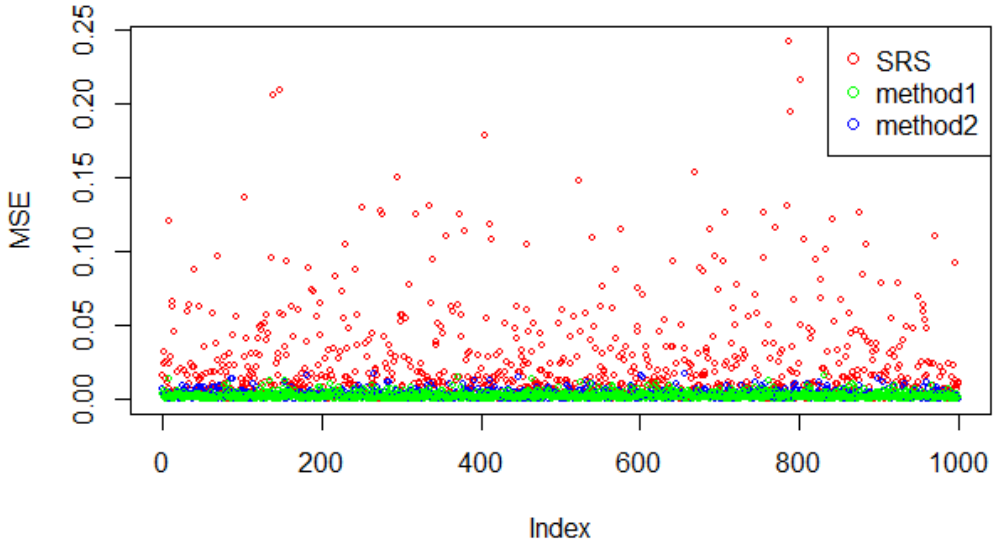

图 3: method1,method2,SRS 去掉截距项后比较

简单随机抽样 (mse1)	0.02188753
方法 1(mse2)	0.002423139
方法 2(mse3)	0.002885307

表 2: method1,method2,SRS 去掉截距项后比较

由图可见每个方法的 MSE 均显著缩小，但是大部分情况方法 1 和方法 2 估计的 $\hat{\beta}$ 的 MSE 都要小于简单随机抽样下的的 $\hat{\beta}$ 的 MSE。并且通过计算三者 MSE 均值，可见方法 1 和方法 2 的 MSE 远小于简单随机抽样，所以，我们认为方法 1 和方法 2 比简单随机抽样更优。

而对于方法 1 和方法 2，效果差别不大。虽然方法 1 看上去略比方法 2 好一些，但是方法 1 计算复杂度是方法 2 计算复杂度的 20 倍，因此，在这个数据集上，认为方法 2 要好于方法 1.

5 展望

- 如果 p 很大，但是 β 很多分量是 0，我们做出以下分析与调整：
  1）若 p 很大以至于 $p > n$，显然 $(\star)$ 的结论不再适用。需要先随机取少量样本，计算 Person corralation，将小于 0.2 的分量去除。用剩余分量，按照原先的方法进行计算。
  2）若 p 很大，但仍然满足 $p > n$。此时我们有两个选择；一是同上种情况，先取少量样本，判断非零分量；二是直接用原方法，选择一组最优 x，再计算 Pearson corralation，去掉较小分量再拟合。若 n 本身就很小（如本实验，n=50），抽取小样本不足以计算出适宜的相关系数，因此宜采用第二种方法。

- 如果 y 是分类变量，我们做出以下分析与调整：
  此时，分类问题变为 $Y = sgn(X\beta + \epsilon)$，我们需要更改参数估计的方法。可以考虑支持向量机、logistic 回归。而在支持向量机中，只有离边界最近的数据（支持向量）对分类有贡献，但是在取样较少时无法准确找出支持向量。因此只能考虑 logistic 回归。可以考虑的损失函数有交叉熵、MSE。此时，$(\star)$ 不再适用，需要重新建立模型进行计算。

6 小组分工及个人总结

- 王津：问题研讨分析 + 数据集抽样算法实现

  个人总结：首先，大作业一定程度纠正了我对数据科学的误解。我之前看过一个数据挖掘比赛的指导视频，演讲者解题时，毫无统计思想，完全靠使用 pytorch 的熟练度。所以我一度觉得统计在数据科学中是没有优势的。但是这次的大作业的结果让我惊讶，我没想到，对抽样方法进行简单调整，结果的准确度会变化如此大。统计思维还是很重要的。其次，大作业督促了我自学统计知识。为了解决问题，我必须要阅读相关资料。虽然最后一问没能成功给出解答，但到底还是熟悉了分类的知识。此外，做大作业时，我意外地发现其他课的被我认为枯燥的内容居然可以用于解决本问题，增加了我对那门课的兴趣，也加深了对一些概念的理解。

- 张智琳：问题研讨分析 + 结果评价算法实现

  个人总结：这次大作业我完成的部分是问题的评价分析，我们组讨论了很多评价方法模型，我们选取了相对直观的参数，但研究过程有助于增进我对统计学习的深入了解，这门课的设定阶段很好，能让我们大二的学生，对数据科学产生初步的认识，并能够明白自己在做什么，学习的意义如此。我们组的四名成员因为我们组的成员来自四个不同的学院，大家交流中可以产生各种各样新奇的想法，这种经历很有趣，这也是我与他们组队的初衷之一，接受不同的思考角度和顺序，受益匪浅。

- 陈富鹏：问题研讨分析 +latex 论文写作

  个人总结：我们讨论了许多方法，天马行空的想法，许多方法停在了理论上，没有算法实现。大家来自不同的学院，思维比较开阔，交流碰撞出了数量可观的火花，在一个人提出想法后，总会有不同角度考虑到的对方法不够合理的质疑，各执己见又相互妥协。我们进行了很长时间腾讯会议，我总会认真思考他们提出的观点和意见，对于后面两个过程比较复杂的问题，我们展开了激烈讨

论，我总会在某些数据集的处理案例中汲取新的观点和知识，受益匪浅，同时，也明白了自己更需要将数学和更多的实际学科结合，从实际意义的角度去探索学习。

- 王皖宜：问题研讨分析 +latex 论文写作

  个人总结：这次大作业打开了我对数据科学的新的认识，我做过一些数学建模的比赛，对于建模过程以及处理方法比较熟悉，但对于纯数学角度考虑某个问题这部分却比较薄弱，这次的题目是抽样方法以及它本身的评价，是我以前忽略的过程，没有训练过或者说是做的不够好。打比赛时做的数据处理部分很粗糙，总会感觉数据集处理的是否严谨和后面的建模操作关系不大，注重建模过程，最后模型评价指数可能只差 0.0001，考虑的第一个方面一定是建模的问题，很难想到去观察数据集的合理性，抽样是否严谨等等。我和具有坚实数理基础的小伙伴们讨论问题，他们提到的观点或是专业词汇等，我甚至需要百度一下证明过程或是相关介绍，虽然困难，但提升很大，真正理解了前面对于模型设计的描述，跳脱了以前单纯百度论文、套用方法、不谈基础只谈上层建筑、没有数理依据就直接写算法的建模，只理解建模的目的，不知道方法的原理。现在视野更广阔，也明白了自己还需努力，有更大的提升空间。

参考文献

[1] 王松桂, 线性统计模型——线性回归与方差分析, 北京,1999, pg.60

[2] 王松桂, 线性统计模型——线性回归与方差分析, 北京,1999, pg.72