Сравнение методов Ридж-регрессии и LASSO в задачах обработки данных

Л. Э. Мелкумова[a], С. Я. Шатских[b]

[a]*Меркури Девелопмент Раша, 443069, ул. Авроры, 110 к.1, Самара, Россия*
[b]*Самарский национальный исследовательский университет имени академика С.П. Королева, 443086, Московское шоссе, 34, Самара, Россия*

Аннотация

Работа посвящена сравнению методов линейной регрессии Ридж и LASSO. Преимущества каждого из этих методов рассматриваются на примерах тестовых данных с вычислениями в пакете R.

Ключевые слова: линейная регрессия; Ридж-регрессия; LASSO; кросс-валидация

1. Многомерная линейная регрессия. Метод наименьших квадратов

В линейной регерссионной модели рассматривается линейное соотношение между некоторой переменной (*откликом*) y и объясняющими переменными (*предикторами*) $x_1, \ldots, x_{k-1}$

$$y = b_0 + b_1 x_1 + \ldots + b_{k-1} x_{k-1} + \varepsilon, \tag{1}$$

где величина ε интерпретируется как «погрешность наблюдений», «флюктуация», «влияние неучтенных факторов».

При различных значениях переменных x_j наблюдается n значений переменной y

$$y_i = b_0 + b_1 x_{i1} + \ldots + b_{k-1} x_{i\,k-1} + \varepsilon_i, \quad i = \overline{1,\, n}\ , \tag{2}$$

где x_{ij} - $i-$ ое наблюдение переменной x_j. Величины ε_i непосредственно не наблюдаются.

Вводя дополнительные параметры

$$x_{10} = x_{20} = \ldots = x_{n0} = 1, \tag{3}$$

запишем систему уравнений (2) в матричном виде

$$\mathbf{Y} = \mathbf{XB} + \mathcal{E}, \tag{4}$$

где

$$\mathbf{Y} = [y_i]_n\,, \qquad \mathbf{X} = \left[x_{ij}\right]_{n\times k}\,, \qquad \mathbf{B} = \left[b_j\right]_k\,, \qquad \mathcal{E} = [\varepsilon_i]_n\,. \tag{5}$$

В рамках регрессионного анализа координаты $b_0, b_1, \ldots, b_{k-1}$ вектора $\mathbf{B}$ считаются неизвестными. Ставится задача построения оценок вектора $\mathbf{B}$ на основе многомерной выборки наблюдений

$$[\mathbf{X}, \mathbf{Y}] = \begin{bmatrix} x_{10} & x_{11} & \ldots & x_{1\,k-1} & y_1 \\ x_{20} & x_{21} & \ldots & x_{2\,k-1} & y_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{n0} & x_{n1} & \ldots & x_{n\,k-1} & y_n \end{bmatrix}. \tag{6}$$

Традиционно для нахождения оценок параметров $b_0, b_1, \ldots, b_{k-1}$ используется *метод наименьших квадратов* (МНК):

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k-1} b_j\, x_{ij} \right)^2 \mapsto \min\,. \tag{7}$$

Координаты вектора

$$\widehat{\mathbf{B}} = [\widehat{b}_j]_k, \tag{8}$$

минимизирующие (7), называют МНК-оценками неизвестных параметров $b_0, b_1, \ldots, b_{k-1}$.

Известно, что при

$$\det \mathbf{X}^t\mathbf{X} \ > \ 0 \tag{9}$$

вектор МНК-оценок вычисляется по формуле

$$\widehat{\mathbf{B}} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{Y}. \tag{10}$$

Введем обозначение для *эмпирической регрессии*:

$$\widehat{\mathbf{Y}} := \mathbf{X}\widehat{\mathbf{B}}. \tag{11}$$

В координатах это соотношение можно переписать в виде

$$\widehat{y}_i := \widehat{b}_0 + \widehat{b}_1 x_{i1} + \ldots + \widehat{b}_{k-1} x_{i\,k-1}, \quad i = \overline{1,n}. \tag{12}$$

Величина $\widehat{y}$ является прогнозом отклика y, который соответствует предикторам $x_1, \ldots, x_{k-1}$.

Ошибка МНК-прогноза

$$RSS := \sum_{i=1}^{n} (\widehat{y}_i - y_i)^2. \tag{13}$$

Коэффициент детерминации

$$R^2 := 1 - \frac{\sum\limits_{i=1}^{n} (\widehat{y}_i - y_i)^2}{\sum\limits_{i=1}^{n} (y_i - \overline{y})^2} \in [0,\, 1] \tag{14}$$

позволяет оценить качество прогноза: чем ближе коэффициент детерминации к единице, тем лучше регрессионная модель (11) описывает данные (6).

1.1. Стандартизация данных

В задачах линейного регрессионного анализа часто применяют стандартизацию исходных данных (см. [1]). Именно, на основе выборки исходных данных (6), используя обозначения

$$\overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}\,, \quad \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i\,, \quad S_y^2 = \sum_{i=1}^{n} (y_i - \overline{y})^2\,, \quad S_j^2 = \sum_{i=1}^{n} \left(x_{ij} - \bar{x}_j\right)^2\,, \quad j = \overline{1, k-1}\,, \tag{15}$$

с помощью центрирования и нормировки вводят стандартизованные переменные

$$v_i := \frac{y_i - \overline{y}}{S_y}\,, \qquad w_{ij} := \frac{x_{ij} - \bar{x}_j}{S_j}\,, \quad i = \overline{1,n}, \quad j = \overline{1, k-1}. \tag{16}$$

Вводя матричные обозначения,

$$\mathbf{V} = [v_i]_n\,, \quad \mathbf{W} = \left[w_{ij}\right]_{n\times(k-1)}\,, \tag{17}$$

в том случае, когда $\det \mathbf{W}^t\mathbf{W} > 0$ вектор МНК-оценок коэффициентов стандартизованной модели вычисляется по формуле

$$\widehat{B} = (\mathbf{W}^t\mathbf{W})^{-1}\mathbf{W}^t\mathbf{V}. \tag{18}$$

Стандартизованная система регрессионных уравнений по сравнению с исходной системой (2) имеет ряд преимуществ. Первое преимущество связано с использованием относительных величин вместо абсолютных. Величины различных предикторов x_j могут иметь разные размерности и диапазоны изменения. В то время как стандартизованые предикторы w_j всегда безразмерны. Во-вторых, влияние каждого предиктора не обязательно связано с его абсолютной величиной x_j, а скорее с относительной величиной его изменчивости w_j.

2. Ридж-регрессия и LASSO

Нередко приходится сталкиваться с ситуацией, когда матрица $\mathbf{X}^t\mathbf{X}$ «близка» к вырожденной. Тогда говорят о наличии *мультиколлинеарности*. В таких ситуациях МНК-оценки формально существуют, но обладают «плохими» статистическими свойствами. Небольшое изменение исходных статистических данных (добавление или изъятие небольшой порции наблюдений) приводит к существенному изменению оценок коэффициентов регрессионной модели, вплоть до изменения их знаков.

Для исследования мультиколлинеарности регрессионной модели используют коэффициенты увеличения дисперсии VIF_j, $j = \overline{1,k}$ (см. [2]). На практике, в том случае когда $VIF_j > 5$ (а тем более когда $VIF_j > 10$), по крайней мере для одного j, считается, что мультиколлинеарность регрессионной матрицы $\mathbf{X}$ велика.

Методы регрессии Ридж и LASSO осуществляют регуляризацию параметров и позволяют преодолеть некоторые недостатки метода наименьших квадратов.

Точность прогноза. Если зависимость между переменными близка к линейной, и число предикторов k значительно меньше объема выборки ($k \ll n$), скорее всего простой метод наименьших квадратов будет давать хорошие результаты. Однако, если k не намного меньше размера выборки n, растет дисперсия прогноза, а его точность падает. Если же $k > n$, метод наименьших квадратов не дает единственного решения и дисперсия прогноза становится бесконечной. Методы регуляризации зачастую позволяют добиться уменьшения десперсии прогноза за счет незначительного увеличения его смещенности. В результате точность прогноза растет.

Интерпретируемость модели. Зачастую модель содержит большое число k предикторов, многие из которых могут не оказывать влияния на значение отклика. Если исключить такие переменные из выборки, модель будет легче интерпретировать. Методы регрессии Ридж и LASSO представляют в этом смысле альтернативу известной технике «выбора подмножества» для уменьшения числа предикторов. В результате применения этих методов коэффиицент (вес) при некоторых предикторах линейной модели приближается к нулю (или становится равным нулю).

3. Ридж-регрессия

Ридж-оценка неизвестного вектора B по стандартизованным наблюдениям $\{\mathbf{W}, \mathbf{V}\}$ определяется равенством

$$\widetilde{B_\lambda} := \left(\mathbf{W}^t\mathbf{W} + \lambda\mathbb{I}\right)^{-1}\mathbf{W}^t\mathbf{V}, \tag{19}$$

где $\mathbb{I}$ – единичная матрица. Параметр $\lambda > 0$ называют *параметром регуляризации*.

Для координатной записи Ридж-оценки будем использовать обозначение

$$\widetilde{B}_\lambda \;=\; \left[\tilde{\beta}_j(\lambda)\right]_{k-1}. \tag{20}$$

Добавление к диагональным элементам матрицы $\mathbf{W}^t\mathbf{W}$ «гребня» λ превращает «плохо обусловленную» матрицу $\mathbf{W}^t\mathbf{W}$ в «хорошо обусловленную» матрицу $(\mathbf{W}^t\mathbf{W} + \lambda\mathbb{I})$. Тем самым удается избежать неприятностей связанных с обращением «плохо обусловленных» матриц. Однако в отличии от МНК-оценки $\widehat{B}$ Ридж-оценка $\widetilde{B}_\lambda$ является смещенной оценкой.

Можно показать, что Ридж-оценка $\widetilde{B}_\lambda$ является решением следующих эквивалентных экстремальных задач.

$$\begin{aligned} &1^\circ \;\; \| \mathbf{V} - \mathbf{W}B \|^2 + \lambda \| B \|^2 \longmapsto \min, \\ &2^\circ \text{ Для любого } \lambda > 0 \text{ существует } t(\lambda) > 0, \text{ при котором} \\ &\| \mathbf{V} - \mathbf{W}B \|^2 \longmapsto \min, \text{ при } \| B \|^2 \le t(\lambda). \end{aligned} \tag{21}$$

Таким образом, Ридж-оценка является МНК-оценкой с ограничением нормы возможных решений (сферическое ограничение на параметры).

4. LASSO

Оценка LASSO[1] $\widetilde{B}_\lambda$ (см. [3], [4]) является решением следующих эквивалентных экстремальных задач по стандартизованным наблюдениям $\{\mathbf{W}, \mathbf{V}\}$.

$$\begin{aligned} &1^\circ \;\; \| \mathbf{V} - \mathbf{W}B \|^2 + \lambda \| B \|_1 \longmapsto \min, \\ &2^\circ \text{ Для любого } \lambda > 0 \text{ существует } t(\lambda) > 0, \text{ при котором} \\ &\| \mathbf{V} - \mathbf{W}B \|^2 \longmapsto \min, \text{ при } \| B \|_1 \le t(\lambda), \end{aligned} \tag{22}$$

где $\| B \|_1 = \sum_{j=1}^{k-1} |\beta_j|$.

В отличие от Ридж-регрессии, LASSO имеет несколько другое ограничение для β_j, $j = \overline{1, k-1}$. Коэффициент λ умножается на ℓ_1-норму вектора $(\beta_1, \ldots, \beta_{k-1})$, тогда как в Ридж-регрессии используется ℓ_2-норма.

Положительным (в плане интерпретируемости модели) результатом такой замены нормы является тот факт, что LASSO, в отличие от Ридж-регрессии, не только осуществляет регуляризацию, но и приравнивает некоторые из коэффициентов к нулю при достаточно большом значении λ. То есть LASSO дополнительно осуществляет выбор подмножества переменных, что позволяет легче интепретировать модель.

Как и в случае Ридж-регрессии, различные значения λ генерируют разные векторы $\widetilde{B}_\lambda$. Поэтому важно правильно выбрать подходящее значение λ. В разделе 5 мы говорим о технике кросс-валидации, которая может использоваться для этих целей.

5. Выбор значения λ для Ридж и LASSO. Кросс-валидация.

Техника кросс-валидации (ее еще называют «скользящим контролем» или «перекрестной проверкой») предоставляет возможность найти «подходящее» значение λ по имеющимся наблюдениям (см. [3], [4], [5]). Здесь «подходящее» нужно понимать в том смысле, что мы стремимся подобрать λ, которое позволит прогнозировать значения отклика с наибольшей точностью. Очевидно, что слишком малые значения λ могут приводить к переобучению, когда модель будет подстраиваться под шум, присутствующий в исходных наблюдениях. Слишком большие λ, напротив, могут помешать определению основной закономерности. В обоих случаях будет расти ошибка, вычисленная для наблюдений, не входящих в исходную выборку (контрольных наблюдений).

Процедура кросс-валидации искусственно разделяет исходные наблюдения на контрольную и обучающую выборки. По обучающей выборке строятся оценки регрессионных коэффициентов. Далее полученная регрессионная модель валидируется на контрольной выборке.

Более точно алгоритм выглядит следующим образом. Сначала вся выборка случайно разделяется на Q блоков. Один из блоков рассматривается как контрольная выборка, а остальные $Q - 1$ в совокупности составляют обучающую выборку. На практике Q обычно выбирают равным 5 или 10. Далее берется вектор значений $\lambda = [\lambda_s]$ с некоторым шагом, и для

[1] Least Absolute Shrinkage and Selection Operator.

каждого из значений λ_s по обучающей выборке строится регрессионная модель. Для каждой модели вычисляется ошибка прогноза, то есть сумма квадратов остатков регрессии

$$RSS^q_{\lambda_s} = \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k-1} \widehat{b}_j(q, \lambda_s)\, x_{ij} \right)^2, \tag{23}$$

где $q = \overline{1, Q}$ есть номер блока, выбранного в качестве контрольной выборки. Далее вычисляется среднее значение этой ошибки по всем блокам:

$$MSE_{\lambda_s} = \frac{1}{Q} \sum_{q=1}^{Q} RSS^q_{\lambda_s}. \tag{24}$$

В качестве подходящего λ выбирается такое λ_s, при которой MSE_{λ_s} будет минимальной.

Другой популярный вариант выбора λ руководствуется правилом «одной стандартной ошибки» (см., например, [6]). Для каждого значения MSE_{λ_s} вычисляется стандартная ошибка среднего, а затем выбирается наибольшее λ_s, при котором MSE_{λ_s} превосходит минимальное значение MSE не более, чем на одну стандартную ошибку. Таким образом мы получаем более «регулярную» модель, при этом «ухудшая» MSE не более, чем на одну стандартную ошибку.

6. Ридж и LASSO на примере тестовых данных

Теперь применим методы регрессии МНК, Ридж и LASSO к реальным данным. Для этого мы будет использовать результаты наблюдений различных характеристик вина, которые можно найти на общедоступном ресурсе UCI Machine Learning Repository ([7]). Набор данных называется Wine Quality и представляет собой значения измерений различных характеристик белого и красного вин Винью Верде («Vinho Verde») из провинции Минью на севере Португалии (см. [8], [9]). В этой работе мы рассматриваем только данные о красном вине. Предикторами являются различные физико-химические свойства вина, такие как кислотность, плотность, содержание сахара и алкоголя. Отклик же есть оценка качества вина по шкале от 0 до 10.

Мы построим регрессионные модели зависимости оценки качества вина от его физико-химических характеристик методами наименьших квадратов, Ридж и LASSO. Для проведения всех вычислений мы использовали математический пакет R и в частности его библиотеку glmnet, которая содержит в себе функции для построения регрессионной модели методами Ридж и LASSO.

Прежде всего нужно загрузить в R библиотеку glmnet.

```
library(glmnet)
```

Данные о вине хранятся в формате CSV. Для загрузки данных в пакет R можно использовать команду read.csv2. Для дальнейшего использования данные из файла мы сохранили в переменную wine.

```
wine = read.csv2("winequality-red.csv", na.strings="N/A", dec=".")
```

Для наблюдений выделили 12 параметров вина, 11 из которых представляет собой значения различных физических и химических показателей, которые мы будем рассматривать как предикторы, а 12-й есть оценка качества вина, целое число от 0 до 10, которое мы считаем откликом. Список названий столбцов для матрицы wine можно посмотреть с помощью команды names (см. таблицу 1).

```
names(wine)
```

Таблица 1. Список предикторов (названия столбцов) матрицы wine

1. fixed.acidity	2. volatile.acidity	3. citric.acid	4. residual.sugar	5. chlorides	6. free.sulfur.dioxide
7. total.sulfur.dioxide	8. density	9. pH	10. sulphates	11. alcohol	12. quality

Таблица данных для красного вина содержит значения для 1599 наблюдений. В этом можно убедиться, используя команду dim для определения размерности матрицы.

```
dim(wine)
[1] 1599   12
```

Для оценки качества регрессионных моделей мы случайным образом разделим выборку пополам на обучающую и контрольную. Так как разбиение выборки на подмножества производятся случайно, результаты будут отличаться от раза к разу. Чтобы избежать этого и чтобы читатель смог повторить приведенные вычисления, будем использовать команду set.seed с фиксированным значением параметра 1.

```
wine.pred.names = names(wine)[1 : length(names(wine)) -1]
wine.pred = wine[, wine.pred.names]

set .seed(1)
train=sample(1:nrow(wine.pred), nrow(wine.pred)/2)
test=(-train)

x = model.matrix(~., wine.pred[train, ]) [, -1]
y = wine$quality[train]
x.test = model.matrix(~., wine.pred[test, ]) [, -1]
y.test = wine$quality[test]
```

Матрицу значений предикторов для обучающей выборки мы будем хранить в переменной x, а отклик в переменной y. Для контрольной выборки мы используем переменные x.test и y.test.

Регрессионную МНК-модель можно построить, используя команду lm. МНК-оценки параметров приведены в первой строке таблицы 3.

```
lm.mod = lm(y~., data = data.frame(x))
```

Команда summary позволяет получить удобную сводку различных характеристик регрессионной модели. В частности, коэффициент детерминации R^2, который в нашем случае равен 0.3334006.

```
summary(lm.mod)$r.squared
[1] 0.3334006
```

Для анализа мультиколлинеарности вычислим значения коэффициентов увеличения дисперсии VIF_j. Для этого будем использовать команду vif из библиотеки car.

```
library(car)
vif(lm.mod)
```

Таблица 2. Коэффициенты увеличения дисперсии VIF_j, $j = \overline{1, 11}$ для красного вина

VIF_1	VIF_2	VIF_3	VIF_4	VIF_5	VIF_6	VIF_7	VIF_8	VIF_9	VIF_{10}	VIF_{11}
7.175034	1.777550	3.424212	1.673862	1.513932	2.043192	2.235502	5.874435	3.239267	1.447768	2.852950

Значения VIF_j приведены в таблице 2. Некоторые из этих коэффициентов довольно велики, как например VIF_1 и VIF_8. Это говорит о значительной, но некритичной мультиколлинеарности: все $VIF_j < 10$.

Перейдем теперь к Ридж-регрессии. Как уже было сказано, для Ридж-регрессии различные значения λ будут генерировать различные наборы $\widehat{b_i}$. Используя пакет R, мы можем посмотреть, как для данной выборки оценки $\widehat{b_i}$ меняются в зависимости от λ.

Для этого сначала выберем некоторый диапазон значений λ и сгенерируем регрессионную модель для всех λ из этого диапазона, используя удобную функцию glmnet из одноименной библиотеки. Заметим, что функция glmnet проводит стандартизацию данных: как предикторы, так и отклик центрируются и нормируются перед построением регрессионной модели. При этом результат всегда выдается в исходных единицах измерения, поэтому, в частности, мы получим ненулевое значение свободного члена $\widehat{b_0}$.

```
lambda.grid = 10^seq(5, -2, length=100)
ridge.mod = glmnet(x, y, alpha=0, lambda=lambda.grid)
```

В данном случае λ принимает 100 различных значений от 10^{-2} до 10^5, где степень меняется с постоянным шагом $\frac{7}{99}$. Зависимость оценок коэффициентов от λ можно изобразить, используя команду

```
plot(ridge.mod, xvar='lambda')
```

Вывод этой команды представлен на рисунке (1) слева. Как можно видеть, при увеличении λ значения оценок $\widehat{b_i}$ «стягиваются» к нулю, то есть норма вектора оценок $\|\widehat{b}(\lambda)\|_2$ уменьшается. При этом отдельные коэффициенты могут на некотором промежутке возрастать с увеличением λ (как, например, коэффициент *density*, обозначенный красным).

В верхней части графика для соответствующего значения $\log \lambda$ с некоторым шагом показано количество ненулевых оценок $\widehat{b_i}$, получившихся в регрессионной модели с использованием данного λ. Как мы видим, для Ридж-регрессии это число постоянно на всем графике и равно исходному числу предикторов модели. Таким образом, хотя Ридж-регрессия и позволяет существенно приблизить к нулю оценки коэффициентов b_i, даже при очень больших λ они не обращаются в нуль.

Здесь интересно сравнить значения оценок коэффициентов $\widehat{b_i}$ для тех же λ с оценками, полученными методом LASSO.

Снова воспользуемся функцией glmnet, чтобы построить регрессионную модель методом LASSO для тех же данных и того же вектора lambda.grid. Единственное, что необходимо поменять, это значение параметра alpha с 0 на 1.

```
lasso.mod = glmnet(x, y, alpha=1, lambda=lambda.grid)
plot(lasso.mod, xvar='lambda')
```

Вывод команды plot показан на рисунке (1) справа. Как видим, метод LASSO так же «стягивает» к нулю значения коэффициентов с ростом λ. Однако в данном случае, оценки не просто приближаются к нулю, они становятся равны нулю при некотором значении λ. Это можно проследить по числам вверху графика, которые показывают количество ненулевых коэффициентов в полученной модели. При $\log \lambda = -5$ число ненулевых коэффициентов 7, а при $\log \lambda = 0$ мы уже получаем нулевую модель. То есть метод LASSO при достаточно больших λ производит выбор переменных, существенных для данной модели.

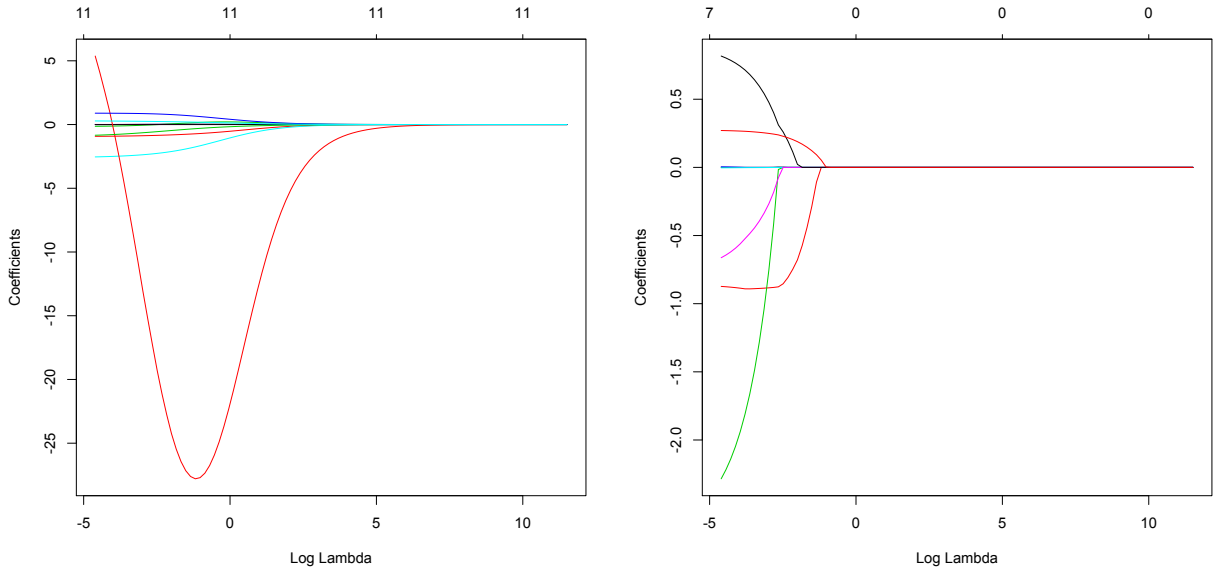

Рис. 1. Зависимость значений оценок коэффициентов β_i, полученных методами Ридж (слева) и LASSO (справа), для красного вина от $\log \lambda$. В верхней части графика показано количество ненулевых коэффициентов $\widehat{b_i}$ в полученной регрессионной модели.

Используя пакет R, мы также можем произвести выбор подходящего значения λ методом кросс-валидации. Для этого можно используется команда cv.glmnet пакета glmnet. Так как разбиение выборки на подмножества в методе кросс-валидации производятся случайно, мы снова используем команду set.seed, чтобы обеспечить воспроизводимость результатов. И здесь команды для Ридж и LASSO отличаются только значением параметра alpha.

```
set.seed(1)
ridge.cv.out = cv.glmnet(x, y, alpha=0)
set.seed(1)
lasso.cv.out = cv.glmnet(x, y, alpha=1)
```

Результаты кросс-валидации на графике, полученные с помощью команды plot, представлены на рисунке (2).

```
plot(ridge.cv.out)
plot(lasso.cv.out)
```

На графиках показана зависимость средней квадратичной ошибки предсказания MSE_λ от $\log \lambda$. Вертикальные отрезки в каждой точке добавляют и отнимают от MSE_λ одну стандартную ошибку. Одна из вертикальных пунктирных линий показывает положение минимума MSE. Вторая пунктирная линия обозначает точку, выбранную по «правилу одной стандартной ошибки».

Пользуясь результатами кросс-валидации, мы можем определить подходящие значения λ для каждого из методов.

```
ridge.bestlam = ridge.cv.out$lambda.min
ridge.lam1se = ridge.cv.out$lambda.1se
lasso.bestlam = lasso.cv.out$lambda.min
lasso.lam1se = lasso.cv.out$lambda.1se
```

В нашем случае для Ридж регрессии $ridge.bestlam = 0.05257397$, $ridge.lam1se = 0.6481565$. Для LASSO соответственно $lasso.bestlam = 0.01056334$, $lasso.lam1se = 0.06186968$.

Полученные значения коэффициентов регрессионной модели для каждого из этих значений *lambda* приведены в

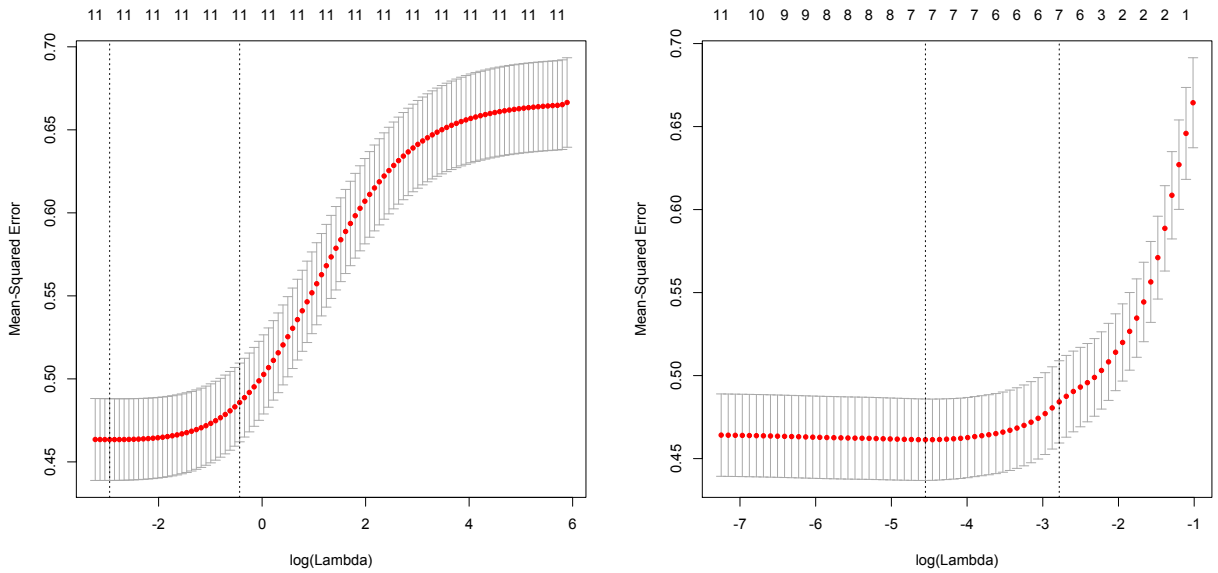

Рис. 2. Зависимость значения средней квадратичной ошибки для моделей, полученных методами Ридж (слева) и LASSO (справа), для красного вина от $\log \lambda$. В верхней части графика показано количество ненулевых коэффициентов $\widehat{b_i}$ в полученной регрессионной модели. Пунктирной линией отмечены значение $\log \lambda$, при котором функция минимальна, и значение, отличающееся от минимума на одну стандартную ошибку.

таблице 3. Чтобы вычислить эти значения в R, нужно построить регрессионную модель с выбранными значениями λ и вывести коэффициенты с помощью команды coef.

```
ridge.mod.best = glmnet(x, y, alpha=0, lambda=ridge.bestlam)
coef(ridge.mod.best)
ridge.mod.1se = glmnet(x, y, alpha=0, lambda=ridge.lam1se)
coef(ridge.mod.1se)

lasso.mod.best = glmnet(x, y, alpha=1, lambda=lasso.bestlam)
coef(lasso.mod.best)
lasso.mod.1se = glmnet(x, y, alpha=1, lambda=lasso.lam1se)
coef(lasso.mod.1se)
```

Таблица 3. Оценки параметров для красного вина, полученные различными методами регрессии

Метод	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}
МНК	-8.915	-0.013	-0.942	-0.164	-0.005	-2.599	0.007	-0.004	14.973	-0.929	0.893	0.294
$Ridge_{\lambda=0.05257397}$	18.610	0.015	–0.876	-0.053	0.006	-2.364	0.005	-0.004	-13.484	-0.638	0.858	0.251
$Ridge_{\lambda=0.6481565}$	29.906	0.0219	-0.611	0.203	0.008	-1.316	0.001	-0.002	-25.305	-0.217	0.504	0.150
$LASSO_{\lambda=0.01056334}$	5.179	0	-0.874	0	0	-2.264	0.004	-0.003	0	-0.657	0.8122	0.270
$LASSO_{\lambda=0.06186968}$	3.891	0.001	-0.880	0	0	-0.345	0	-0.001	0	-0.164	0.385	0.243

Как мы можем видеть, как для Ридж, так и для LASSO, при переходе от *bestlam* к $lam1se$ модель становится более «регулярной» в том смысле, что усиливается «стягивающая» способность регрессии. В случае LASSO количество нулевых элементов не изменилось, хотя и изменился их набор. Элемент b_1, который был равен нулю при $\lambda = lasso.bestlam = 0.01056334$ снова становится ненулевым для $\lambda = lasso.lam1se = 0.05980285$, зануляется элемент b_{10}.

Теперь сравним суммы квадратов ошибок *RSS* для каждой из моделей. Мы будем вычислять RSS сначала на обучающей выборке, а затем на контрольной выборке.

```
y.lm.train = predict(lm.mod, newdata = data.frame(x))
sum((y.lm.train - y)^2)
y.lm.test = predict(lm.mod, newdata = data.frame(x.test))
sum((y.lm.test - y.test)^2)

y.ridge.best.train = predict(ridge.mod.best, newx=x)
sum((y.ridge.best.train - y)^2)
y.ridge.best.test = predict(ridge.mod.best, newx=x.test)
sum((y.ridge.best.test - y.test)^2)

y.ridge.1se.train = predict(ridge.mod.1se, newx=x)
sum((y.ridge.1se.train - y)^2)
```

```
y.ridge.1se.test = predict(ridge.mod.1se, newx=x.test)
sum((y.ridge.1se.test - y.test)^2)

y.lasso.best.train = predict(lasso.mod.best, newx=x)
sum((y.lasso.best.train - y)^2)
y.lasso.best.test = predict(lasso.mod.best, newx=x.test)
sum((y.lasso.best.test - y.test)^2)

y.lasso.1se.train = predict(lasso.mod.1se, newx=x)
sum((y.lasso.1se.train - y)^2)
y.lasso.1se.test = predict(lasso.mod.1se, newx=x.test)
sum((y.lasso.1se.test - y.test)^2)
```

Таблица 4. Значения RSS на обучающей (RSS_{train}) и контрольной (RSS_{test}) выборках для различных методов регрессии

RSS	МНК	$Ridge_{\lambda=0.05257397}$	$Ridge_{\lambda=0.6481565}$	$LASSO_{\lambda=0.01056334}$	$LASSO_{\lambda=0.06186968}$
RSS_{train}	354.7427	355.846	380.3165	356.2117	376.4762
RSS_{test}	318.1009	316.3684	342.7885	315.7833	328.3998

Значения ошибок для всех методов приведены в таблице 4. Как видим, RSS на обучающей выборке предсказуемо минимальна у МНК-модели, так как в отличие от методов Ридж и LASSO, МНК не накладывает ограничений на коэффициенты b_i. RSS у моделей Ридж и LASSO вновь предсказуемо больше при выборе λ по правилу одной стандартной ошибки. Интересно, что на контрольной выборке значения ошибки расположены иначе: минимальное значение RSS у LASSO при $\lambda = lasso.bestlam$, чуть больше у Ридж при $\lambda = ridge.bestlam$ и только третью позицию занимает МНК. В данном случае Ридж и LASSO работают лучше МНК на контрольной выборке.

7. Заключение

Результаты статистического анализа данных Wine Quality ([8], [9]) позволяют сделать следующие выводы об использовании регрессионных методов МНК, Ридж и LASSO:

- Методы Ридж и LASSO «стягивают» к нулю оценки коэффициентов $\widehat{b_i}$ в том смысле, что уменьшается норма вектора оценок при увеличении λ.
- Ридж-регрессия не обращает в нуль коэффициенты $\widehat{b_i}$ даже при больших λ.
- LASSO, в отличие от МНК и Ридж-регрессии, осуществляет выбор подмножества переменных, то есть некоторые коэффициенты $\widehat{b_i}$ обращаются в нуль, что упрощает интерпретацию результатов регрессионного анализа.
- RSS на обучающей выборке для МНК меньше, чем для Ридж и LASSO.
- В рассмотренных примерах при оптимальном выборе λ RSS на контрольной выборке для МНК больше, чем для Ридж и LASSO.
- По сравнению с RSS на обучающей выборке, использование контрольной выборки для подсчета RSS для найденных регрессионных моделей позволяет получить более адекватную оценку качества этих моделей.

8. Благодарности

Работа выполнена при поддержке РФФИ (грант № 16-41-630-676, грант № 16-01-00184А).

Литература

[1] Yan, X. Regression Analysis: Theory and Computing [Text] / X. Yan, X. G. Su. — [S. l.] : World Scientific Publishing Co. Pte. Ltd., 2009.
[2] Kutner, M. H. Applied Linear Regression Models [Text] / M. H. Kutner, C. J. Nachtsheim, J. Neter. — 4th edition. — [S. l.] : McGraw-Hill Irwin, 2004.
[3] Hastie, T. Statistical Learning with Sparsity. The Lasso and Generalizations [Text] / T. Hastie, R. Tibshirani, M. Wainwright. — [S. l.] : Chapman & Hall, 2015.
[4] Efron, B. Computer Age Statistical Inference: Algorithms, Evidence and Data Science [Text] / B. Efron, T. Hastie. — [S. l.] : Institute of Mathematical Statistics Monographs, 2016.
[5] An Introduction to Statistical Learning with Applications in R [Text] / G. James, D. Witten, T. Hastie, R. Tibshirani. — [S. l.] : Springer, 2013.
[6] Hastie, T. The elements of statistical learning: Data Mining, Inference, and Prediction [Text] / T. Hastie, R. Tibshirani, J. Friedman. — 2nd edition. — [S. l.] : Springer, 2009.
[7] Lichman, M. UCI Machine Learning Repository [Text]. — 2013. — URL: http://archive.ics.uci.edu/ml.
[8] Cortez, P. UCI Machine Learning Repository — Wine Quality Data Set [Electronic resource]. — 2009. — URL: http://archive.ics.uci.edu/ml/datasets/Wine+Quality.
[9] Modeling Wine Preferences by Data Mining from Physicochemical Properties [Text] / P. Cortez, A. Cerdeira, F. Almeida [et al.] // Decision Support Systems. — 2009. — Vol. 47, no. 4. — P. 547–553.