Национальный исследовательский университет

Высшая школа экономики

Кафедра математической экономики и эконометрики

Курсовая работа
на тему «Сравнение Ridge Regression и LASSO в условиях мультиколлинеарности»

Студент группы 312И
Васильев Сергей

Научный руководитель
Демешев Борис Борисович

Москва, 2013

Содержание

1 Введение

Одной из часто встречаемых проблем в статистике является мультиколлинеарность, когда регрессоры линейно зависимы друг от друга. Проблемой её считают из-за крупных стандартных ошибок оценок (широкие доверительные интервалы и ошибки второго рода как следствие), плюс при совершенной мультиколлинеарности возникает ситуация, когда МНК-оценки получить нельзя вообще (из-за необратимости матрицы X^TX). Один из способов борьбы с ней – это регуляризация. В данной работе рассматривается Ridge Regression (также известная как l_2 регуляризация и регуляризация Тихонова) и LASSO (также известное как l_1 регуляризация), методы уменьшающие размер коэффициентов путём наложения «штрафа» на размер RSS. Кроме того, будет рассмотрено как эти методы ведут себя при наличии проблемы переспецификации модели, которой зачастую сопутствует и мультиколлинеарность.

В данной работе будет приведена теория LASSO и Ridge Regression, далее они будут применены к данным (синтетическим полиномам и реальным данным финансового сектора), и будут представлены упражнения, которые могут быть применены для контроля освоения методов.

2 Теория

2.1 Модель

Рассматривается классическая линейная модель: $Y = X\beta + \varepsilon$, где Y – вектор наблюдений объясняемой переменной размерности $n\times1$, X – матрица наблюдений объясняющих переменных размерности $n \times p$, β – вектор коэффициентов размерности $p \times 1$, а ε - вектор ошибок модели размерности $n \times 1$. Кроме того, предполагается что $\varepsilon \backsim N(0.\sigma^2)$, то есть ошибка является Гауссовым шумом с нулевым матожиданием и дисперсией равной σ^2.

Далее, в рамках данной работы будут рассматриваться модели (а точнее наборы данных), в которых присутствует мультиколлинеарность, то есть линейная связь между регрессорами. Есть две проблемы, которые возникают из-за неё. Первая и действительно серьезная проблема возникает, когда мы сталкиваемся с совершенной мультиколлинеарностью, когда мы можем в явном виде записать зависимость между регрессорами. В данном случае мы просто не получим оценки МНК, так как матрица X^TX становится необратимой. Вторая проблема заключается в том, что при наличии мультиколлинеарности, увеличивается дисперсия оценок.

Основной признак наличия мультиколлинеарности – это диссонанс F-статистики и индивидуальных t-статистик (регрессия в целом значима, а регрессоры по отдельности незначимы). В качестве более формального критерия будет использован VIF (Variance Inflation Factor), измеряющий на сколько выросла дисперсия коэффициента из-за наличия коллинеарности. Рассчитывается он следующим образом:

$$VIF = \frac{1}{1 - R_j^2} \tag{1}$$

где R_j^2 – коэффициент детерминации для вспомогательной регрессии следующей модели:

$$X_j = \alpha_0 + \alpha_1 X_1 + ... + \alpha_{j-1} X_{j-1} + \alpha_{j+1} X_{j+1} + ... + X_p + \varepsilon \tag{2}$$

Традиционно считается, что мы сталкиваемся с серьезной мультиколлинеарностью, когда $VIF > 5$.

2.2 Ridge Regression

Идея Ridge Regression уходит корнями в первую проблему, возникающую из-за мультиколлинеарности – необратимости матрицы X^TX. Постановка задачи выглядит следующим образом:

$$\min_{\beta}(Y - X\beta)^T(Y - X\beta) \tag{3}$$

при условии:

$$\sum_{j=1}^{p} \beta_j^2 \leqslant t \tag{4}$$

Можно видеть, что формулировка похожа на МНК, но теперь добавляется ограничение на размер коэффициентов. Запишем лагранжиан:

$$L = (Y - X\beta)^T(Y - X\beta) + \lambda(\sum_{j=1}^{p} \beta_j^2 - t) \tag{5}$$

Так как условие первого порядка $\partial L/\partial\beta = 0$, то постановку задачи можно переписать постановку Ridge Regression следующим образом (также известном как $PRSS$ – Penalised Residual Sum of Squares):

$$\min_{\beta}(Y - X\beta)^T(Y - X\beta) + \lambda\sum_{j=1}^{p} \beta_j^2 \tag{6}$$

Интуитивный смысл RR в том, что она уменьшает коэффициенты (точнее коэффициенты «сжимаются» к нулю и друг к другу) с помощью параметра λ, который и определяет степень уменьшения или «сжатия» (чем больше λ, тем сильнее). Такая задача не представляет вычислительной сложности и мы можем получить оценки коэффициентов в явном виде:

$$\hat{\beta}^{ridge} = (X^TX + \lambda I)X^TY \tag{7}$$

где I – диагональная единичная матрица. За счет того, что к матрице X^TX прибавляется диагональная матрица λI с ненулевыми элементами, мы всегда можем получить оценки, даже в случае совершенной мультиколлинеарности.

Другим важным и полезным свойством Ridge Regression является то, что всегда существует такая λ, что MSE для Ridge Regression будет меньше чем MSE для МНК. Это свойство также известно как «Теорема о существовании» [1, стр. 62].

Наконец, в случае большого числа коррелированных регрессоров, один огромный положительный коэффициент может «гаситься» другим не менее огромным отрицательным. Введение ограничения на размер коэффициентов избавляет нас от такой проблемы.

2.3 LASSO

Второй метод – это LASSO (least absolute shrinkage and selection operator), которое во многом похоже на Ridge Regression [2, стр. 267]. Так, постановка задачи выглядит следующим образом:

$$\min_{\beta}(Y - X\beta)^T(Y - X\beta) \tag{8}$$

при условии:

$$\sum_{j=1}^{p} |\beta_j| \leqslant t \tag{9}$$

С помощью лагранжиана можно записать это выражение следующим образом:

$$\min_{\beta}(Y - X\beta)^T(Y - X\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \tag{10}$$

Видно, что единственным отличием от RR является функциональная форма ограничения. В остальном механизм LASSO похож – мы вводим штраф на размер коэффициентов, а λ определяет степень их уменьшения. Ещё один неприятный момент: постановка LASSO – это задача квадратичного программирования и мы не можем получить выражение для оценки $\hat{\beta}^{LASSO}$ в явном виде. На практике используются апроксимационные алгоритмы такие как $LARS$.

Говоря о поведение LASSO касательно MSE, Райан Тибширани отмечает, что оно сравнимо с таковым у Ridge Regression [3, стр. 10], но найти формально строгого подтверждения мне не удалось. Тем не менее, у LASSO есть очень полезная особенность, связанная с формой ограничения – оно само может выбирать спецификацию модели, то есть отделять «существенные» коэффициенты от «несущественных». Почему это так?

Для простоты рассмотрим двухфакторную модель: $Yi = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \varepsilon_i$, тогда мы можем расписать ограничения (4) и (9) в следующем виде:

$$RR : \beta_1^2 + \beta_2^2 \leqslant t \tag{11}$$

$$LASSO : |\beta_1| + |\beta_2| \leqslant t \tag{12}$$

То есть ограничение для RR на плоскости (β_1, β_2) будет выглядеть как окружность, в то время как ограничение LASSO представляет собой ромб. Линии уровня для RSS будут эллипсами. Представим эту задачу графически [4, стр. 71]:

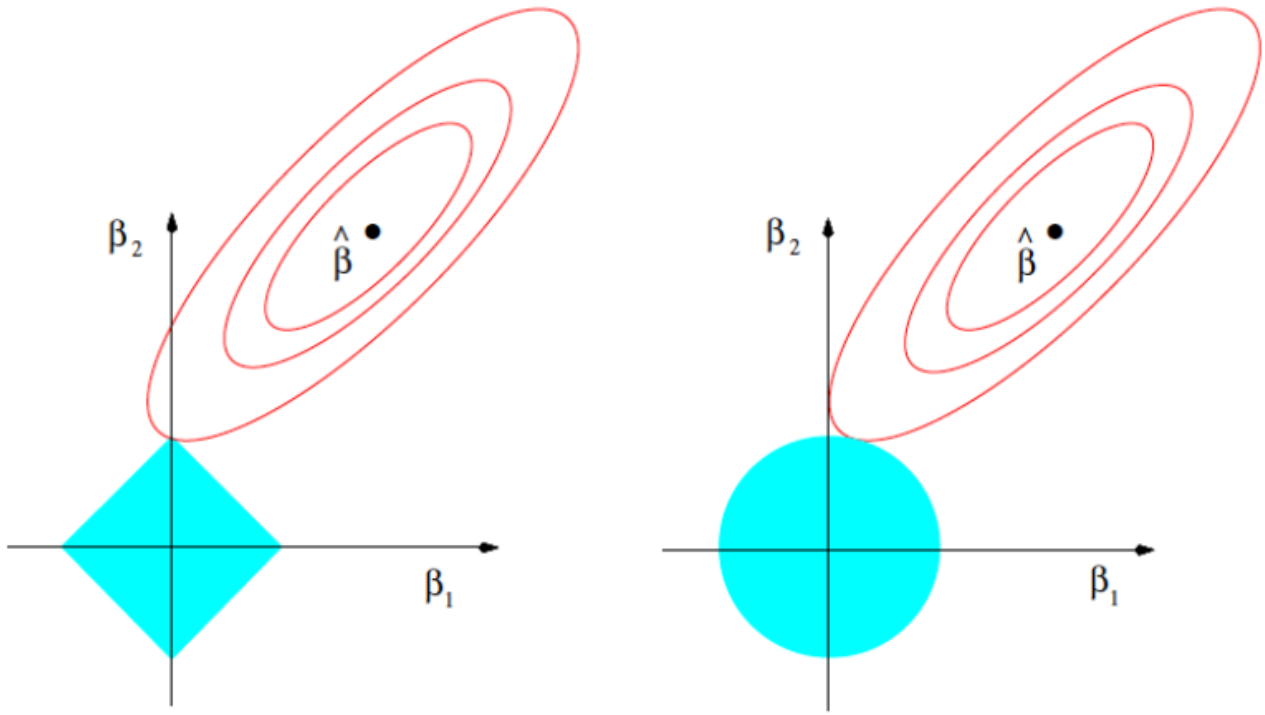

Рис. 1: Решения Ridge Regression и LASSO

Можно видеть, что LASSO обратил коэффициент β_1 в ноль. И обобщая, стоит отметить, что RR уменьшает коэффициенты, но не обращает их в ноль (очень редко и по чистой случайно-

сти), в то время как LASSO за счет функциональной формы ограничения делает это постоянно, так как касание происходит в углу ромба, а если $p > 2$, то ограничение становится ромбоидом с множеством углов и плоских сторон, что ещё сильнее увеличивает шансы обращения коэффициентов в ноль. Польза здесь в том, что RR дает нам возможность делать лучше прогнозы (за счет меньшего MSE), а LASSO в добавок к этому выдает ясную и простую для интерпретации модель.

2.4 Выбор λ

LASSO и RR имеют общие проблемы. Первая – это отсутствие адекватных методов построения доверительных интервалов. Вторая проблема заключается в том, что нам необходимо выбирать λ. Я в своей работе для реализации LASSO и Ridge Regression буду использовать язык R и библиотеку *glmnet*[1], под авторством Тревора Хасти, Роберта Тибширани и Джерома Фридмана. Этот пакет реализует более общий метод, носящий название *«elastic net»*. Это обощение RR и LASSO со следующим ограничением на RSS:

$$(1-\alpha)\sum_{j=1}^{p}\beta_j^2 + \alpha\sum_{j=1}^{p}|\beta_j| \leqslant t \tag{13}$$

Это линейная комбинация из ограничений RR и LASSO, и чтобы реализовать сами LASSO мы будем устанавливать $\alpha = 1$. Для RR будет использован базовый пакет MASS. Выбор λ осуществляется с помощью кросс-валидации (точнее с помощью K-fold cross validation). Суть данного метода заключается в том, что набор данных разбивается на K поднаборов одинаковой длины. Затем для каждой λ выбирается один поднабор для валидации, а остальные $K-1$ используются для получения оценок. Эта процедура повторяется K раз, чтобы каждый поднабор мог быть использован для валидации. Соответственно рассчитываются (для каждого выбора поднабора для валидации) ошибки кросс-валидации в зависимости от λ (для их расчета используется MSE), а после строится агрегированная ошибка кросс-валидации через усреднение отдельных ошибок кросс-валидации. Далее, находится минимум этой агрегированной ошибки кросс-валидации, и полученное значение λ^* и выбирается.

При реализации в R, для нахождения оптимальной λ для LASSO пакет *glmnet* использует кросс-валидацию, описанную выше с количеством поднаборов равным 10 по умолчанию. Для Ridge Regregession пакетом MASS используется Generalized Cross Validation. Рассчитывается следующий показатель [5, стр. 8]:

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i(\lambda)}{1 - \frac{1}{n}tr(H)}\right)^2 \tag{14}$$

где $\hat{Y} = HY$. Далее выбирается такая λ, что это выражение минимально.

3 Примеры

В данном разделе будут применены Ridge Regression и LASSO для того, чтобы продемонстрировать свойства, описанные в предыдущем разделе. Для начала мы рассмотрим модели с полиномиальной функциональной формой, так как в них автоматически появляется мультиколлинеарность, а, кроме того, не имея априорных суждений о модели, выбрать правильную степень полинома может быть очень проблематично (так как, например, полиномы третьей и пятой степени очень похожи друг на друга). Всего мы рассмотрим три таких случая:

[1] http://cran.r-project.org/web/packages/glmnet/index.html

1. n=6, истинная степень полинома – куб
2. n=50, истинная степень полинома – квадрат, «маленькие» коэффициенты
3. n=50, истинная степень полинома – квадрат, «большие» коэффициенты

Далее мы применим данные методы к реальным данным на примере финансовых рынков. Будет рассмотрена модифицированная версия модели CAPM (Capital Asset Pricing Model), где будет использовано одновременно три рыночных индекса, которые связаны между собой, и мы можем ожидать наличие мультиколлинеарности (эти ожидания подтвердятся).

3.1 Первый полином

Начнем с самого экзотичного и самого, на мой взгляд, интересного случая. Мы сгенерируем шесть наблюдений с помощью полинома третьей степени и попробуем обсчитать полином пятой степени:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \beta_4 X_i^4 + \beta_5 X_i^5 + \varepsilon_i \tag{15}$$

Помимо того, что в полиноме автоматом возникает мультиколлинеарность (в удалении от особенности (экстремума, точки перегиба) все полиномиальные функции являют собой параллельные прямые), в данном случае огромное влияние будет оказывать проблема переспецификации, так как через N точек на плоскости всегда можно провести полином максимум N-1 степени. Начнем с МНК оценки:

```
set.seed(1897)  #установим зерно генератора для воспроизводимости
n.obs <- 6  #число наблюдений
x <- runif(n.obs)  #с помощью непрерывного равномерного распределения
# генерируем 6 наблюдений
y <- 10 + 2 * x + 1.5 * x^2 + x^3 + rnorm(n.obs)  #генерируем наблюдения объясняемой
# переменной как полином третьей степени с случайной ошибкой N(0,1)
model.ls <- lm(y ~ poly(x, 5, raw = TRUE))  #получаем оценки МНК
```

Как можно видеть на рисунке 2, с помощью метода наименьших квадратов мы получили такие оценки, что они идеально описывают наши данные ($RSS = 0$), но совершенно не годятся, ни для предсказания, ни для интерпретации. Более того, посмотрим на полученные оценки в таблице 1. Как мы можем видеть, они невообразимо огромные и одни огромные положительные коэффициенты находятся по соседству с другими огромными отрицательными, а RR и LASSO как раз вводят штраф за большой размер коэффициентов.

Начнем с Ridge Regression:

```
library(MASS)  #подключаем библиотеку с lm.ridge
model.rr <- lm.ridge(y ~ poly(x, 5, raw = TRUE), lambda = seq(0, 10, 0.01))
select(model.rr)  #пытаемся получить значение лямбды

## modified HKB estimator is Inf
## modified L-W estimator is Inf
## smallest value of GCV  at 0
```

Получить значение λ у нас не выходит, так как вспомним 14. В числителе выражение $y_i - \hat{y}_i(\lambda)$, а в случае $\lambda = 0$ оно минимально (равно и само 0). Но наша цель в данном случае – повысить адекватность оценок относительно МНК, так что возьмем любую ненулевую λ, например, равную 0.2.

```
model.rr.fin <- lm.ridge(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5), lambda = 0.2)
```

Мы решили проблему огромных коэффициентов и теперь полученные оценки (таблица 1) для Ridge Regression позволяют нам прогнозировать, но они все ещё не вносят ясности для интерпретации, так как RR распределяет вклад на все пять степеней. Посмотрев на рисунок 2, мы можем видеть, что RR-оценки уже более адекватны нашим данным.

Попробуем получить более простую для интерпретации модель с помощью LASSO. Нам нужно будет привести данные к матричному виду наши наблюдения в силу особенностей библиотеки *glmnet*.

```
library(glmnet)
X <- cbind(poly(x, 5, raw = TRUE))  #приводим к матричному виду
colnames(X) <- c("X", "X2", "X3", "X4", "X5")  #присваиваем имена столбцам
cv.las <- cv.glmnet(X, y, alpha = 1)  #осуществляем кросс-валидацию
las.lambda <- cv.las$lambda.min  #вытаскиваем оптимальное значение лямбды
model.las <- glmnet(X, y, lambda = las.lambda)
las.est <- predict(model.las, X, type = "coef")  #вытаскиваем оценки
```

Нам не повезло – LASSO выбрало полином пятой степени (занулив только коэффициент при x), что может быть объяснено тем, что полиномы третий и пятой степени очень схожи, плюс очень маленькое число наблюдений. В целом, как мы увидим и в последующих примерах, LASSO зачастую работает не очень удачно в роли «feature selection» (не удачно относительно той функциональной формы, по которой мы генерируем данные). Посмотрев на рисунок 2, можно заметить, что Ridge Regression и LASSO дали нам схожий результат.

Коэффициент	OLS	Ridge Regression	LASSO
Константа	-935.3049	9.681	9.9564
X	8618.1326	1.087	0
X^2	-29847.5347	1.165	1.639
X^3	49566.0275	1.272	2.5292
X^4	-39682.805	1.341	1.2364
X^5	12312.3857	1.377	0.491

Таблица 1: Оценки МНК, Ridge Regression и LASSO

То как меняются уравнения регрессии RR и LASSO для разных λ можно увидеть на рисунке 3 . Примечательно, что LASSO при $\lambda = 10$ обратил все коэффициенты в ноль.

Рассмотрим ещё одно свойство RR и LASSO – возможность получить оценки в случае p>n. Воспользуемся полиномом девятой степени. Считать оценки МНК не имеет смысла, так как матрица $X^T X$ необратима. Начнем с Ridge Regression:

```
model.rr.9 <- lm.ridge(y ~ poly(x, 9, raw = TRUE), lambda = seq(0, 10, 0.00001))
select(model.rr.9)

## modified HKB estimator is -1.974e-31
## modified L-W estimator is -5.08e-26
## smallest value of GCV  at 1.57

# в данном случае мы можем получить лямбду (выбираем через GCV)
model.rr.9 <- lm.ridge(y ~ poly(x, 9, raw = TRUE), lambda = 1.57042)
```

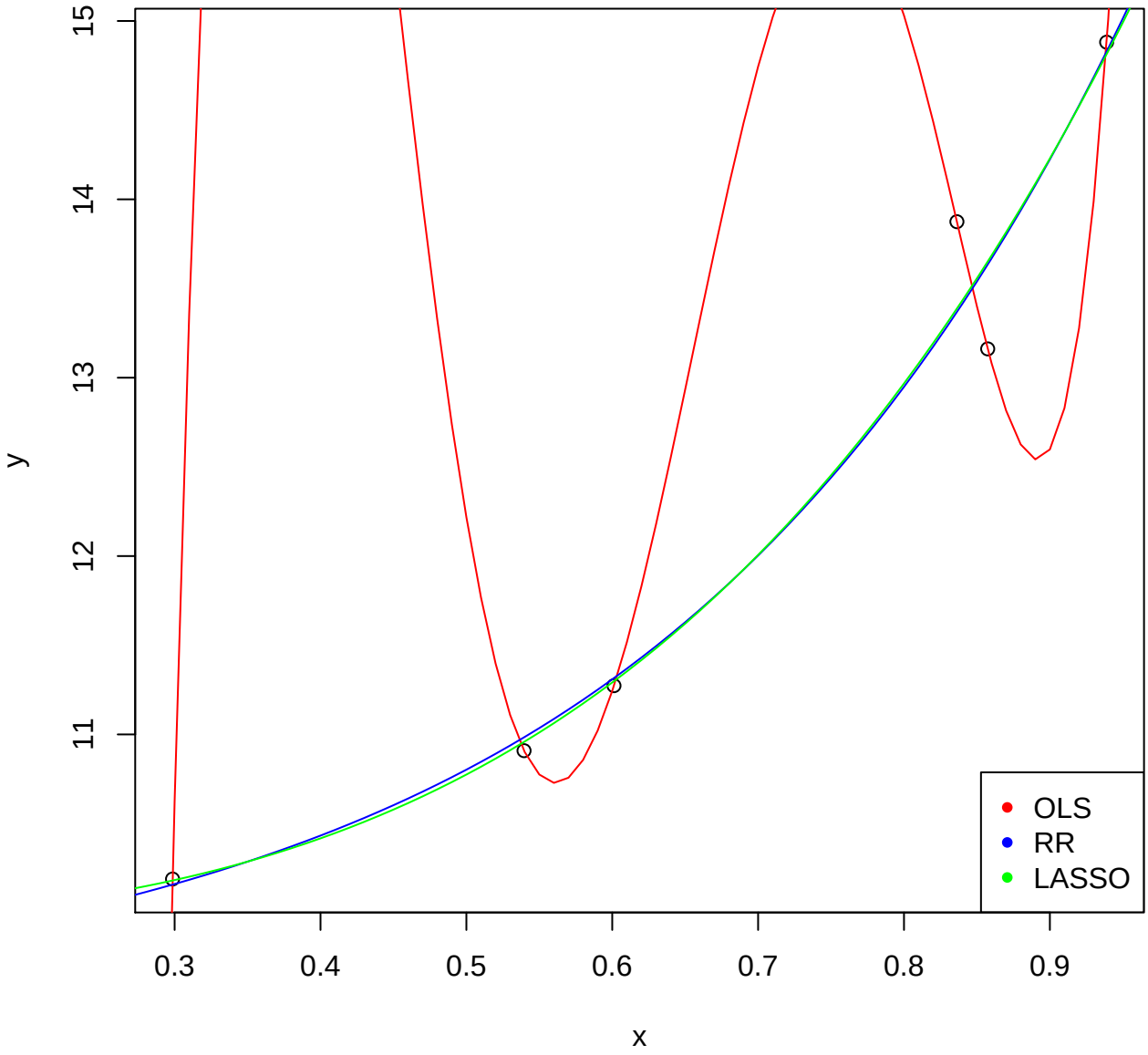

Рис. 2: Уравнения для МНК, Ridge Regression, LASSO

Полученные оценки можно увидеть в таблице 2, причем RR не обратил ни один коэффициент в ноль, а ещё сильнее уменьшил коэффициенты из предыдущей (15), перенеся долю объяснения на новые регрессоры. Теперь посмотрим на LASSO:

```
X9 <- cbind(poly(x, 9, raw = TRUE))
colnames(X9) <- c("X", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9")
cv.las.9 <- cv.glmnet(X9, y, alpha = 1)
las.lambda.9 <- cv.las.9$lambda.min
model.las.9 <- glmnet(X9, y, lambda = las.lambda.9)
las.est.9 <- predict(model.las.9, X9, type = "coef")
```

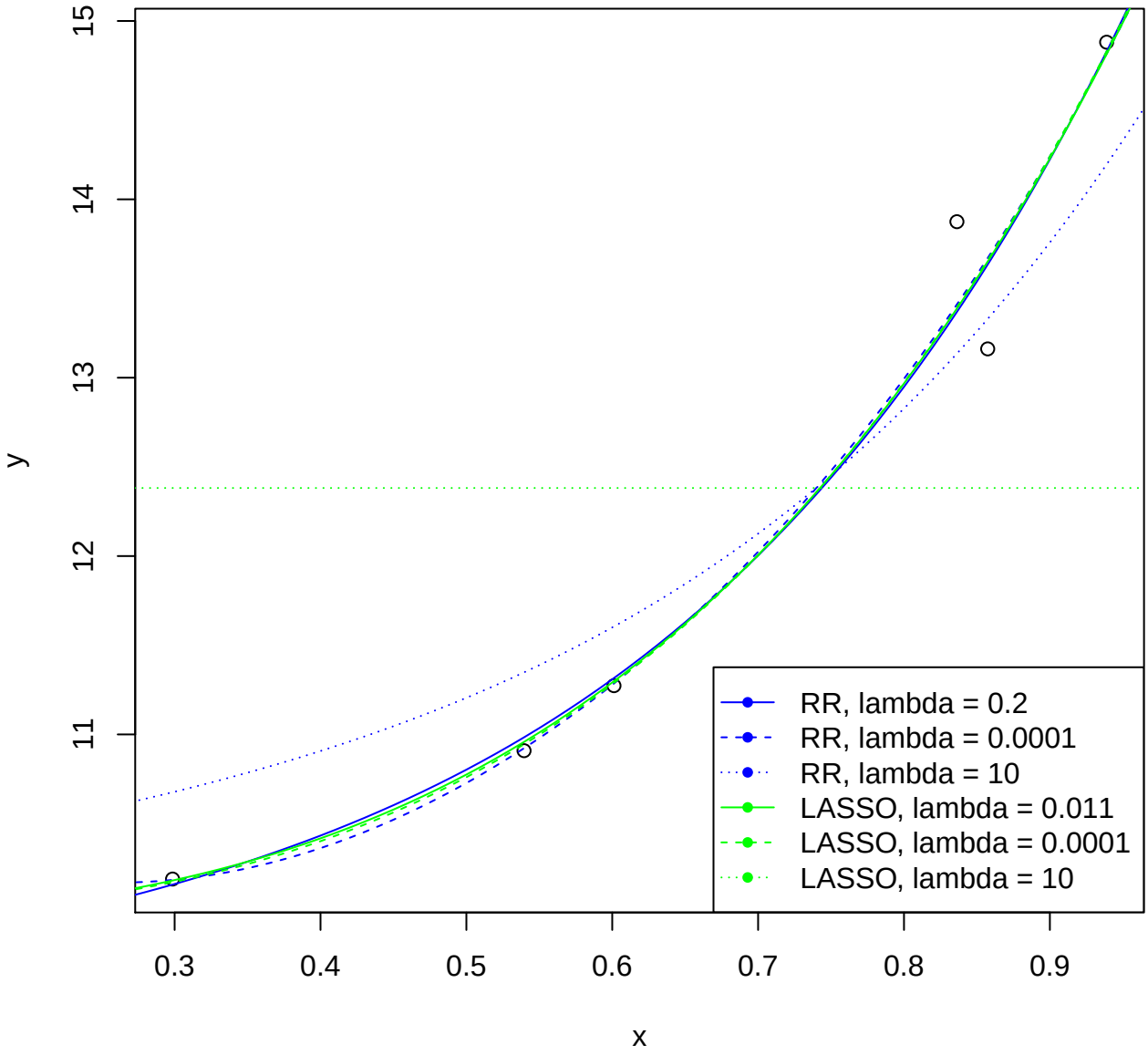

Рис. 3: Уравнения регрессии для разных λ

Коэффициент	Ridge Regression	LASSO	Коэффициент	Ridge Regression	LASSO
Константа	9.8184136	10.0704	X^5	0.682	0
X	1.2667699	0	X^6	0.619	0
X^2	0.9375685	0.8552	X^7	0.554	0
X^3	0.8217719	3.6544	X^8	0.487	0
X^4	0.7468023	1.1437	X^9	0.417	0

Таблица 2: Оценки Ridge Regression и LASSO

В этот раз механизм «feature selection» у LASSO сработал относительно лучше – обнулены все коэффициенты старше четвертой степени, что положительно отражается на возможности

интерпретации полученных результатов. В принципе, основное преимущество LASSO относительно Ridge Regression в его способности обращать коэффициенты в ноль, добавляя ясности интерпретации.

3.2 Второй полином

Второй пример, который мы рассмотрим – это полином второй степени с 50 наблюдениями и «маленькими» коэффициентами (меньше 0.5). Оценивать будем пытаться модель заведомо переспецифицированную – с 15 регрессорами. Получим оценки МНК:

```
set.seed(1897)  #установим зерно генератора для воспроизводимости
n.obs <- 50
x <- runif(n.obs, 0, 10)  #с помощью непрерывного равномерного распределения
# генерируем 50 наблюдений от 0 до 10
y <- 5 + 0.42 * x + 0.31 * x^2 + rnorm(n.obs)  #генерируем наблюдения объясняемой
# переменной как полином второй степени с случайной ошибкой N(0,1)
model.ls <- lm(y ~ poly(x, 15, raw = TRUE))
# получаем оценки МНК
```

Полученные оценки в таблице 3. Отсутствуют коэффициенты при X^{13} и X^{15}, так МНК не смог их получить из-за высокой степени мультиколлинеарности и, как следствия, сингулярности матрицы X^TX. Три последних коэффициента близки к нулю, но не равны ему (проблема в точности отображения). Мы опять видим проблему больших коэффициентов с противоположными знаками, а кроме того видна мультиколлинеарность, так как p-значение для F-статистики равно 0, но все регрессоры незначимы. Более того, можно заметить, что у коэффициентов высокие стандартные отклонения.

	Оценка	$\hat{\sigma}_{\hat{\beta}}$	t-статистика	P-значение
Константа	11.38	4.84	2.35	0.02
X	-32.34	43.26	-0.75	0.46
X^2	62.69	142.60	0.44	0.66
X^3	-58.76	233.33	-0.25	0.80
X^4	29.16	220.34	0.13	0.90
X^5	-5.90	131.05	-0.05	0.96
X^6	-1.33	51.74	-0.03	0.98
X^7	1.22	13.97	0.09	0.93
X^8	-0.37	2.61	-0.14	0.89
X^9	0.06	0.33	0.19	0.85
X^{10}	-0.01	0.03	-0.24	0.81
X^{11}	0.00	0.00	0.28	0.78
X^{12}	-0.00	0.00	-0.32	0.75
X^{14}	0.00	0.00	0.40	0.69

Таблица 3: Оценки МНК

Получим оценки Ridge Regression:

```
model.rr <- lm.ridge(y ~ poly(x, 15, raw = TRUE), lambda = seq(0, 10, 0.01))
select(model.rr)
```

```
## modified HKB estimator is 2.471e-19
## modified L-W estimator is 0.1015
## smallest value of GCV  at 0.06


model.rr.fin <- lm.ridge(y ~ poly(x, 15, raw = TRUE), lambda = 0.0592)
```

Полученные оценки в таблице 4. В данном случае, RR многие коэффициенты обратил практически в ноль, что ожидаемо, так как и сами истинные значения коэффициентов малы.

Перейдем к LASSO:

```
X <- cbind(poly(x, 15, raw = TRUE))
colnames(X) <- c("X", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10",
    "X11", "X12", "X13", "X14", "X15")
cv.las <- cv.glmnet(X, y, alpha = 1)
las.lambda <- cv.las$lambda.min
model.las <- glmnet(X, y, lambda = las.lambda)
las.est <- predict(model.las, X, type = "coef")
```

Полученный результат не может не радовать: LASSO выбрал спецификацию, которую мы задумывали, так как коэффициент при X^{15} настолько мал, что не оказывает никакого влияния. Таким образом, мы получили благодаря LASSO в данном случае удобную для интерпретации модель, не размышляя априорно о спецификации (касательно степени полинома, число 15 выбрано случайно, единственный критерий, который я использовал – достаточно большое число регрессоров) ни секунды.

Коэффициент	Ridge Regression	LASSO	Коэффициент	Ridge Regression	LASSO
Константа	5.536e+00	5.4794	X^8	-6.809e-08	0
X	4.381e-01	0.3162	X^9	-6.328e-09	0
X^2	1.882e-01	0.3061	X^{10}	-4.835e-10	0
X^3	1.449e-02	0	X^{11}	-2.610e-11	0
X^4	7.452e-04	0	X^{12}	1.229e-13	0
X^5	1.131e-05	0	X^{13}	3.197e-13	0
X^6	-3.380e-06	0	X^{14}	6.487e-14	0
X^7	-5.944e-07	0	X^{15}	9.900e-15	1.1624×10^{-15}

Таблица 4: Оценки Ridge Regression и LASSO

3.3 Третий полином

Последний полином, который мы рассмотрим, как и прошлый имеет 50 наблюдений и вторую степень, но на этот раз у него будут «большие» коэффициенты. Точно также будем оценивать сильно переспецифицированной моделью пятнадцатой степени. Получим оценки МНК:

```
set.seed(897)  #установим зерно генератора для воспроизводимости
n.obs <- 50
x <- runif(n.obs, 0, 10)  #с помощью непрерывного равномерного распределения
# генерируем 50 наблюдений от 0 до 10
```

```
y <- 15 + 10 * x + 7 * x^2 + rnorm(n.obs)  #генерируем наблюдения объясняемой
# переменной как полином второй степени с случайной ошибкой N(0,1)
model.ls <- lm(y ~ poly(x, 15, raw = TRUE))
# получаем оценки МНК
```

Полученные оценки в таблице 5. Как и в предыдущих двух примерах, мы вновь сталкиваемся с огромными коэффициентами противоположных знаков, более того, можно видеть, что они крупнее, чем в предыдущем случае (таблица 3). Оценки для X^{13} и X^{15} отсутствуют из-за необратимости матрицы X^TX, а три последних коэффициента близки к нулю, но не равны ему (вновь точность отображения виной тому). Наконец, видна мультиколлинеарность, так как точное p-значение для F-статистики равно 0, но все регрессоры незначимы по отдельности.

	Оценка	$\hat{\sigma}_{\hat{\beta}}$	t-статистика	P-значение
Константа	11.97	4.69	2.55	0.02
X	36.58	43.41	0.84	0.41
X^2	-85.15	141.60	-0.60	0.55
X^3	174.23	231.21	0.75	0.46
X^4	-190.96	219.61	-0.87	0.39
X^5	129.00	131.97	0.98	0.33
X^6	-56.49	52.73	-1.07	0.29
X^7	16.56	14.40	1.15	0.26
X^8	-3.30	2.71	-1.22	0.23
X^9	0.45	0.35	1.27	0.21
X^{10}	-0.04	0.03	-1.32	0.20
X^{11}	0.00	0.00	1.36	0.18
X^{12}	-0.00	0.00	-1.39	0.17
X^{14}	0.00	0.00	1.44	0.16

Таблица 5: Оценки МНК

Получим оценки Ridge Regression:

```
model.rr <- lm.ridge(y ~ poly(x, 15, raw = TRUE), lambda = seq(0, 1, 0.00001))
select(model.rr)

## modified HKB estimator is 2.457e-18
## modified L-W estimator is 0.0001841
## smallest value of GCV  at 0.00018


model.rr.fin <- lm.ridge(y ~ poly(x, 15, raw = TRUE), lambda = 0.00018)
```

Полученные оценки в таблице 6. RR обратил большую часть коэффициентов практически в ноль, как и в прошлый раз, решив проблему огромных коэффициентов противоположных знаков. Основное отличие от второго полинома заключается в величине λ, которая в этот раз оказалась сильно меньше, что можно объяснить тем, что «большие» коэффициенты требуют меньшего штрафа на размер коэффициентов.

Получим оценки LASSO:

```
X <- cbind(poly(x, 15, raw = TRUE))
colnames(X) <- c("X", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10",
    "X11", "X12", "X13", "X14", "X15")
cv.las <- cv.glmnet(X, y, alpha = 1)
las.lambda <- cv.las$lambda.min
model.las <- glmnet(X, y, lambda = las.lambda)
las.est <- predict(model.las, X, type = "coef")
```

С выбором спецификации LASSO вновь справился отлично. Таким образом, и в случае «больших» коэффициентов мы получили удобную для интерпретации модель, хоть LASSO и перенес часть влияния коэффициентов в свободный член.

Коэффициент	Ridge Regression	LASSO	Коэффициент	Ridge Regression	LASSO
Константа	1.472e+01	24.4601	X^8	6.764e-07	0
X	1.127e+01	9.1432	X^9	4.278e-08	0
X^2	6.163e+00	6.8479	X^{10}	9.828e-11	0
X^3	1.824e-01	0	X^{11}	-3.672e-10	0
X^4	-8.813e-03	0	X^{12}	-5.538e-11	0
X^5	-9.588e-04	0	X^{13}	-4.638e-12	0
X^6	-1.186e-05	0	X^{14}	-6.155e-14	0
X^7	5.290e-06	0	X^{15}	6.546e-14	0

Таблица 6: Оценки Ridge Regression и LASSO

3.4 Реальные данные

Рассмотрим пример из финансовой сферы, а конкретно модель CAPM:

$$R_i - R_f = \alpha + \beta(R_m - R_f) \tag{16}$$

где R_i – ставка доходности на актив, R_f – безрисковая ставка доходности, R_m – рыночная доходность, но мы рассмотрим более широкую версию. Возьмем три показателя рыночной доходности (индексы ММВБ [$MICEX$][2], РТС [$RTSI$] и РТС-финансы [$RTSfn$]), а в роли «безрискового» актива возьмем ОФЗ под номером 26806 [OFZ]. Рассмотрим в роли объясняющих переменных акции компаний из разных отраслей: Газпром (сырье)[$GAZP$], МТС (телекоммуникации)[$MTSS$] и Сбербанк (финансовый сектор)[$SBER$]. Исследовать будем на дневных данных[3] с 14.09.2012 по 05.06.2013.

Начнем с Газпрома. Модель данных выглядит следующим образом:

$$(GAZP-OFZ)_i = \alpha+\beta_1(RTSI-OFZ)_i+\beta_2(MICEX-OFZ)_i+\beta_3(RTSfn-OFZ)_i+\varepsilon_i \tag{17}$$

Получим оценки МНК:

```
dataset <- read.csv("~/cp_3kb/shares.csv")  #загружаем данные
gazp.ls <- lm(I(GAZP - OFZ) ~ I(RTSI - OFZ) + I(MICEX - OFZ) + I(RTSfn - OFZ),
    data = dataset)
```

[2] в квадратных скобках указаны имена соответствующих переменных

[3] данные взяты с сайта http://finam.ru

Полученные оценки в таблице 7. В данной регрессии коэффициент детерминации R^2 равен 0.642, а гипотеза о незначимости регрессии в целом отвергается с F-статистикой равной 100.8 с точным p-значением 0. Проверим на наличие мультиколлинеарности с помощью VIF:

```
library(car)  #подгружаем библиотеку, дабы не считать VIF вручную
vif(gazp.ls)

##  I(RTSI - OFZ) I(MICEX - OFZ) I(RTSfn - OFZ)
##         39.341         49.970          7.669
```

Как видно из результатов, мультиколлинеарность присутствует (VIF>5 для каждого), но тем не менее все регрессоры и так значимы, то есть она нам не страшна. Но всё равно посмотрим, какие результаты дадут Ridge Regression и LASSO.

	Оценка	$\hat{\sigma}_{\hat{\beta}}$	t-статистика	P-значение
Константа	-162.62	38.90	-4.18	0.00
RTSI - OFZ	-0.44	0.05	-8.86	0.00
MICEX - OFZ	0.51	0.08	6.59	0.00
RTSfn - OFZ	0.69	0.09	7.84	0.00

Таблица 7: Оценки МНК для Газпрома

Начнем с Ridge Regression:

```
gazp.rr <- lm.ridge(I(GAZP - OFZ) ~ I(RTSI - OFZ) + I(MICEX - OFZ) + I(RTSfn -
    OFZ), data = dataset, lambda = seq(0, 1, 0.001))
select(gazp.rr)  #выберем оптимальную лямбду

## modified HKB estimator is 0.0293
## modified L-W estimator is 0.5553
## smallest value of GCV  at 0.039

gazp.rr.fin <- lm.ridge(I(GAZP - OFZ) ~ I(RTSI - OFZ) + I(MICEX - OFZ) + I(RTSfn -
    OFZ), data = dataset, lambda = 0.039)
```

Полученные оценки в сводной таблице 8. В данном случае Ridge Regression не дает какого-либо интересного результата: мы получили немного уменьшенные оценки МНК. Можно наглядно увидеть как уменьшаются коэффициенты с ростом лямбды на рисунке 4. Коэффициенты на графике такие крупные, из-за того, что там они стандартизированы.

Теперь рассмотрим LASSO:

```
attach(dataset)  #выгружаем переменные из набора данных в пространство имен
X <- cbind(RTSI - OFZ, MICEX - OFZ, RTSfn)  #готовим данные для glmnet
colnames(X) <- c("RTSI-ОФЗ", "MICEX-ОФЗ", "RTSfn-ОФЗ")

gazp <- GAZP - OFZ  #создаем объясняему переменную для Газпрома

# за одно и для МТС со Сбербанком
mtss <- MTSS - OFZ
sber <- SBER - OFZ
```

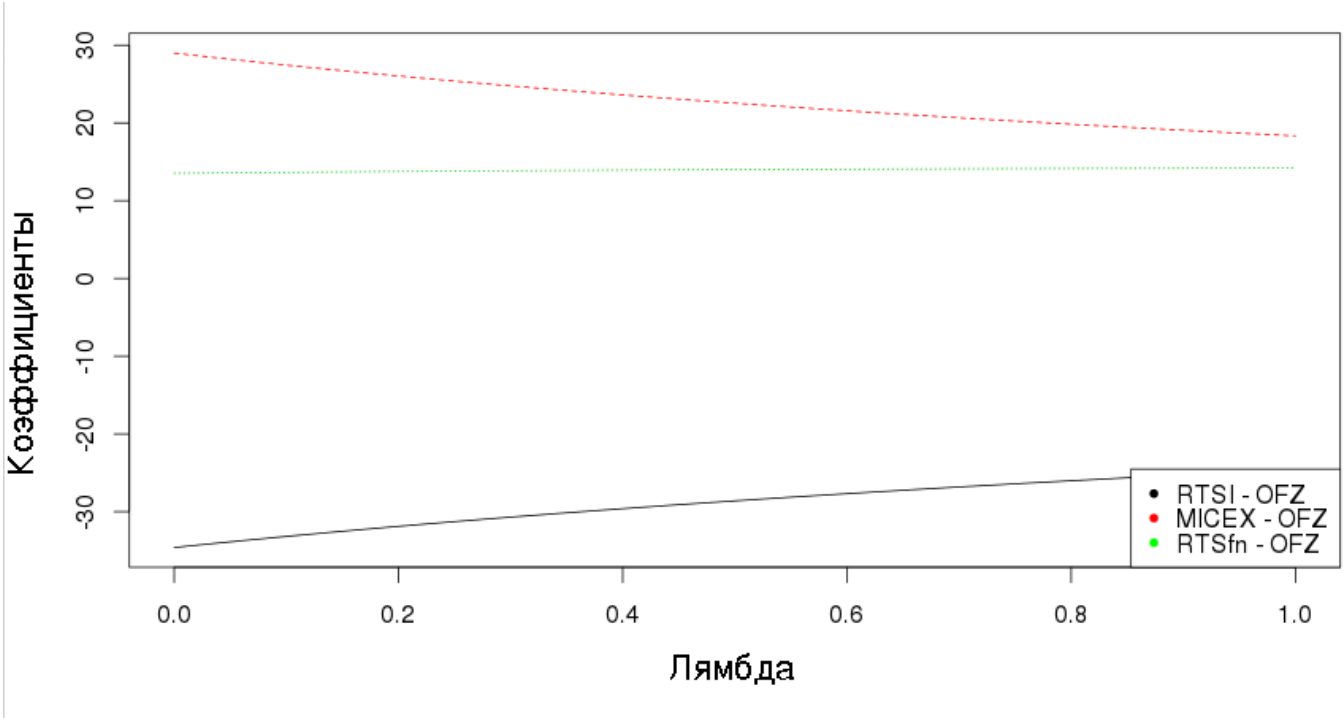

Рис. 4: Коэффициенты в зависимости от λ в Ridge Regression

```
cv.gazp <- cv.glmnet(X, gazp)
gazp.lamb <- cv.gazp$lambda.min
gazp.las <- glmnet(X, gazp, lambda = gazp.lamb)
las.est <- predict(gazp.las, X, type = "coef")

detach(dataset)
```

Полученные оценки в сводной таблице 8. Результат LASSO в отличие от RR представляет интерес, так как теперь наибольшее влияние оказывает не РТС-финансы, а ММВБ, что чисто интуитивно правдоподобно, так как сам Газпром торгуется на ММВБ. Тем не менее, это может быть просто удачным совпадением, но замечателен сам факт, что LASSO изменил «приоритет» регрессоров.

	OLS	Ridge Regression	LASSO
Константа	-162.62	-158.21	−292.6349
RTSI - OFZ	-0.44	-0.43	−0.4955
MICEX - OFZ	0.51	0.50	0.6469
RTSfn - OFZ	0.69	0.69	0.5292

Таблица 8: Оценки МНК, Ridge Regression и LASSO для Газпрома

Следующей компанией рассмотрим МТС. Получим МНК оценки:

```
mtss.ls <- lm(I(MTSS - OFZ) ~ I(RTSI - OFZ) + I(MICEX - OFZ) + I(RTSfn - OFZ),
    data = dataset)
```

Полученные оценки в таблице 9. Коэффициент детерминации R^2 равен 0.248, а гипотеза о незначимости регрессии в целом отвергается с F-статистикой 19.37 с точным p-значением 0. В целом, результат МНК хуже, чем в случае с Газпромом – индекс ММВБ незначим, как и константа (которую отбрасывать, конечно же, не станем), и R^2 сильно меньше. Мультиколлинеарность сохраняется (регрессоры те же и, следовательно, VIF'ы тоже), так что, отбросив индекс ММВБ, можно совершить ошибку второго рода. С другой стороны результаты RR и LASSO могут быть более впечатляющими.

	Оценка	$\hat{\sigma}_{\hat{\beta}}$	t-статистика	P-значение
Константа	69.29	63.55	1.09	0.28
RTSI - OFZ	0.29	0.08	3.65	0.00
MICEX - OFZ	-0.14	0.13	-1.09	0.28
RTSfn - OFZ	-0.88	0.14	-6.12	0.00

Таблица 9: Оценки МНК для МТС

Получим оценки Ridge Regression:

```
mtss.rr <- lm.ridge(I(MTSS - OFZ) ~ I(RTSI - OFZ) + I(MICEX - OFZ) + I(RTSfn -
    OFZ), data = dataset, lambda = seq(0, 1, 0.001))
select(mtss.rr)

## modified HKB estimator is 0.1931
## modified L-W estimator is 2.891
## smallest value of GCV  at 0.539


mtss.rr.fin <- lm.ridge(I(MTSS - OFZ) ~ I(RTSI - OFZ) + I(MICEX - OFZ) + I(RTSfn -
    OFZ), data = dataset, lambda = 0.539)
```

Оценки RR в сводной таблице 10. В этот раз действия RR заметны сильнее: уменьшения коснулись в основном индекса ММВБ, незначимого в МНК оценках, а вот индекс РТС-финансы не был затронут практически вообще.

Посмотрим на оценки LASSO:

```
cv.mtss <- cv.glmnet(X, mtss)
mtss.lamb <- cv.mtss$lambda.min
mtss.las <- glmnet(X, mtss, lambda = mtss.lamb)
las.est <- predict(mtss.las, X, type = "coef")
```

Результаты для LASSO в таблице 8 и они неожиданны. LASSO переложил объяснение с РТС-финансов на ММВБ и РТС (от которых зависимость отрицательная) и, при этом, увеличил константу в разы. Вызвано это может быть незначимостью константы, но такие оценки вызывают сомнения касательно их адекватности.

Наконец, перейдем к представителю финансового сектора – Сбербанку. Получим МНК-оценки:

```
sber.ls <- lm(I(SBER - OFZ) ~ I(RTSI - OFZ) + I(MICEX - OFZ) + I(RTSfn - OFZ),
    data = dataset)
```

	OLS	Ridge Regression	LASSO
Константа	69.29	52.03	227.7741
RTSI - OFZ	0.29	0.25	0.3746
MICEX - OFZ	-0.14	-0.08	−0.2999
RTSfn - OFZ	-0.88	-0.88	−0.7347

Таблица 10: Оценки МНК, Ridge Regression и LASSO для МТС

Оценки находятся в таблице 11. Коэффициент детерминации R^2 равен 0.2797, гипотеза о незначимости регрессии в целом отвергается с F-статистикой равной 22.62 и точным p-значением 0. Кроме того, на 5% уровне значимости только константа и индекс РТС являются значимыми.

	Оценка	$\hat{\sigma}_{\hat{\beta}}$	t-статистика	P-значение
Константа	-43.71	20.93	-2.09	0.04
RTSI - OFZ	0.07	0.03	2.49	0.01
MICEX - OFZ	-0.04	0.04	-0.88	0.38
RTSfn - OFZ	-0.03	0.05	-0.55	0.59

Таблица 11: Оценки МНК для Сбербанка

Посмотрим на Ridge Regression:

```
sber.rr <- lm.ridge(I(SBER - OFZ) ~ I(RTSI - OFZ) + I(MICEX - OFZ) + I(RTSfn -
    OFZ), data = dataset, lambda = seq(0, 1, 0.001))
select(sber.rr)

## modified HKB estimator is 0.5894
## modified L-W estimator is 2.476
## smallest value of GCV  at 1


sber.rr.fin <- lm.ridge(I(SBER - OFZ) ~ I(RTSI - OFZ) + I(MICEX - OFZ) + I(RTSfn -
    OFZ), data = dataset, lambda = 1.7701)
```

Полученные оценки в таблице 12. Как мы и ожидаем, RR уменьшает размер коэффициентов, но в данном случае он уменьшил как незначимый ММВБ, так и значимый на 5% РТС.

Получим оценки LASSO:

```
cv.sber <- cv.glmnet(X, sber)
sber.lamb <- cv.sber$lambda.min
sber.las <- glmnet(X, sber, lambda = sber.lamb)
las.est <- predict(sber.las, X, type = "coef")
```

Полученные нами оценки (в таблице 12) смысловой нагрузкой не обладают по отношению к МНК оценкам.

	OLS	Ridge Regression	LASSO
Константа	-43.71	-54.871	−26.4972
RTSI - OFZ	0.07	0.045	0.0634
MICEX - OFZ	-0.04	-0.007	−0.0632
RTSfn - OFZ	-0.03	-0.030	0.0679

Таблица 12: Оценки МНК, Ridge Regression и LASSO для Сбербанка

4 Упражнения

4.1 Ridge Regression

Рассматривается классическая линейная модель: $Y = X\beta + \varepsilon$ со свободным членом и числом регрессоров равным k. Известно также, что $\varepsilon \sim N(0, \sigma^2)$.

(a) Выпишите постановку Ridge Regression в форме целевой функции и ограничения

(b) Выпишите постановку Ridge Regression в форме лагранжиана

(c) Найдите оценки $\hat{\beta}^{RR}$ в явном виде

(d) Какую проблему матрицы $X^T X$ решает Ridge Regression и в каких случаях эта проблема возникает в принципе?

(e) Будет ли оценка $\hat{\beta}^{RR}$ смещена? Найдите $E(\hat{\beta}^{RR})$

(e) Как зависит параметр λ (из пункта (b)) от параметра t (из пункта (a))?

(f) Что будет с оценками, если λ будет равной нулю, а если устремится к бесконечности?

(g) Каким преимуществом обладают оценки Ridge Regression по отношению к методу наименьших квадратов?

4.2 LASSO

Рассматривается классическая линейная модель: $Y = X\beta + \varepsilon$ со свободным членом и числом регрессоров равным k. Известно также, что $\varepsilon \sim N(0, \sigma^2)$.

(a) Выпишите постановку LASSO в форме целевой функции и ограничения

(b) Выпишите постановку LASSO в форме лагранжиана

(c) Как зависит параметр λ (из пункта (b)) от параметра t (из пункта (a))?

(d) Что будет с оценками, если λ будет равной нулю, а если устремится к бесконечности?

(e) Каким свойством обладает LASSO относительно спецификации модели? Чем оно обусловлено?

4.3 RR и LASSO

Рассматривается двухфакторная линейная модель: $Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \varepsilon_i$. В пространстве (β_1, β_2) изобразите линии уровня для RSS и ограничения Ridge Regression и LASSO для случая:

(a) Оценок коэффициентов метода наименьших квадратов

(b) Один из коэффициентов для LASSO равен нулю (и покажите почему для Ridge Regression такая ситуация практически невозможна)

(c) Любых ненулевых оценок коэффициентов (при какой λ возможна такая ситуация)

Как соотносятся λ для пунктов (a), (b) и (c)? Подумайте, какие ещё формы ограничения могли бы выполнять роль «feature selection»?

4.4 Счетная задача

Рассматривается однофакторная линейная модель в центрированных переменных: $y_i = \beta x_i + \varepsilon_i$, где $y_i = Y_i - \overline{Y}$ и $x_i = X_i - \overline{X}$, а $\varepsilon \sim N(0, \sigma^2)$. Также известно что:

$$y = \begin{pmatrix} -2 \\ -1 \\ 0 \\ 1 \\ 2 \end{pmatrix} \quad x = \begin{pmatrix} -2 \\ -3 \\ 0 \\ 4 \\ 1 \end{pmatrix}$$

(a) Получите оценку коэффициента β методом наименьших квадратов

(b) Выведите оценку Ridge Regression для данной модели

(c) Пусть $\lambda = 9$. Получите оценку $\hat{\beta}^{RR}$, как она соотносится с оценкой метода наименьших квадратов?

(d) Выведите оценку LASSO для данной модели

(e) Пусть $\lambda = 6$. Получите оценку $\hat{\beta}^{LASSO}$. А что будет, если $\lambda = 26$

4.4.1 Решение

(a) Оценка МНК: $\hat{\beta} = \frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n} x_i^2} = \frac{4+3+0+4+2}{4+9+0+16+1} = \frac{13}{30} = 0.43$

(b) Постановка Ridge Regression:

$$PRSS = \sum_{i=1}^{n} (y_i - \beta x_i)^2 + \lambda \beta^2 \longrightarrow min$$

Возьмем производную по параметру β:

$$\frac{\partial PRSS}{\partial \beta} = -2 \sum_{i=1}^{n} (y_i - \beta x_i) x_i + 2\lambda\beta = 0$$

$$-\sum_{i=1}^{n} y_i x_i + \beta \sum_{i=1}^{n} x_i^2 + 2\lambda\beta = 0$$

$$\beta(\sum_{i=1}^{n} x_i^2 + \lambda) = \sum_{i=1}^{n} y_i x_i$$

Таким образом, оценка для Ridge Regression выглядит следующим образом:

$$\hat{\beta}^{RR} = \frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n} x_i^2 + \lambda}$$

(c) Подставим $\lambda = 9$ в выражение для оценки:$\hat{\beta}^{RR} = \frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n} x_i^2 + \lambda} = \frac{13}{30+9} = \frac{1}{3}$

(d) Рассмотрим два случая. Пусть $\beta \geqslant 0$, тогда постановка LASSO:

$$PRSS = \sum_{i=1}^{n} (y_i - \beta x_i)^2 + \lambda\beta \longrightarrow min$$

Возьмем первую производную по β:

$$\frac{\partial PRSS}{\partial \beta} = -2\sum_{i=1}^{n} (y_i - \beta x_i) x_i + \lambda = 0$$

$$-2\sum_{i=1}^{n} y_i x_i + 2\beta \sum_{i=1}^{n} x_i^2 + \lambda = 0$$

Тогда оценка LASSO выглядит следующим образом:

$$\hat{\beta}^{LASSO} = \frac{\sum_{i=1}^{n} y_i x_i - \lambda/2}{\sum_{i=1}^{n} x_i^2}, \beta \geqslant 0$$

В случае $\beta < 0$ просто меняется знак перед λ. Оценка выглядит следующим образом:

$$\hat{\beta}^{LASSO} = \frac{\sum_{i=1}^{n} y_i x_i + \lambda/2}{\sum_{i=1}^{n} x_i^2}, \beta < 0$$

(d) Мы рассматриваем случай $\beta \geqslant 0$, так как числитель $\sum_{i=1}^{n} y_i x_i = 13 > 0$, а суть метода в приближение оценки коэффициента к нулю. Подставляем $\lambda = 6$ в выражение оценки LASSO: $\hat{\beta}^{LASSO} = \frac{\sum_{i=1}^{n} y_i x_i + \lambda/2}{\sum_{i=1}^{n} x_i^2} = \frac{13-6/2}{30} = \frac{10}{30} = \frac{1}{3}$. Мы получили такую же оценку как и в случае Ridge Regression, но при разных λ.

Подставим $\lambda = 26$ в выражение оценки LASSO: $\hat{\beta}^{LASSO} = \frac{\sum_{i=1}^{n} y_i x_i + \lambda/2}{\sum_{i=1}^{n} x_i^2} = \frac{13-26/2}{30} = \frac{0}{30} = 0$. Таким образом, при данном значении λ LASSO обратил коэффициент в ноль.

Список литературы

[1] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, February 1970.

[2] Robert Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, page 267–288, 1996.

[3] Ryan Tibshirani. Modern regression 2: The lasso, 2013.

[4] Trevor Hastie, Robert Tibshirani, and J. Jerome H. Friedman. *The elements of statistical learning.* Springer New York, 2001.

[5] Ali R. Syed. *A review of cross validation and adaptive model selection.* PhD thesis, Georgia State University, 2011.