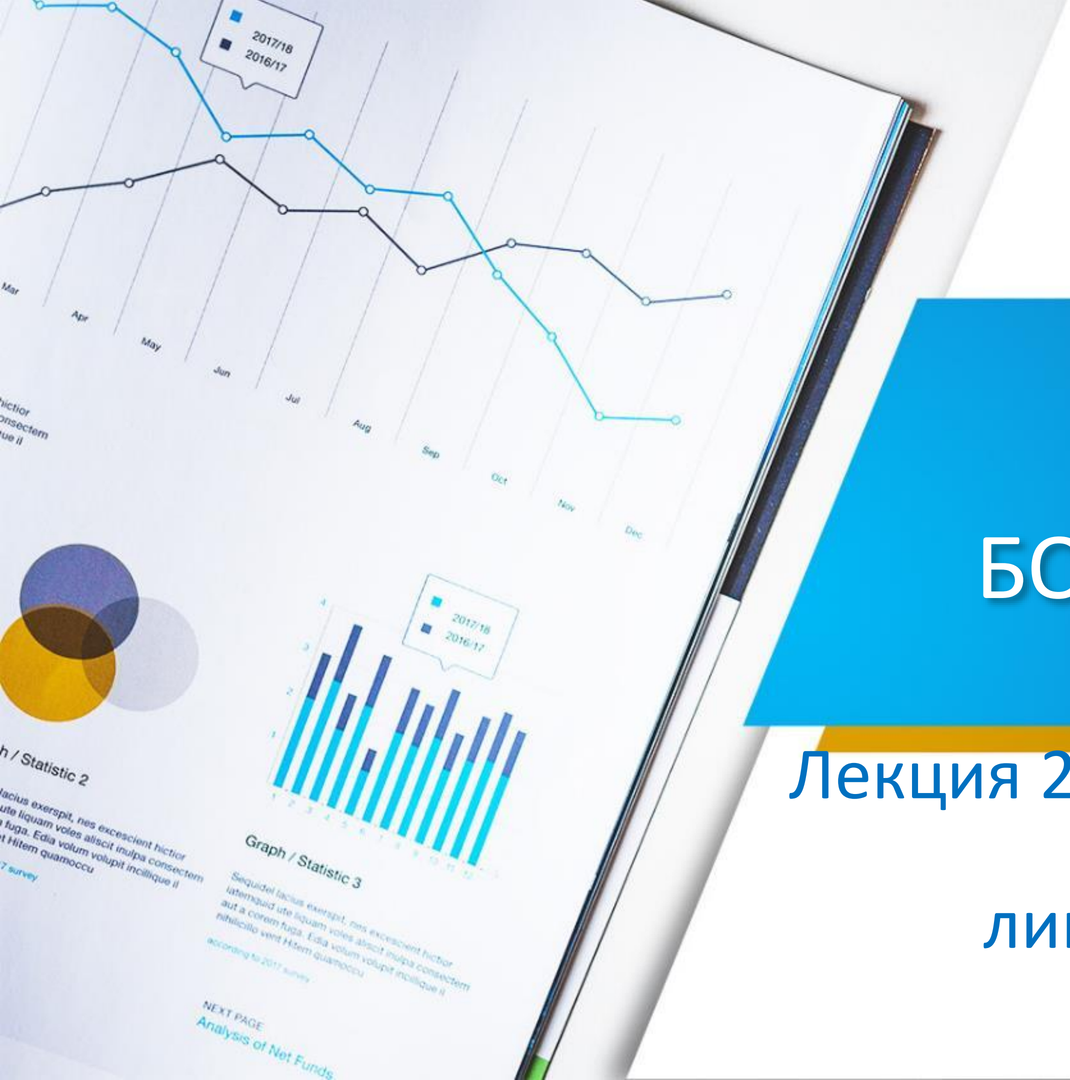


СТАТИСТИКА БОЛЬШИХ ДАННЫХ

Лекция 21. Методы регуляризации в задаче множественной линейной регрессии. Часть 1

Введение

В этой лекции рассматриваются отличные от метода наименьших квадратов подходы к решению задач множественной линейной регрессии и отбору значимых факторов при больших размерностях модели.

- Мотивация и постановка задачи регуляризации
- Методы регуляризации: гребневая и лассо регрессии
- Примеры
- Заключение

Мотивация и постановка задачи регуляризации

Стандартная линейная модель: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + e$

В этой лекции мы обсудим модификации обычного критерия наименьших квадратов и методов отбора переменных в целях достижения лучшей точности прогноза, отбора значимых факторов и интерпретируемости модели.

- Если число наблюдений n много больше, чем число переменных p ($n>p$), то МНК-оценки имеют малые дисперсии.
- Если $n \approx p$, то тогда растет дисперсия прогноза и уменьшается его точность.
- Если $p>n$, тогда нет единственности МНК-оценок коэффициентов регрессии и МНК неприменим.

Мотивация и постановка задачи регуляризации

Задача, решение которой не единственно, относится к **некорректным задачам**. Для решения некорректных задач применяются методы регуляризации. Один из таких методов заключается во введении дополнительных слагаемых в целевую функцию МНК.

Отбор значимых факторов. Увеличение числа *p* переменных модели обычно связано с включением в модель незначимых или зависимых факторов. Методы отбора факторов (прямой, обратный и смешанный отбор) здесь в силу эффекта больших данных (p>n) не работают. Желательно предложить метод, который позволил бы в этой ситуации исключать незначимые и лишние факторы.

Мотивация и постановка задачи регуляризации

Интерпретируемость модели. Часто наблюдается, что некоторые переменные множественной линейной регрессии в действительности не связаны с откликом. Удалить эти переменные можно положив значения соответствующих коэффициентов регрессии равными нулю.

Существует несколько альтернатив использованию МНК:

- отбор подмножества факторов (Subset Selection)
- сокращение размерности (Dimension Reduction)
- регуляризация (Shrinkage) - гребневая регрессия и метод лассо.

Ниже мы остановимся на методах регуляризации процедуры оценивания по МНК в случае большой размерности модели (случай $p>n$).

Гребневая регрессия

МНК-процедура подгонки оценок β_0, $\beta_1, \dots, \beta_p$ минимизирует выражение

$$RSS = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2.$$

Гребневая регрессия получена из *МНК* введением функции штрафа $\lambda \sum_{j=1}^{p} \beta_j^2$ на величину коэффициентов $\beta_1, \dots, \beta_p$. В результате решается задача нахождения коэффициентов гребневой (ridge) регрессии $\beta_j^{(r)}$ минимизирующих

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2 = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2,$$

где $\lambda > 0$ — штрафной коэффициент.

Гребневая регрессия

Метод представляет собой компромисс между двумя различными критериями. Как и в случае *МНК*, гребневая регрессия нацелена на подгонку к данным, уменьшая величину *RSS*. Однако, стремление к уменьшению штрафной функции обеспечивает сжатие оценок β_j к нулю.

Коэффициент штрафа λ используется для баланса влияния на оценки обоих слагаемых в целевой функции.

- Если λ=0, то гребневая регрессия совпадает с МНК-регрессией.
- При росте λ влияние штрафа растет, а оценки коэффициентов β_j стремятся к нулю.

Заметим, что штраф не накладывается на β_0, который просто измеряет среднее значение отклика при нулевых значениях факторов.

Особенности поведения гребневой регрессии

Когда число факторов *p* приближается к числу наблюдений *n*, то дисперсия МНК-оценок неограниченно возрастает. Далее, если *p>n,* то МНК-оценки будут не единственны, т.е., метод наименьших квадратов в этом случае неприменим.

В рассмотренных случаях ($p \leq n$ и $p > n$) оценки гребневой регрессии в целом будут вполне приемлемыми, уступая в точности оценкам МНК в случае малых *p* и являясь компромиссом между значениями дисперсии оценок и их смещения в случае больших *p* – метод гребневой регрессии универсален.

Метод лассо

Гребневая регрессия имеет один очевидный недостаток — она всегда включает в себя все p факторов. Штраф $\lambda\sum_j \beta_j^2$ сожмет все коэффициенты но не сделает их равными нулю. Увеличение параметра λ безусловно уменьшит рассеяние оценок коэффициентов регрессии, но никогда не приведет к исключению из модели какой-либо переменной.

Метод лассо *(LASSO -- Least Absolute Shrinkage and Selection Operator)*(Tibshirani, 1996) представляет собой альтернативу гребневой регрессии, которая не имеет указанного недостатка. Коэффициенты лассо $\beta_j^{(L)}$ минимизируют следующую функцию цели с негладким штрафом $\lambda\sum_{j=1}^{p}|\beta_j|$:

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = RSS+\lambda\sum_{j=1}^{p}|\beta_j|\,.$$

Метод лассо

Метод лассо и гребневая регрессия имеют близкие формализации. Единственное различие состоит в использовании в качестве штрафных функций различных норм, l_2-норму в гребневой регрессии и l_1-норму в лассо.

Как и в случае гребневой регрессии, лассо сжимает оценки к нулю. Однако, в случае лассо l_1-штраф имеет эффект обращения в нуль некоторых коэффициентов регрессии при достаточно большом значении параметра настройки λ. Следовательно, лассо обладает свойством отбора переменных (variable selection).

Модели, сгенерированные на основе лассо, значительно легче интерпретировать, чем модели гребневой регрессии. Можно сказать, что лассо порождает *разреженные модели*, т.е., модели, работающие с некоторым подмножеством множества переменных – подмножество отобранных значимых факторов.

Метод лассо

При $\lambda = 0$ лассо дает подгонку по *МНК.*
Когда λ становится достаточно большим, лассо дает нулевую модель, в которой все оценки коэффициентов равны нулю.

Между этими двумя крайними случаями гребневая регрессия и лассо сильно отличаются друг от друга:

- В зависимости от величины λ лассо может генерировать модель с любым числом значимых факторов.
- Напротив, гребневая регрессия включает в себя все факторы, хотя величины оценок коэффициентов при них зависят от λ.

Пример 1

Зададим число и вектор в 10-мерном пространстве: $\beta_0 = 3;$ $\beta^* = (1{,}0, \dots 0, -1).$

Зададим 5 векторов значений факторов размерности $p = 10$:

$$x_k = (x_{k1}, x_{k2}, \dots, x_{kp}),\ \ k = 1, \dots, n;\ \ n = 5.$$

Вычислим 5 значений отклика: $y_k = \beta_0 + (\beta^*, x_k) + e_k,\ \ k = 1, \dots, n.$

e_k – случайная ошибка, распределенная по нормальному закону с нулевым средним.

Задача: оценить вектор β^* и свободный член β_0 по значениям отклика y_k и значениям факторов - векторам x_k.

Это задача множественной линейной регресии со значениями n=5 и p=10.

Применим метод гребневой регрессии и метод лассо и сравним их результаты.

Решение задачи оформим в пакете R.

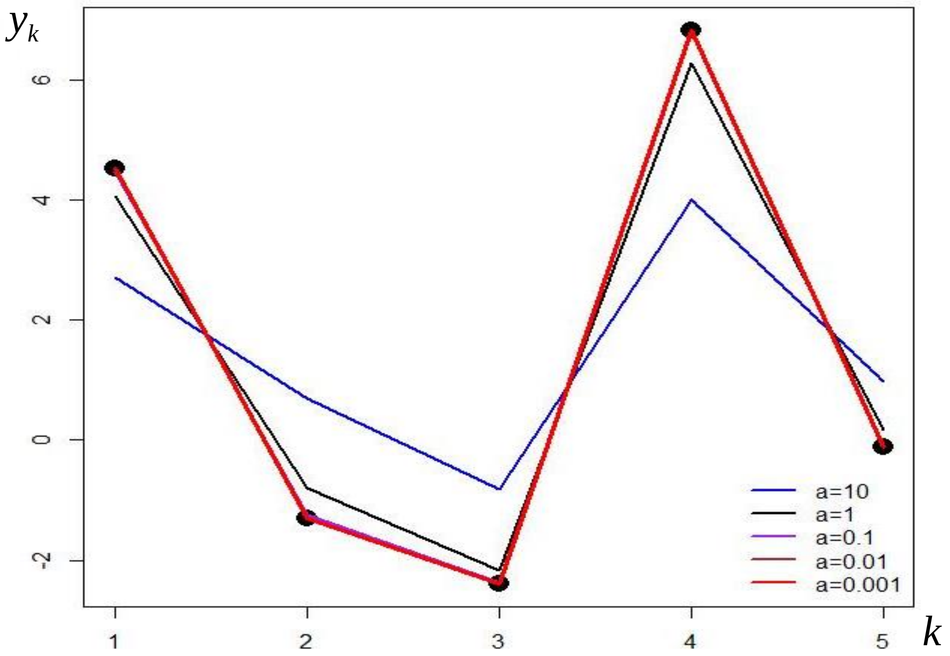
Пример 1

```
library(glmnet)
p<-10
n<-5
b<-c(1,0,0,0,0,0,0,0,0,-1)
b0<-3
X<-matrix(nrow=n,ncol=p)
for (j in 1:p) { for (i in 1:n){X[i,j]<- rnorm(1,0,3)}}
X
           [,1]       [,2]       [,3]      [,4]       [,5]       [,6]       [,7]        [,8]      [,9]      [,10]
[1,]  0.2422478  0.1384479 -1.3618779 -2.637685  0.3566118 -1.0923557  0.7041286  1.35862104 -1.752460 -1.2596165
[2,]  0.1466275  2.4573147 -0.4119873  1.460291 -3.1334400 -0.9963586 -2.1645335 -0.38991209  4.388400  4.4265086
[3,] -1.2353552 -2.6302537 -3.0084073  2.027529  4.0841774  3.0425508  0.8452573 -5.12048739 -1.269423  4.3189563
[4,]  1.8505990 -1.7720798 -0.8919914 -1.873983  0.1038625  0.8889142 -0.1988888  3.13081581  7.217463 -2.0865766
[5,] -4.0534826 -5.9159780  1.2720261 -2.523003  0.3691043 -5.1880821  5.4514842 -0.02112122 -2.736105 -0.8520606
y<-vector(length=p)
for (i in 1:n) {y[i]<- b0+b%*%X[i,]+rnorm(1,0,0.1)}
y
[1]  4.5284846 -1.3036088 -2.3982667  6.8408026 -0.1077062
a<-c(10,1,0.1,0.01,0.001)
```

Пример 1

```
20) ridge.mod=glmnet(X,y,alpha=0,lambda=a)
21) ridge.coef=predict(ridge.mod,type="coefficients",s=a)[0:p+1,]
22) ridge.coef
23) 11 x 5 sparse Matrix of class "dgCMatrix"
24)                       1            2           3            4            5
25) (Intercept)  1.652831326  1.685281800  1.67026925  1.670502138  1.670531948
26) V1           0.213407384  0.439418009  0.50301726  0.519477260  0.521188481
27) V2           0.007866803  0.003995102  0.01130088  0.007904743  0.007575406
28) V3          -0.031222534 -0.205675963 -0.28294845 -0.285307916 -0.285562318
29) V4          -0.233254404 -0.480167081 -0.54852311 -0.562772609 -0.564244067
30) V5           0.005367610  0.105569445  0.14346285  0.146823085  0.147173808
31) V6           0.032769200  0.114476865  0.13193654  0.134417817  0.134651713
32) V7          -0.025127596 -0.039494606 -0.03475861 -0.037074943 -0.037311452
33) V8           0.190244607  0.349816483  0.38957493  0.390773062  0.390872074
34) V9           0.062744611  0.088914988  0.08326954  0.080159643  0.079827322
35) V10         -0.200075628 -0.413808885 -0.46082066 -0.462059095 -0.462171648
```

Пример 1

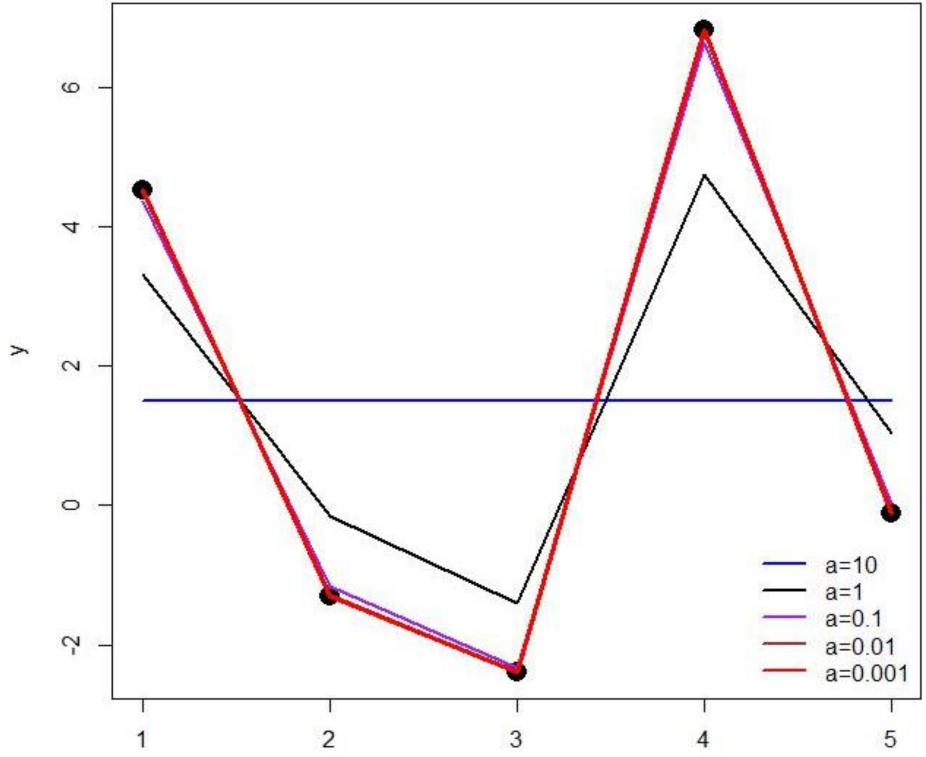


λ	**RSS**
10	19.02
1	0.8977
0.1	0.01157
0.01	0.0001224
0.001	0.000001916

Пример 1

```
lasso.mod=glmnet(X,y,alpha=1,lambda=a)
lasso.coef=predict(lasso.mod,type="coefficients",s=a)[0:p+1,]
lasso.coef
11 x 5 sparse Matrix of class "dgCMatrix"
                   1          2          3           4            5
(Intercept) 1.511941  2.3122056  2.9448813  2.96581760 2.9594408533
V1                 .  0.4285897  0.9219040  0.95155657 0.9573593412
V2                 .          .          .           .            .
V3                 .          .          . -0.03464204 -0.0425538183
V4                 .          .          .           . -0.0070229625
V5                 .          .          .           .  0.0000872198
V6                 .          .          .           .            .
V7                 .          .          .           .            .
V8                 .  0.1525901          .           .            .
V9                 .          .          .           . -0.0020012179
V10                . -0.5575692 -0.9573958 -0.99406932 -0.9937642554
```

Пример 1



λ	**RSS**
10	63.34
1	9.474
0.1	0.1041
0.01	0.00206
0.001	0.00004207