Линейная регрессия

1. Линейная регрессия

Задача восстановления регрессии заключается в обучении модели, предсказывающей значения вещественной целевой переменной y по значениям входных переменных $x_i, i = 1,...,d$. Простейшей моделью зависимости является линейная: $y = f(x;b) = \sum_{j=1}^{d} b_j h_j(x)$, где b_j - параметры модели, которые требуется подобрать на этапе обучения, $h_j(x)$ - заданные функции векторного аргумента x.

Для подбора коэффициентов b_j может применяться метод наименьших квадратов, который в достаточно общей форме можно представить в следующем виде:

$$\hat{b} = \arg\min_b \sum_{i=1}^{n} w_i (y_i - \sum_{j=1}^{d} b_j h_j(x) + c_i)^2 = \arg\min_b \|W(y - Xb + c)\|_2^2$$

где y - вектор-столбец, содержащий значения целевой переменной прецедентов обучающей выборки, X - матрица предикативных переменных $x_{i,j} = h_j(x_i)$, W - диагональная матрица корней из весов прецедентов $W = diag\left(\sqrt{w_1},...,\sqrt{w_n}\right)$, c - вектор-столбец смещений.

Для решения данной оптимизационной задачи предназначена функция, работа которой основана на QR-разложении матрицы *X*.

Использование:

lm(formula, data, subset, weights, na.action, method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)

Аргументы:

formula – формула

data – фрейм данных или список, содержащий переменные, использованные в символическом описании модели formula. Если data = NULL имена, использованные в formula, должны быть доступны в текущем рабочем пространстве

subset – вектор, определяющий подвыборку, которую следует использовать для обучения;

weights – вектор весов прецедентов;

na.action – функция для обработки пропущенных значений в выборке;

method – строковое описание метода решения задачи. method = "qr" определяет использование QR-разложения для отыскания коэффициентов *b*. method = "model.frame" обозначает, что будет лишь сформирован фрейм данных, содержащий фигурирующие в модели переменные, поиск коэффициентов b выполнен не будет;

model – логическое значение, которое определяет будет ли в качестве элемента списка, описывающего обученную модель, возвращен фрейм данных, содержащий фигурирующие в модели переменные;

x – логическое значение, которое определяет будет ли в качестве элемента списка, описывающего обученную модель, возвращена матрица X;

y – логическое значение, которое определяет будет ли в качестве элемента списка, описывающего обученную модель, возвращен вектор y;

qr – логическое значение, которое определяет будет ли в качестве элемента списка, описывающего обученную модель, возвращен список с информацией о выполненном QR-разложении;

singular.ok – логическое значение, определяющее следует ли выдавать ошибку в случае неполноты столбцового ранга матрицы X;

contrasts – список, определяющий интерпретацию номинальных признаков, заданных факторами. Так как метод наименьших квадратов не может естественным образом обрабатывать номинальные переменные, их предварительно необходимо перевести в количественные. Все допустимые способы данного преобразования приводят к обучению одинаковых моделей (выдающих одинаковые предсказания для одних и тех же входов), однако данный параметр может быть полезен для более удобной интерпретации модели;

offset – вектор смещений c.

Выходные значения:

coefficients – полученные коэффициенты $\hat{b}$;

fitted.values – предсказания полученной модели на обучающей выборке;

residuals – остатки $y_i - f(x_i;\hat{b})$;

df.residual – количество степеней свободы;

rank – ранг матрицы X;

call – строка вызова функции lm;

weights – веса прецедентов;

offset – смещения;

na.action – информация об обработанных пропущенных значениях;

contrasts – отображения номинальных переменных;

model – фрейм данных, содержащий фигурирующие в модели переменные;

x – матрица X;

y – вектор y;

xlevels – уровни факторов, содержащих значения номинальных переменных.

Для анализа построенной модели может быть использована функция summary(object, correlation = FALSE, symbolic.cor = FALSE, ...), где object - список, возвращенный функцией lm с параметром qr = TRUE. Параметр correlation определяет, будет ли вычислена матрица корреляции коэффициентов модели, symbolic.cor - следует ли выводить на экран матрицу корреляции в числовом или символьном виде. Результатом работы функции summary является список со следующими элементами:

residuals - взвешенные остатки $\sqrt{w_i}\left(y_i - f(x_i;\hat{b})\right)$;

coefficients - матрица размеров $d^* \times 4$, столбцы которой содержат оценные коэффициенты $\hat{b}$, их стандартные ошибки, значения t-статистики и p-value. Количество строк матрицы d^* соответствует количеству линейно независимых переменных в модели;

aliased – логический вектор, задающий множество линейно независимых переменных модели, определяемое значениями FALSE;

sigma – остаточная стандартная ошибка

$$\sigma = \sqrt{\frac{1}{n-d^*}\sum_{i=1}^{n} w_i \left(y_i - f(x_i;\hat{b})\right)^2} \;;$$

df – вектор, равный $(d^*, n-d^*, d)$, содержащий показатели количества степеней свободы;

fstatistic – вектор, содержащий значение F-статистики для проверки значимости модели и показатели ее степеней свободны;

r.squared – коэффициент детерминации;

adj.r.squared – подправленный коэффициент детерминации;

cov.unscaled – оценка матрицы ковариации коэффициентов;

correlation – оценка матрицы кореляции коэффициентов;

symbolic.cor – логическое значение, определяющее следует ли выводить оценку матрицы корреляции коэффициентов в символьном виде.

Пример 1.

В качестве примера рассмотрим задачу восстановления зависимости уровня озона в воздухе от уровня солнечной радиации, скорости ветра и температуры, воспользовавшись набором данных airquality из пакета datasets, который содержит измеренные значения соответствующих показателей в Нью-Йорке в 1973 году.

Обучающее множество: airquality

Признаки (независимые переменные):

1. уровень солнечной радиации
2. скорость ветра
3. температура

Зависимая переменная:

-- уровень озона

```
library(datasets)
air = airquality[, c("Ozone", "Solar.R", "Wind", "Temp")]
air
plot(Ozone ~ Solar.R, data=air)
plot(Ozone ~ Wind, data=air)
plot(Ozone ~ Temp, data=air)
```

```
f = lm(Ozone ~ ., data = air, subset = !is.na(Solar.R) & !is.na(Ozone))
f
summary(f)
```

Дадим интерпретацию полученным результатам. Для исследуемой зависимости была получена модель

$$Ozone = b_0 + b_1 \cdot Solar.R + b_2 \cdot Wind + b_3 \cdot Temp =$$
$$= -64.34208 + 0.05982 \cdot Solar.R - 3.33359 \cdot Wind + 1.65209 \cdot Temp$$

Стандартные ошибки коэффициентов равны 23.05472, 0.02319, 0.65441 и 0.25353 соответственно. Для выполнения статистических тестов с нулевыми гипотезами о том, что $b_i = 0$, были подсчитаны значения t-статистик и p-value. На основе полученных результатов, можно сделать вывод, что при уровне значимости α=0.01 коэффициент b_1 следует признать незначимым для модели, а все остальные коэффициенты — значимыми. Для проведения статистического теста о значимости всей модели в целом (нулевая гипотеза $b_i = 0,\ i = 1,...,d^*$), было вычислено значение F-статистики, равное 54.83. Исходя из того, что в предположении об истинности нулевой гипотезы данная величина должна иметь распределение Фишера со степенями свободы $d^* - 1 = 3$ и $n - d^* - 1 = 111 - 3 - 1 = 107$, было вычислено p-value < 2×10^{-16}. Следовательно, в данном случае нулевую гипотезу следует отвергнуть и считать построенную модель статистически значимой.

2. Гребневая регрессия

Одним из наиболее популярных методов регуляризации является гребневая регрессия, заключающаяся в решении следующей оптимизационной задачи:

$$\hat{b}^{ridge} = \arg\min_b \sum_{i=1}^{n} (y_i - \sum_{j=1}^{d} b_j h_j(x))^2 + \lambda \sum_{j=1}^{d} b_j^2 = \arg\min_b \|y - Xb\|_2^2 + \|b\|_2^2.$$

Для решения данной задачи предназначена функция **lm.ridge**

Использование:

```
lm.ridge(formula, data, subset, na.action, lambda = 0, model = FALSE, x = FALSE, y = FALSE, contrasts = NULL, ...)
```

Аргументы:

Большинство параметров совпадает с параметрами функции **lm**.

Параметр **lambda** представляет собой вектор, содержащий величины λ, для которых требуется решить оптимизационную задачу, приведенную выше. Если в описание модели **formula** включен свободный член, то соответствующий коэффициент не будет учитываться в штрафной компоненте целевой функции.

Выходное значение:

coef – матрица коэффициентов $\hat{b}^{ridge}$ для всех λ;
lambda – вектор использованных значений λ.

Для графического отображения зависимости величин коэффициентов $\hat{b}^{ridge}$ от λ можно к результату функции **lm.ridge** применить функцию **plot**.

Чтобы получить коэффициенты полученных линейных моделей можно применить функцию **coef(object, ...)** или **coefficients(object, ...)**, где **object** – объект, возвращенный функцией **lm.ridge**. Каждая строка матрицы коэффициентов соответствует определенному значению λ. Применим гребневую регрессию к рассмотренной ранее задаче предсказания уровня озона в воздухе.

Пример 2.

Обучающее множество: airquality

```
library(MASS)
library(datasets)
air = airquality[, c("Ozone", "Solar.R", "Wind", "Temp")]
f = lm.ridge(Ozone ~ ., data = air, subset = !is.na(Solar.R) & !is.na(Ozone), lambda = seq(1, 10000, by = 10))
plot(f)
```

Задание

1. Загрузите данные из файла reglab1.txt. Используя функцию lm, постройте регрессию (используйте разные модели). Выберите наиболее подходящую модель, объясните свой выбор.
2. Реализуйте следующий алгоритм для уменьшения количества признаков, используемых для построения регрессии: для каждого $k \in \{0,1,...,d\}$ выбрать подмножество признаков мощности k^1, минимизирующее остаточную сумму квадратов *RSS*. Используя полученный алгоритм, выберите оптимальное подмножество признаков для данных из файла reglab2.txt. Объясните свой выбор. Для генерации всех возможных сочетаний по m элементов из некоторого множества x можно использовать функцию combn(x, m, ...).
3. Загрузите данные из файла cygage.txt. Постройте регрессию, выражающую зависимость возраста исследуемых отложений от глубины залегания, используя веса наблюдений. Оцените качество построенной модели.
4. Загрузите данные Longley (макроэкономические данные). Данные состоят из 7 экономических переменных, наблюдаемых с 1947 по 1962 годы (*n=16*):
GNP.deflator - дефлятор цен,
GNP - валовой национальный продукт,
Unemployed – число безработных
Armed.Forces – число людей в армии
Population – население, возраст которого старше 14 лет
Year - год
Employed – количество занятых
Построить регрессию lm(Employed ~ .) .

Исключите из набора данных longley переменную "Population". Разделите данные на тестовую и обучающую выборки равных размеров случайным образом. Постройте гребневую регрессию для значений $\lambda = 10^{-3+0.2\cdot i}, i = 0,...,25$, подсчитайте ошибку на тестовой и обучающей выборке для данных значений λ, постройте графики. Объясните полученные результаты.

5. Загрузите данные EuStockMarkets из пакета « datasets». Данные содержат ежедневные котировки на момент закрытия фондовых бирж: Germany DAX (Ibis), Switzerland SMI, France CAC, и UK FTSE. Постройте на одном графике все кривые изменения котировок во времени. Постройте линейную регрессию для каждой модели в отдельности и для всех моделей вместе. Оцените, какая из бирж имеет наибольшую динамику.

6. Загрузите данные JohnsonJohnson из пакета «datasets». Данные содержат поквартальную прибыль компании Johnson & Johnson с 1960 по 1980 гг. Постройте на одном графике все кривые изменения прибыли во времени. Постройте линейную регрессию для каждого квартала в отдельности и для всех кварталов вместе. Оцените, в каком квартале компания имеет наибольшую и наименьшую динамику доходности. Сделайте прогноз по прибыли в 2016 году во всех кварталах и в среднем по году.

7. Загрузите данные sunspot.year из пакета «datasets». Данные содержат количество солнечных пятен с 1700 по 1988 гг. Постройте на графике кривую изменения числа солнечных пятен во времени. Постройте линейную регрессию для данных.

8. Загрузите данные из файла пакета «UKgas.scv». Данные содержат объемы ежеквартально потребляемого газа в Великобритании с 1960 по 1986 гг. Постройте линейную регрессию для каждого квартала в отдельности и для всех кварталов вместе. Оцените, в каком квартале потребление газа имеет наибольшую и наименьшую динамику доходности. Сделайте прогноз по потреблению газа в 2016 году во всех кварталах и в среднем по году.

9. Загрузите данные cars из пакета «datasets». Данные содержат зависимости тормозного пути автомобиля (футы) от его скорости (мили в час). Данные получены в 1920 г. Постройте регрессионную модель и оцените длину тормозного пути при скорости 40 миль в час.