关于 LASSO 方法在不同线性模型中的有效性探索

——《统计算法基础》课程大作业

PB20071456 付乐豪 PB20151752 王姝懿

07/07/2023

1 介绍

1.1 研究背景

在《统计算法基础》课程中，我们介绍了近端梯度法在 LASSO 方法中的应用。PPT-5 的第 69-71 页给出了一个具有稀疏性的线性模型（以下简称为例 1）。在例 1 中，通过 LASSO 方法估计后发现对回归系数的估计效果很好；但在我们完成课后作业时发现，第四次作业的第三题（以下简称为例 2）也是一个具有稀疏性的线性模型，我们对此使用 LASSO 估计后发现对于回归系数的估计效果并不理想，尽管例 2 中估计的系数 $\hat{\beta}$ 本身具有稀疏性，但是和真参数 β_0 距离较远。两个具有稀疏性的线性模型，为什么例 1 中 LASSO 的效果很好而例 2 中效果欠佳，这引起了我们的好奇。通过讨论后我们将原因归结于样本量和参数的维数：例 1 中样本量 n 设置为 500，参数维数 p 设置为 8；而例 2 中样本量 n 设置为 100，参数维数 p 设置为 300。我们认为可能是样本量 n 和参数维数 p 的比例影响了 LASSO 的效果。

另一方面，LASSO 本身是通过引入 L1 正则项给估计的系数带来了稀疏性，其中的超参数 λ 在 LASSO 方法中扮演着至关重要的作用。我们很好奇在不同的具有稀疏性的线性模型下，究竟适不适合使用 LASSO 方法；如果要使用 LASSO 方法的话，应该如何选取超参数 λ。本文中，我们将会探讨（1）超参数 λ 如何影响回归系数的估计。（2）线性模型中的模型变量，包括模型本身的稀疏程度、样本量和参数维数的比例以及线性模型中误差的分布对于 LASSO 有效性的影响。（3）对于不同的模型如何选取 λ。我们将尝试通过数值模拟的办法对于超参数 λ 的选取给出一些定性指导。

1.2 研究内容

1.2.1 具体内容

本文探索的内容有：验证超参数 λ 的选择对于 LASSO 问题的重要性；在不同模型下（包括模型本身的稀疏程度、样本量和参数维数的比例以及线性模型中误差的分布这三个变量）测试 LASSO 的有效性；如果适合使用 LASSO，我们将尝试通过数值模拟的办法对于如何选取超参数 λ 给出一些定性指导，如下面的流程图：

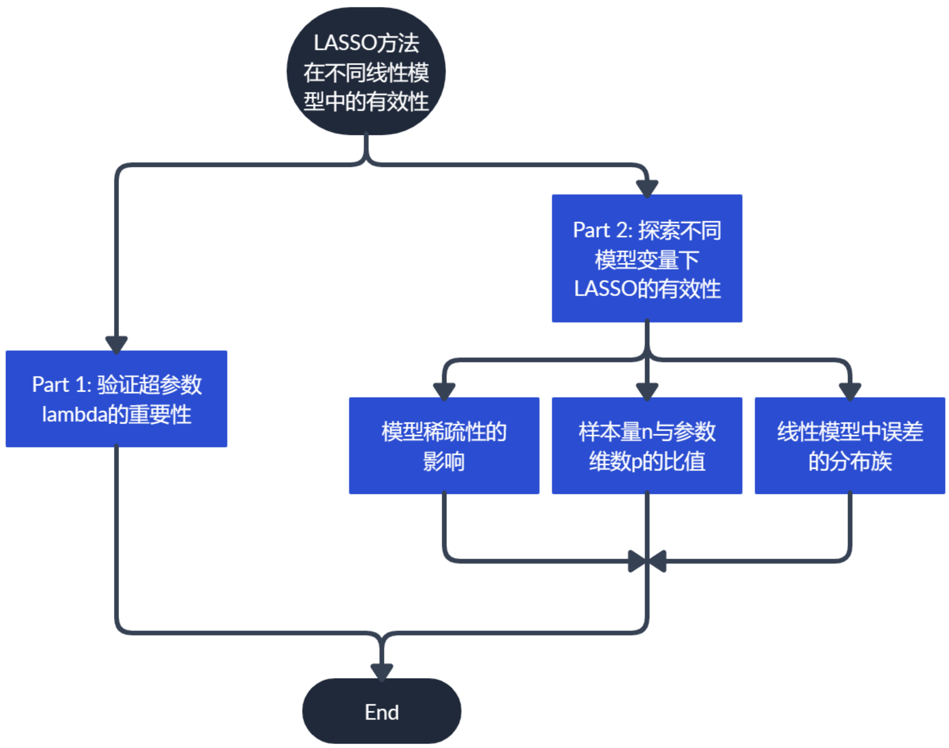

Figure 1: LASSO 流程图

1.2.2 模型理论

假设有数据 $y \in \mathbb{R}^n, X \in \mathbb{R}^{n\times p}$ 满足线性模型 $y = X\beta + \epsilon$，其中 $y = (y_1, \cdots, y_n)^{\mathrm{T}}, X = (x_1, \cdots, x_n)^{\mathrm{T}}$。普通最小二乘估计（OLS）为

$$\hat{\beta}_{OLS} = \arg\min_{\beta} \|y - X\beta\|_2^2 = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y \tag{1}$$

而如果回归系数 β 具有稀疏性（即 β 中只有少数分量非零），此时 LASSO 方法通过在惩罚项中引入 β 的 L1 正则项对于系数的估计引入稀疏性。具体来说，LASSO 考虑如下求解问题

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{2}$$

其中 λ 为人为给定的超参数。区别于 OLS，此问题没有显示解，需要使用数值迭代的办法求解，具体求解算法见 **Algorithm 1**。

1.2.3 关联性与创新点

关联性方面。首先，本文研究的主题 LASSO 方法是 PPT-5 中近端梯度法的一个应用。其次，我们的问题启发于上课和课后作业中的例 1 和例 2 中，LASSO 方法有效性的不一致性。最后，在我们生成不同的线性模型时，也用到了 PPT-1 中所讲授的生成随机数的知识。

创新点方面。首先，上课和作业仅仅是对于两个特定稀疏线性模型使用了 LASSO 方法进行估计，并没有深入探讨为什么两个例子中 LASSO 方法的有效性不一致，什么情况下适合使用 LASSO，而本文对此做出了进一步的探索。其次，课堂中只提到了 LASSO 估计时，超参数 λ 的选取很重要，并没有具体讲解 λ 会如何影响估计，也没有介绍不同情况下应该如何选取 λ，而本文对此做了一些数值实验，尝试给出一些定性的理论指导。

2 模拟实验

2.1 实验设计

2.1.1 求解算法与拟合程度衡量标注

首先，我们给出使用近端梯度法求解 LASSO 方法的优化问题（2）的算法:

Algorithm 1: 近端梯度法求解 LASSO

Initialization: $\beta_k = \mathbf{1}_p, d = 1, iter = 1$
while $iter \leq 100$ and $d \geq \epsilon$ **do**
  $s_t \leftarrow \frac{n}{\|X^{\mathrm{T}}X\|}$ ▷ 近端梯度法中我们使用固定步长 s_t
  $t \leftarrow \beta_k + \frac{1}{n} s_t X^{\mathrm{T}}(y - X\beta_k)$
  $\beta_{k+1} \leftarrow \mathcal{S}_{s_t\lambda}(t)$ ▷ 其中 $\mathcal{S}_\lambda(x)_j = sign(x)(|x_j| - \lambda)_+$ 为收缩算子
  $d \leftarrow \|\beta_{k+1} - \beta_k\|_1$
  $\beta_k \leftarrow \beta_{k+1}$
end
Output: $\beta_k, iter$

为了方便对比不同模型下 LASSO 方法的估计效果，对于回归系数真值 β_0 和 LASSO 的估计值 $\hat{\beta}$ 我们定义**误差率**:

$$e = \frac{\|\hat{\beta} - \beta_0\|_1}{\|\beta_0\|_1} \tag{3}$$

需要注意的是，当误差率 $e=0$ 的时候说明估计值完全拟合；而如果 $\hat{\beta}=\mathbf{0}_p$，此时误差率 $e=1$。以下实验结果中将会看到，当 λ 过大的时候，误差率会趋于 1，所以我们在衡量估计准确度时，我们认为如果 $e\geq 1$，$\hat{\beta}$ 的估计精度甚至不如 $\mathbf{0}_p$，此时的估计是无效的。

2.1.2 Part 1: 超参数的重要性验证

在这一部分，我们想要知道在同一个模型中，选取不同的超参数 λ 对于 LASSO 估计的精度影响如何。这里我们选取了例 1 和例 2 作为测试的例子。具体参数设置见 Table 1：

Table 1: Part 1 中两个例子的参数设置

测试用例	**样本量** n	**参数维数** p	**回归系数** β	**误差分布**
例 1	500	8	$(3,1.5,0,0,2,0,0,0)^{\mathrm{T}}$	$\mathcal{N}(0,1)$
例 2	100	300	$(\mathbf{1}_{10}^{\mathrm{T}},\mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$

我们在 $(0,3)$ 区间中选取以 0.1 为公差的等差数列作为 λ 的所有可能取值。对于每一个 λ 取值，计算其对应的 LASSO 估计，并比较不同 λ 带来的估计误差率。

2.1.3 Part 2: 不同模型变量下的 LASSO 有效性

在这一部分，我们想要知道在不同的稀疏线性模型变量下，LASSO 估计是否还有效，和 OLS 的估计精度相比如何，对于不同的模型，我们应该如何选取 λ 以达到较小的估计误差。为此，我们选取了:（1）模型本身的稀疏程度（2）样本数和参数维数的比值 n/p（3）误差的分布，这三个可能会影响 LASSO 估计精度的模型参数来探索，以下分别介绍三组实验中的参数设置。

（1）模型的稀疏程度

我们通过控制其余变量相同，只改变 β 中的非 0 分量维数来改变模型的稀疏性：非 0 分量越多，模型稀疏性约差；非 0 分量越少，模型越稀疏。具体参数设置见 Table 2。

（2）样本数和参数维数的比值 n/p

我们通过控制其余变量相同并恒定参数维数，只改变样本量 n 来改变模型中的样本量与参数维数的比值，具体参数设置见 Table 3。

（3）误差的分布

我们控制其余变量相同，只改变生成线性模型响应值 y 时的误差分布，具体参数设置见 Table 4。

Table 2: Part 2 模型的稀疏程度实验参数设置

测试用例编号	**样本量** n	**参数维数** p	**回归系数** β	**误差分布**
A	100	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$
B	100	300	$(\mathbf{1}_{20}^{\mathrm{T}}, \mathbf{0}_{280}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$
C	100	300	$(\mathbf{1}_{50}^{\mathrm{T}}, \mathbf{0}_{250}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$
D	100	300	$(\mathbf{1}_{100}^{\mathrm{T}}, \mathbf{0}_{200}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$

Table 3: Part 2 模型样本量与参数维数实验参数设置

测试用例编号	**样本量** n	**参数维数** p	**回归系数** β	**误差分布**
A	10	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$
B	50	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$
C	300	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$
D	3000	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$

Table 4: Part 2 模型中误差的分布实验参数设置

测试用例编号	**样本量** n	**参数维数** p	**回归系数** β	**误差分布**
A1	100	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,1)$
A2	100	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,3)$
A3	100	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,4)$
A4	100	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	$\mathcal{N}(0,10)$
B	100	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	t_2
C	100	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	Cauchy(1,0)
D	100	300	$(\mathbf{1}_{10}^{\mathrm{T}}, \mathbf{0}_{290}^{\mathrm{T}})^{\mathrm{T}}$	Chisq(2)

2.2 实验结果

对于 2.2 中共两个部分，四组实验，实验参数具体设置见2.1 实验设计，以下通过图给出实验结果。其中每一个图的横轴为超参数 λ 的取值，纵轴为在此 λ 取值下，通过近端梯

度法数值求解 LASSO 问题后得到的回归系数估计值 $\hat{\beta}$ 的误差率。

2.2.1 Part 1: 超参数的重要性验证实验结果

例 1

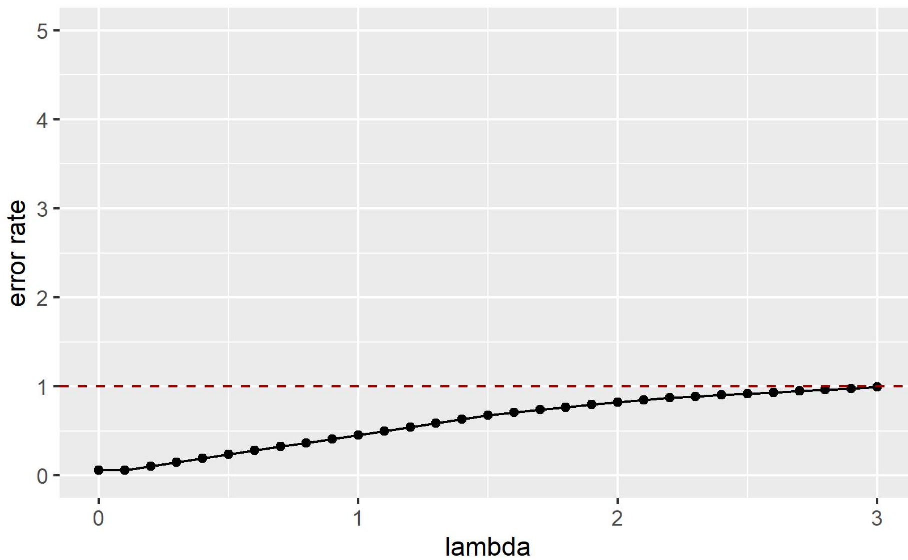

Figure 2: 超参数重要性验证例 1

例 2

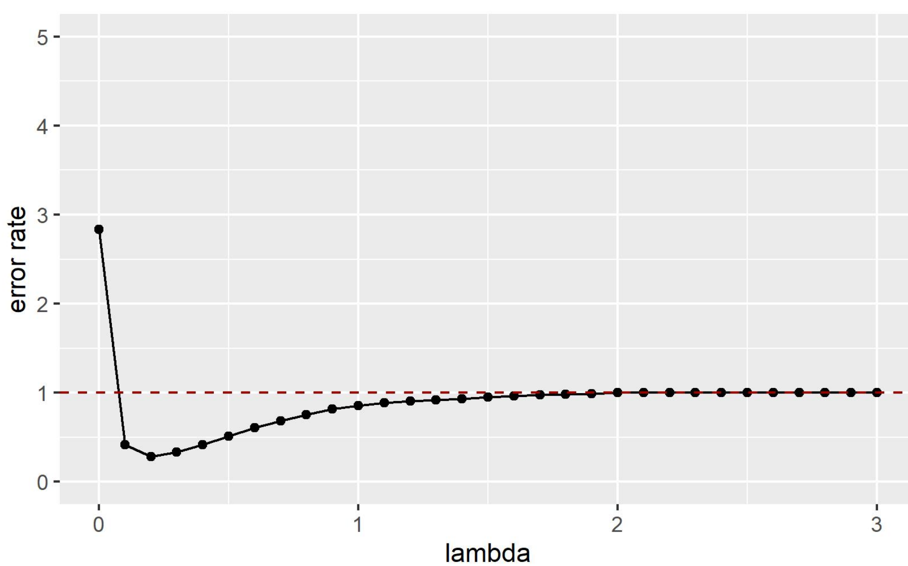

Figure 3: 超参数重要性验证例 2

2.2.2 Part 2: 不同模型变量下的 LASSO 有效性实验结果

（1）模型的稀疏程度

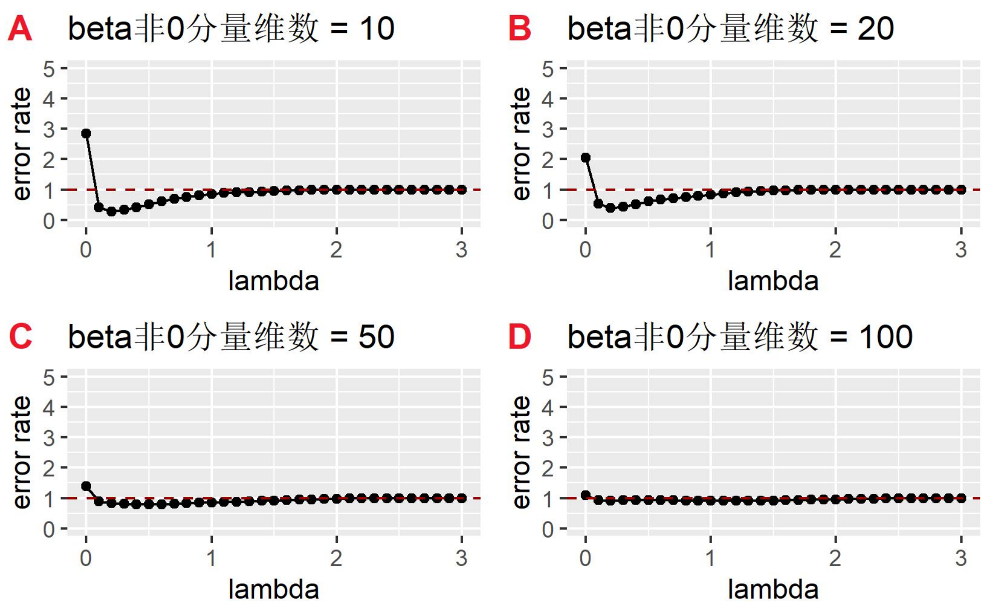

Figure 4: 模型的稀疏程度探索

（2）样本数和参数维数的比值 n/p

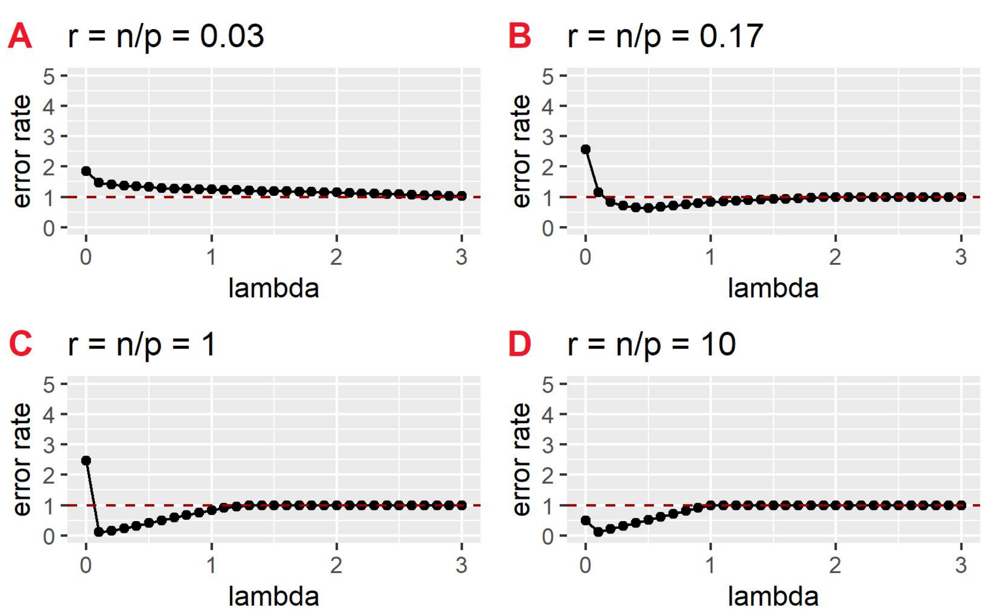

Figure 5: 样本数和参数维数的比值 n/p 探索

（3）误差的分布

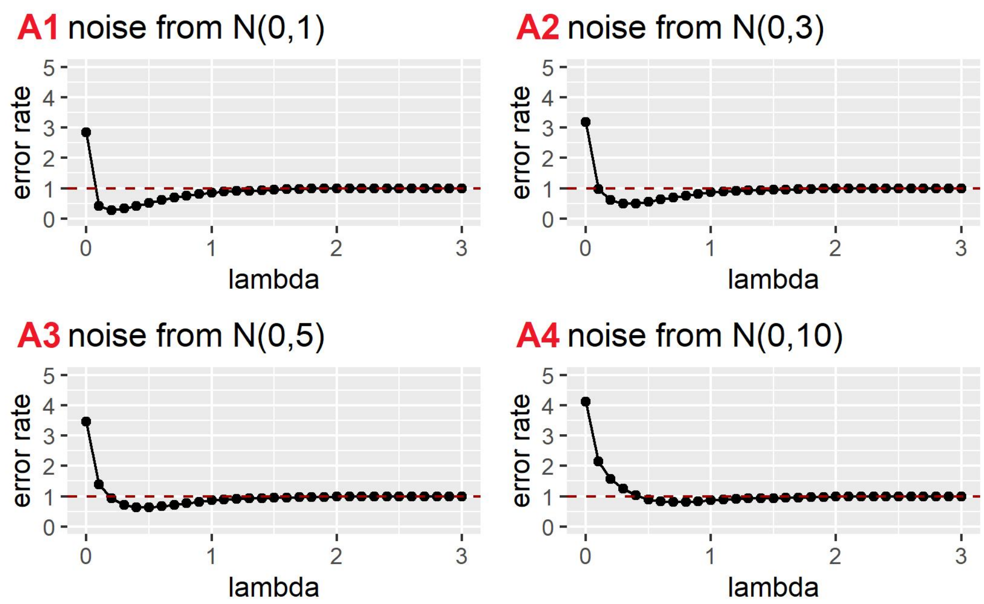

Figure 6: 误差的分布探索（正态分布对比）

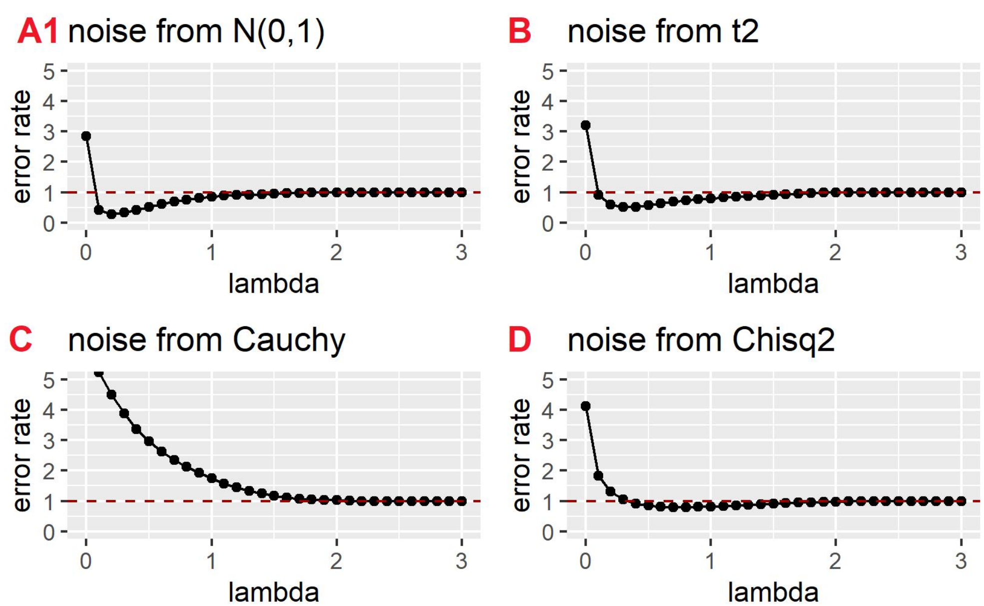

Figure 7: 误差的分布探索（不同分布族对比）

2.3 分析与归纳

对于上述实验结果，四组实验的现象分别归纳如下：

- 例 1 中，超参数 λ 的选取对于降低回归系数的估计误差效果并不明显；而例 2 中，当 λ 在 0.1 附近时，LASSO 方法对于降低估计误差很有效。两个例子中，当 λ 很大时，我们定义的误差率会趋于 1，通过输出具体的估计值可以看出，此时 $\hat{\beta} = \mathbf{0}_p$。

- 模型的稀疏程度对于 LASSO 的效果影响很大，在稀疏性较明显的组，通过在目标函数中添加 L1 正则项，LASSO 估计起到了很好的作用；而在那些稀疏性不明显的组，LASSO 对于降低估计的误差率效果不明显。

- 在样本数较大的组，使用 LASSO 方法确实能够降低误差，但是相比于 OLS（$\lambda = 0$）的情况，其降低作用并不明显；而在样本数较少的组，虽然使用 LASSO 方法后误差仍然不低，但是相比于 OLS，LASSO 能够显著的降低误差率。

- 当误差来自于正态分布时，误差的波动越小（误差方差越小），LASSO 方法的效果越好。而当误差来自于 t 分布，LASSO 能够提高估计误差；当误差来自于卡方分布时，LASSO 的效果就不那么明显了；如果误差来自于柯西分布，则引入 LASSO 方法相较于 OLS 仍然是无效的。

对于我们文章想要探索的三个问题：（1）LASSO 方法中的超参数 λ 如何影响回归系数的估计。（2）线性模型中的模型变量对于 LASSO 有效性的影响。（3）对于不同的模型如何选取 λ。我们通过实验结果可以得到以下结论：

- 稀疏线性模型中，超参数 λ 的选取对于回归系数 β 的估计精度影响很大。过小的 λ 可能会导致模型受到误差污染过大，从而估计效果很差；而过大的 λ 会导致惩罚项中过于看重 L1 正则项，从而使 λ 趋于 $\mathbf{0}_p$。

- 模型的稀疏性，样本量与参数维数的比值以及误差的分布都对 LASSO 方法的有效性有影响。模型的稀疏性越明显，LASSO 越有效。样本量较大时，LASSO 方法和 OLS 相差不大，而样本量较小时，LASSO 方法能够很好的提高估计精度。误差来自于正态分布时，波动越小（误差方差越小），LASSO 方法的效果越好；而当误差来自于别的分布时，LASSO 的有效性不如正态的情况。

- 不同的问题的超参数 λ 的最优选取没有固定的表达，但是如果此问题中 LASSO 方法效果较好，最优的 λ 往往落在区间 $(0, 1)$ 中。

此外，基于上述结论，我们对于文章开头提到的问题做出解答。例 1 中尽管使用 LASSO 方法后估计效果较好，误差率在 0.06，但我们倾向于认为这是由于样本量远大于参数维数导致的，这时 LASSO 方法和 OLS 比较起来并没有显著的优势；而例 2 中尽管使用 LASSO 方法后误差率在 0.82 左右，但这是由于样本数太少，小于参数维数，我们对于模型的认知不够导致的，LASSO 方法此时相较于 OLS，显著降低了估计的误差。这个结论既在意料之外又在情理之中。

3 总结

本文探讨了 LASSO 方法在不同线性模型中的有效性，通过不同的模拟实验，对于文章开头提到的三个问题进行了探索，并在文章结尾出给出了结论。本文的出发点是课程中介绍的一道例题和上机实验中的一道例题，由此出发对 LASSO 方法进行了深入的研究，在模拟实验中通过数值办法了解其性质，既做到了和上课关联还兼顾了拓展，是一次成功的探索。

当然，本文仍然有一些不足之处，例如在研究当稀疏线性模型的误差 ϵ 来自不同分布时，LASSO 方法估计的有效性时，我们并没有找到适合用对应误差分布建模的实际例子，对于这一组实验在实际应用中是否有用仍然需要进一步的查阅资料与收集数据；对于超参数 λ 的选取只能给出一个"最优的 λ 往往落在区间 $(0,1)$ 之间"，而并没有深入的提出对于不同的模型，究竟应该如何选取 λ。这些问题还有待进一步的研究。

A 附录：代码

0 课本例题与作业题

0.1 Estimation function

```
soft <- function(x, lam){
  ifelse(abs(x) > lam, sign(x)*(abs(x)-lam), 0)
}

l1_norm <- function(x, y){
  sum(abs(x))
}

Prox_lasso <- function(n, p, beta0, lambda, iters, output, error_dis =
   "norm", sigma2 = 1){
  set.seed(1)
  x = matrix(rnorm(n*p), n, p)
  if(error_dis == "norm"){
    y = x %*% beta0 + rnorm(n, mean=0, sd=sqrt(sigma2))
  }else if(error_dis == "chisq"){
    y = x %*% beta0 + rchisq(n, df=2)
  }else if(error_dis == "t"){
    y = x %*% beta0 + rt(n, df=2)
  }else if(error_dis == "cauchy"){
    y = x %*% beta0 + rcauchy(n)
  }
  N = iters
  beta_k <- rep(0, p)
  beta_diff = 1; beta_diff_l = c()
  iter = 0
  epsilon = sqrt(.Machine$double.eps)
  step = n/(svd(t(x)%*%x)$d[1])

  # iteration
  for(iter in 1:N){
    iter = iter + 1
    temp = beta_k + (1/n)*step*t(x)%*%(y - x%*%beta_k)
    beta_k_1 = sapply(temp, soft, lambda*step)
    beta_diff = l1_norm(beta_k_1 - beta_k)
    beta_diff_l = c(beta_diff_l, beta_diff)
```

```
    beta_k = beta_k_1
    if(beta_diff < epsilon) break
  }
  result = rbind(round(beta0, 2), round(beta_k, 2))
  rownames(result) = c("beta0", "beta.hat")
  colnames(result) = seq(from=1, to=p, by=1)
  nonzero_index = (apply(result, 2, function(x) sum(x == 0)) != 2)
  error_rate = l1_norm(beta0 - beta_k)/l1_norm(beta0)
  if(output == T){
    cat("dim of x = (", n, ", ", p, ")", "\n", "dim of y = ", n, "\n",
       "max_iters = ", N, "\n\n", sep = "")
    cat("以下仅输出参数的非0列: ", "\n")
    print(result[, nonzero_index])
    cat("iteration:", iter, "\t", "L1 loss:", l1_norm(beta0 - beta_k),
       "\n")
    cat("error rate = ", error_rate, "\n")
    plot(beta_diff_l, xlab = "t", ylab = "Error", type = "l", main = "
       L1 loss between beta_k and beta_{k+1}")
  }
  return(error_rate)
}
ggerror_curve <- function(lambda_seq, error_rate_l){
  data = data.frame(lambda_seq = lambda_seq, error_rate_l = error_rate_
     l)
  p = ggplot(data=data, aes(x=lambda_seq, y=error_rate_l))
  p = p + geom_point() + geom_line() + labs(x = "lambda", y = "error
     rate") + geom_hline(aes(yintercept=1), colour="#990000", linetype=
     "dashed") + coord_cartesian(ylim = c(0, 5))
  return(p)
}
```

0.2 PPT 例子，例 1

```
# generate data
n = 500; p = 8; lambda = 0.1; iters = 100
beta0 = c(3, 1.5, 0, 0, 2, 0, 0, 0)
error_rate = Prox_lasso(n, p, beta0, lambda, iters, output = T)
```

0.3 作业例子，例 2

```
n = 100; p = 300; lambda = 0.2; iters = 1000
beta0 = c(rep(1/sqrt(10), 10), numeric(p-10))
```

```
error_rate = Prox_lasso(n, p, beta0, lambda, iters, output = T)
```

1 Part1 验证 lambda 对于稀疏模型的回归估计的重要性

1.1 例 1

```
n = 500; p = 8; iters = 100
beta0 = c(3, 1.5, 0, 0, 2, 0, 0, 0)
lambda_seq = seq(from=0, to=3, by=0.1)
error_rate_l = numeric(length(lambda_seq))
for(i in 1:length(lambda_seq)){
  error_rate = Prox_lasso(n, p, beta0, lambda = lambda_seq[i], iters =
     iters, output = F)
  error_rate_l[i] = error_rate
}
# plot(x=lambda_seq, y=error_rate_l, type="l")
p = ggerror_curve(lambda_seq, error_rate_l)
p
ggsave("output/part1/example1.jpg", p)
```

1.2 例 2

```
n = 100; p = 300; iters = 100
beta0 = c(rep(1, 10), numeric(p-10))
lambda_seq = seq(from=0, to=3, by=0.1)
error_rate_l = numeric(length(lambda_seq))
for(i in 1:length(lambda_seq)){
  error_rate = Prox_lasso(n, p, beta0, lambda = lambda_seq[i], iters =
     iters, output = F)
  error_rate_l[i] = error_rate
}
# plot(x=lambda_seq, y=error_rate_l, type="l")
p = ggerror_curve(lambda_seq, error_rate_l)
p
ggsave("output/part1/example2.jpg", p)
```

2 Part2 以下变量变化时，最优的 lambda 的变化情况

2.1 模型稀疏性

```
n = 100; p = 300; iters = 100
beta0 = matrix(numeric(4*p), 4, p)
beta0[1, 1:10] = rep(1, 10)
```

```
beta0[2, 1:20] = rep(1, 20)
beta0[3, 1:50] = rep(1, 50)
beta0[4, 1:100] = rep(1, 100)
lambda_seq = seq(from=0, to=3, by=0.1)
error_rate_l = matrix(numeric(4*length(lambda_seq)), 4, length(lambda_
    seq))

for(i in 1:4){
  for(j in 1:length(lambda_seq)){
    error_rate = Prox_lasso(n, p, beta0 = beta0[i,], lambda = lambda_
       seq[j], iters = iters, output = F, error_dis = "norm")
    error_rate_l[i,j] = error_rate
  }
}

p1 = ggerror_curve(lambda_seq, error_rate_l[1,]) + ggtitle(paste0("beta
    非0分量维数 = ", 10))
p2 = ggerror_curve(lambda_seq, error_rate_l[2,]) + ggtitle(paste0("beta
    非0分量维数 = ", 20))
p3 = ggerror_curve(lambda_seq, error_rate_l[3,]) + ggtitle(paste0("beta
    非0分量维数 = ", 50))
p4 = ggerror_curve(lambda_seq, error_rate_l[4,]) + ggtitle(paste0("beta
    非0分量维数 = ", 100))
p = ggpubr::ggarrange(p1, p2, p3, p4, nrow = 2, ncol = 2, labels = c('A
    ', 'B', 'C', 'D'), font.label = list(color = "#ef1828"))
p
ggsave("output/part2/sparse_beta.jpg", p)
```

2.2 r = n/p（即样本量与参数维数的比值）

```
p = 300; iters = 100; n_seq = c(10, 50, 300, 3000); r = n_seq/p
beta0 = c(rep(1, 10), numeric(p-10))
lambda_seq = seq(from=0, to=3, by=0.1)
error_rate_l = matrix(numeric(length(n_seq)*length(lambda_seq)), length
    (n_seq), length(lambda_seq))

for(i in 1:length(n_seq)){
  for(j in 1:length(lambda_seq)){
    error_rate = Prox_lasso(n_seq[i], p, beta0, lambda = lambda_seq[j],
        iters = iters, output = F)
    error_rate_l[i,j] = error_rate
```

```
  }
}

p1 = ggerror_curve(lambda_seq, error_rate_l[1,]) + ggtitle(paste0("r =
   n/p = ", round(n_seq[1]/p, 2)))
p2 = ggerror_curve(lambda_seq, error_rate_l[2,]) + ggtitle(paste0("r =
   n/p = ", round(n_seq[2]/p, 2)))
p3 = ggerror_curve(lambda_seq, error_rate_l[3,]) + ggtitle(paste0("r =
   n/p = ", round(n_seq[3]/p, 2)))
p4 = ggerror_curve(lambda_seq, error_rate_l[4,]) + ggtitle(paste0("r =
   n/p = ", round(n_seq[4]/p, 2)))
p = ggpubr::ggarrange(p1, p2, p3, p4, nrow = 2, ncol = 2, labels = c('A
   ', 'B', 'C', 'D'), font.label = list(color = '#ef1828'))
p
ggsave("output/part2/r=n_p(p=300).jpg", p)
```

2.3 误差来自不同分布时，error rate 的变化

2.3.1 正态，不同方差

```
n = 100; p = 300; iters = 100
beta0 = c(rep(1, 10), numeric(p-10))
sigma2_seq = c(1,3,5,10)
lambda_seq = seq(from=0, to=3, by=0.1)
error_rate_l = matrix(numeric(length(sigma2_seq)*length(lambda_seq)),
   length(sigma2_seq), length(lambda_seq))

for(i in 1:length(sigma2_seq)){
  for(j in 1:length(lambda_seq)){
    error_rate = Prox_lasso(n, p, beta0, lambda = lambda_seq[j], iters
       = iters, output = F, error_dis = "norm", sigma2 = sigma2_seq[i])
    error_rate_l[i,j] = error_rate
  }
}

p1 = ggerror_curve(lambda_seq, error_rate_l[1,]) + ggtitle(paste0("
   noise from N(0,", sigma2_seq[1], ")"))
p2 = ggerror_curve(lambda_seq, error_rate_l[2,]) + ggtitle(paste0("
   noise from N(0,", sigma2_seq[2], ")"))
p3 = ggerror_curve(lambda_seq, error_rate_l[3,]) + ggtitle(paste0("
   noise from N(0,", sigma2_seq[3], ")"))
p4 = ggerror_curve(lambda_seq, error_rate_l[4,]) + ggtitle(paste0("
```

```
    noise from N(0,", sigma2_seq[4], ")"))
p = ggpubr::ggarrange(p1, p2, p3, p4, nrow = 2, ncol = 2, labels = c('
    A1', 'A2', 'A3', 'A4'), font.label = list(color = "#ef1828"))
p
ggsave("output/part2/norm_var.jpg", p)
```

2.3.2 其他分布族，t, Cauchy, Chisq

```
n = 100; p = 300; iters = 100
beta0 = c(rep(1, 10), numeric(p-10))
lambda_seq = seq(from=0, to=3, by=0.1)
error_rate_l = matrix(numeric(4*length(lambda_seq)), 4, length(lambda_
    seq))

# N(0, 1)
for(j in 1:length(lambda_seq)){
  error_rate = Prox_lasso(n, p, beta0, lambda = lambda_seq[j], iters =
      iters, output = F, error_dis = "norm")
  error_rate_l[1,j] = error_rate
}
# t_2
for(j in 1:length(lambda_seq)){
  error_rate = Prox_lasso(n, p, beta0, lambda = lambda_seq[j], iters =
      iters, output = F, error_dis = "t")
  error_rate_l[2,j] = error_rate
}
# Cauchy
for(j in 1:length(lambda_seq)){
  error_rate = Prox_lasso(n, p, beta0, lambda = lambda_seq[j], iters =
      iters, output = F, error_dis = "cauchy")
  error_rate_l[3,j] = error_rate
}
# Chisq_2
for(j in 1:length(lambda_seq)){
  error_rate = Prox_lasso(n, p, beta0, lambda = lambda_seq[j], iters =
      iters, output = F, error_dis = "chisq")
  error_rate_l[4,j] = error_rate
}

p1 = ggerror_curve(lambda_seq, error_rate_l[1,]) + ggtitle(paste0("
    noise from N(0,1)"))
```

```r
p2 = ggerror_curve(lambda_seq, error_rate_l[2,]) + ggtitle(paste0("
   noise from t2"))
p3 = ggerror_curve(lambda_seq, error_rate_l[3,]) + ggtitle(paste0("
   noise from Cauchy"))
p4 = ggerror_curve(lambda_seq, error_rate_l[4,]) + ggtitle(paste0("
   noise from Chisq2"))
p = ggpubr::ggarrange(p1, p2, p3, p4, nrow = 2, ncol = 2, labels = c('
   A1', 'B', 'C', 'D'), font.label = list(color = "#ef1828"))
p
ggsave("output/part2/diff_dist.jpg", p)
```