Линейная регрессия

Горюнов Ю.В., Дружков П.Н., Золотых Н.Ю., Половинкин А.Н.

6 октября 2013 г.

Содержание

1 Линейная регрессия

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

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

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} w_i (y_i - \sum_{j=1}^{d} \beta_j h_j(x_i) + \delta_i)^2 = \arg\min_{\beta} \|W(y - X\beta + \delta)\|_2^2, \quad (1)$$

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 $W = diag(\sqrt{w_1}, \ldots, \sqrt{w_n})$, δ — вектор-столбец смещений.

Для решения данной оптимизационной задачи предназначена функция `lm(formula, data, subset, weights, na.action, method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)`, работа которой основана на QR-разложении матрицы X.

Функция принимает следующие аргументы:

- `formula` — формула, описывающая восстанавливаемую зависимость;
- `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 β. `method = "model.frame"` обозначает, что будет лишь сформирован фрейм данных, содержащий фигурирующие в модели переменные, поиск коэффициентов β 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. Т.к. 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 δ.

Функция `lm` возвращает список со следующими элементами:

- `coefficients` — полученные коэффициенты $\hat{\beta}$;
- `fitted.values` — предсказания полученной модели на обучающей выборке;
- `residuals` — остатки $y_i - f(x_i; \hat{\beta})$;
- `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 $\sqrt{w_i}(y_i - f(x_i; \hat{\beta}))$;
- `coefficients` — матрица размеров $d^* \times 4$, столбцы которой содержат оценные коэффициенты $\hat{\beta}$, 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 $\hat{\sigma} = \sqrt{\frac{1}{n-d^*}\sum_{i=1}^{n} w_i(y_i - f(x_i; \hat{\beta}))^2}$;
- `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.

Полученные результаты также выводятся функцией `summary` на экран в удобочитаемом виде.

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

```
> library(datasets)
> air = airquality[, c("Ozone", "Solar.R", "Wind", "Temp")]
> f = lm(Ozone ~ ., data = air, subset = !is.na(Solar.R) &
    !is.na(Ozone))
> f

Call:
lm(formula = Ozone ~ ., data = air, subset = !is.na(Solar.R) &
    !is.na(Ozone))

Coefficients:
(Intercept)      Solar.R         Wind         Temp
  -64.34208      0.05982     -3.33359      1.65209

> summary(f)

Call:
lm(formula = Ozone ~ ., data = air, subset = !is.na(Solar.R) &
    !is.na(Ozone))

Residuals:
    Min      1Q  Median      3Q     Max
-40.485 -14.219  -3.551  10.097  95.619

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -64.34208   23.05472  -2.791  0.00623 **
Solar.R       0.05982    0.02319   2.580  0.01124 *
Wind         -3.33359    0.65441  -5.094 1.52e-06 ***
Temp          1.65209    0.25353   6.516 2.42e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.18 on 107 degrees of freedom
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.5948
F-statistic: 54.83 on 3 and 107 DF,  p-value: < 2.2e-16
```

Дадим интерпретацию полученным результатам. Для исследуемой зависимости была получена модель $Ozone = \hat{\beta}_0 + \hat{\beta}_1 \cdot Solar.R + \hat{\beta}_2 \cdot Wind + \hat{\beta}_3 \cdot Temp = -64.34208 + 0.05982 \cdot Solar.R - 3.33359 \cdot Wind + 1.65209 \cdot Temp$. Стандартные ошибки коэффициентов равны 23.05472, 0.02319, 0.65441 и 0.25353 соответственно. Для выполнения статистических тестов с нулевыми гипотезами о том, что $\beta_i = 0$, были подсчитаны значения t-статистик и p-value. На основе полченных результатов, можно сделать вывод, что при уровне значимости $\alpha = 0.01$ коэффициент β_1 следует признать незначимым для модели, а все остальные коэффициенты — значимыми. Для проведения статистического теста о значимости всей модели в целом (нулевая гипотеза $\beta_i = 0, i = \overline{1, d^*}$), 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 $p - value < 2 \times 10^{-16}$.

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:

$$\hat{\beta}^{ridge} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{d} \beta_j h_j(x_i))^2 + \lambda \sum_{j=1}^{d} \beta_j = \arg\min_{\beta} \|y - X\beta\|_2^2 + \|\beta\|_2^2. \tag{2}$$

Для решения данной задачи предназначена функция `lm.ridge(formula, data, subset, na.action, lambda = 0, model = FALSE, x = FALSE, y = FALSE, contrasts = NULL, ...)`, большинство параметров которой 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.

Функция `lm.ridge` возвращает список, содержащий, в том числе, следующие элементы:

- `coef` — матрица коэффициентов $\hat{\beta}^{ridge}$ для всех λ;
- `lambda` — вектор использованных значений λ.

Для графического отображения зависимости величин коэффициентов $\hat{\beta}^{ridge}$ 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.

```
> 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)
```

Полученный график изменения коэффициентов при изменении λ приведен на рис. 1.

3 Задания к лабораторной работе

1. Загрузите данные из файла `reglab1.txt`. Используя функцию `lm`, постройте регрессию (используйте разные модели). Выберите наиболее подходящую модель, объясните свой выбор.

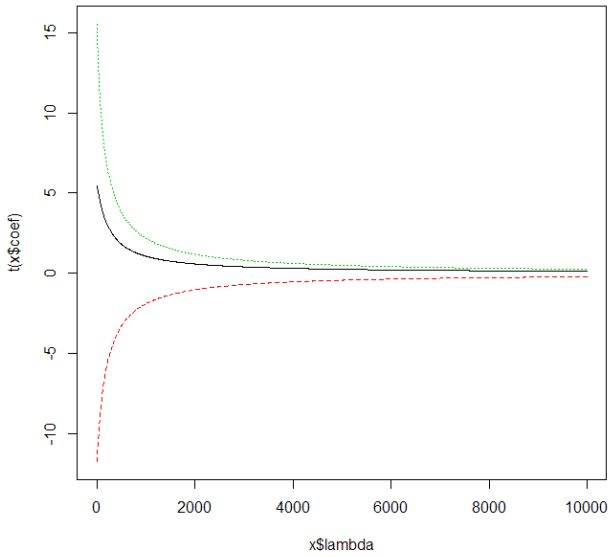

Рис. 1:

2. Реализуйте следующий алгоритм для уменьшения количества признаков, используемых для построения регрессии: для каждого $k \in 0, 1, \ldots, d$ выбрать подмножество признаков мощности k[1], минимизирующее остаточную сумму квадратов RSS. Используя полученный алгоритм, выберите оптимальное подможество признаков для данных из файла `reglab2.txt`. Объясните свой выбор. Дайте интерпретацию вычисленным значениям t-статистики и p-value для коэффициентов $\hat{\beta}$.

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 из файла `alligators.txt`. Выберите лучшую регресионную модель (возможно нелинейную), отражающую зависимость веса аллигатора от его длины.

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

[1] Для генерации всех возможных сочетаний по `m` элементов из некоторого множества `x` можно использовать функцию `combn(x, m, ...)`.