Выбор сложности модели через максимизацию производной суммы квадратов остатков

Борис Демешев

14 мая 2015

Минимизация суммы квадратов остатков

Простая модель линейной регрессии:

$$y_i = \beta_1 + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \varepsilon_i$$

МНК: подбираем β минимизируя RSS.

Проблема с параметром сложности модели

$$y_i = \beta_1 + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \varepsilon_i$$

Однако нельзя подобрать k минимизируя RSS.

Чем больше сложность модели, k, тем меньше будет RSS.

Дилемма сложность–сумма квадратов остатков универсальна

- LASSO
- Ridge-регрессия
- Классификационные деревья
- Ядерная оценка функции плотности
- Ядерная оценка в непараметрической регрессии

...

Кратко о LASSO

Зафиксируем k.

Минимизируем по β сумму квадратов остатков, оштрафованную на сложность модели:

$$\min RSS + \lambda \cdot (|\beta_2| + |\beta_3| + \ldots + |\beta_k|)$$

Чем больше штрафной параметр λ, тем (грубо говоря) ближе оптимальные β^* будут к нулю.

Оптимизировать по λ бессмысленно.

Известное решение — кросс-валидация

1. Поделим имеющуюся выборку на 10 частей.
2. Зафиксируем некоторое значение штрафного параметра λ.
3. По подвыборке состоящей из всех частей кроме первой оценим β с помощью LASSO.
4. Используя полученные $\hat{\beta}_{(1)}$ получим прогнозы для наблюдений из первой части.

5-6. Повторим шаги 3-4 для подвыборки, в который из всех наблюдений удалена вторая часть.

Известное решение — кросс-валидация

7-8. Повторим шаги 3-4 для подвыборки, в который из всех наблюдений удалена третья часть.

...

n. Получив прогнозы для каждого наблюдения посчитаем $RSS_{cv}(\lambda)$ для зафиксированного λ

$n+1$. Проделав шаги 2-n для разных λ, выберем то, которое минимизирует RSS_{cv}.

Новое решение — максимизация производной RSS

1. Зафиксируем некоторое значение штрафного параметра λ.
2. По всей выборке оценим β с помощью LASSO.
3. Проделав шаги 1-2 для разных λ, получим зависимость $RSS(\lambda)$, выберем то λ, которое максимизирует $dRSS(\lambda)/d\lambda$.

Численный пример

Искусственные данные: $x_{ik} \sim N(0,1)$, $z_{ik} \sim N(0,1)$, $\varepsilon_i \sim N(0,1)$, 200 наблюдений:

$$y_i = 2 + 3x_{i1} - 2x_{i2} + \varepsilon_i$$

С помощью LASSO оцениваем регрессию y_i на x_{i1}, x_{i2}, z_{i1}, z_{i2} при разных лямбда

```
cv.fit <- cv.glmnet(X,y, lambda = lambdas)
```

Результаты для разных штрафных коэффициентов

```
coefs <- as.matrix(coef(cv.fit, s=c(0,0.5,1,2,10)))
coefs <- rbind(c(0,0.5,1,2,10),coefs)
rownames(coefs)[1] <- "lambda"
colnames(coefs) <- NULL
pander(coefs, digits=3)
```

lambda	0	0.5	1	2	10
(Intercept)	2.08	2.06	2.04	2	1.97
x1	3.03	2.56	2.08	1.13	0
x2	-2.08	-1.59	-1.1	-0.118	0
z1	-0.0111	0	0	0	0
z2	0.0535	0	0	0	0

Кросс-валидация выбирает классический МНК

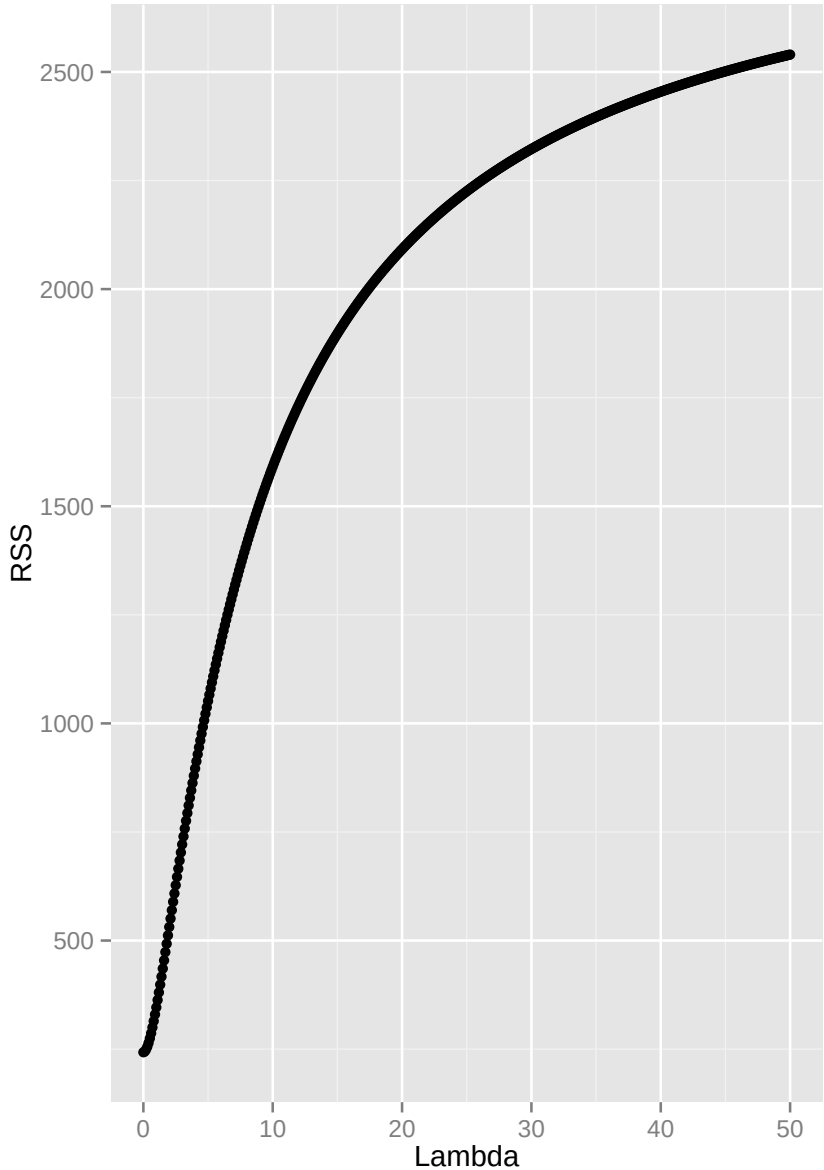
```
cv.fit$lambda.min
```

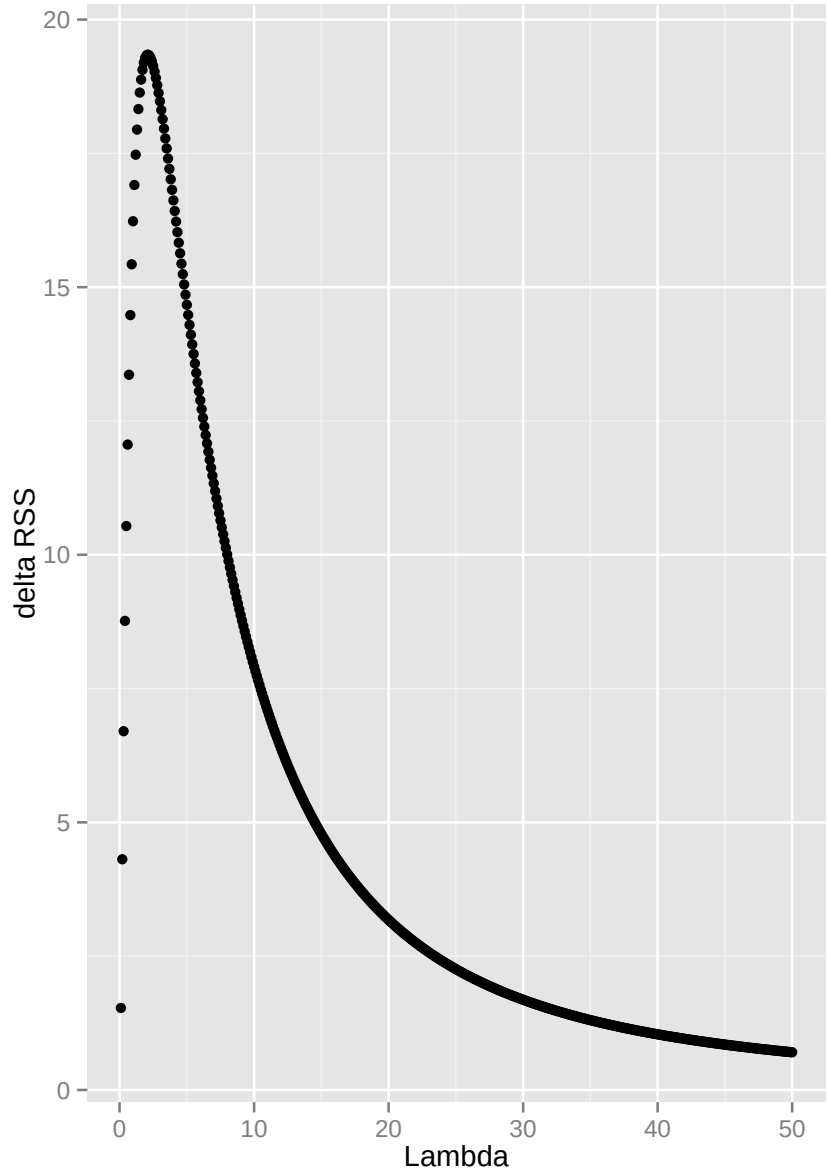
```
## [1] 0
```

```
coef(cv.fit,s="lambda.min")
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  2.08083726
## x1           3.03390002
## x2          -2.07708077
## z1          -0.01110494
## z2           0.05347505
```

Зависимость RSS и $dRSS/d\lambda$ от λ





Оценка лямбда максимизацией $dRSS/d\lambda$

```
lambda.mdr <- lambdas_inc[drss==max(drss)]
lambda.mdr
```

```
## [1] 2.1
```

```
coef(cv.fit, s=lambda.mdr)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  1.9963984
## x1           1.0318885
## x2          -0.0202203
## z1           .
## z2           .
```

Потери по сравнению с “идеальным” решением

model	lambda	RSS_TSS	scorr2	n_coefs
Идеальная	NA	0.08365	0.917	2
Кросс-валидация	0	0.08266	0.9173	4
Максимум производной	2.1	0.1875	0.9164	2

Свойства метода

- Метод зависит от выбора целевого показателя.

Например, вместо суммы квадратов остатков можно взять сумму квадратов модулей.

- У функции $dRSS/d\lambda$ может быть несколько локальных максимумов

Эти слады доступны по ссылке goo.gl/GFMeG3