

МАШИННОЕ ОБУЧЕНИЕ
ЛАБОРАТОРНЫЙ ПРАКТИКУМ
(предварительный вариант)

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

14 октября 2007 г.

1. Практическое машинное обучение

1.1. Задача восстановления регрессии

1.1.1. Функция `lm`

Описание:

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

`formula` — символическое описание модели. Детали см. ниже.

`data` — фрейм данных, откуда берутся данные. Если не указано (или указано `NULL`), то все данные встречающиеся в модели — это векторы рабочего пространства. В противном случае данные — это поля указанного фрейма данных.

`subset` — вектор, определяющий подмножество данных, участвующих в модели. Необязательный параметр.

`weight` — вектор весов. Может быть или `NULL`, или числовым. В последнем случае веса используются в методе наименьших квадратов. Необязательный параметр.

`na.action` — функция, показывающая, что надо сделать с моделью, если в ней есть `NA`.

`method` — используемый метод. В настоящее время доступно только `"qr"`.

`model`, `x`, `y`, `qr` — логические значения. Если `TRUE`, то функция возвращает соответствующие компоненты модели: найденные коэффициенты β_j , матрицу данных \mathbf{X} , столбец ответов \mathbf{y} , QR -разложение.

`singular.ok` — если `FALSE`, то вырожденность (ранг матрицы \mathbf{X} меньше числа столбцов) приводит к ошибке.

Детали: `model` — формула для выбора математической модели. Рассмотрим некоторые примеры.

Формула	Модель
$Y \sim A$	$Y = \beta_0 + \beta_1 A$
$Y \sim -1 + A$	$Y = \beta_1 A$
$Y \sim A + I(A^2)$	$Y = \beta_0 + \beta_1 A + \beta_2 A^2$
$Y \sim \text{poly}(A, 2)$	$Y = \beta_0 + \beta_1 A + \beta_2 A^2$
$Y \sim A + B$	$Y = \beta_0 + \beta_1 A + \beta_2 B$
$Y \sim A:B$	$Y = \beta_0 + \beta_1 AB$
$Y \sim A*B$	$Y = \beta_0 + \beta_1 A + \beta_2 B + \beta_3 AB$
$Y \sim (A + B)^2$	$Y = \beta_0 + \beta_1 A + \beta_2 B + \beta_4 AB$

Общий вид формулы `model` следующий:

ответ \sim оп1 терм1 оп2 терм2 оп3 терм3 ...

где `ответ` — вектор y (или выражение, значение которого есть вектор).

`опi` — оператор, определяющий включение (+) или исключение (-) терма, при этом если `оп1` есть +, то его можно опустить

`терми` — вектор, матрица, 1, фактор или *формульное выражение*. Во всех случаях `терми` определяет коллекцию столбцов, которые необходимо включить в матрицу X или исключить из нее, причем 1 соответствует столбцу из единиц (по умолчанию всегда включен).

Формульное выражение состоит из факторов, векторов или матриц, соединенных формульными операторами. Возможны следующие формульные операторы.

$Y \sim M$ — см. выше.

$M_1 + M_2$ — см. выше.

$M_1 - M_2$ — см. выше.

$M_1:M_2$ — тензорное произведение M_1 и M_2 . Если M_1 и M_2 — факторы, то the “subclasses” factor.

$M_1 \%in\% M_2$ — аналогично предыдущему, но с другой кодировкой.

$M_1 * M_2$ — эквивалентно $M_1 + M_2 + M_1:M_2$.

M_1 / M_2 — эквивалентно $M_1 + M_2 \%in\% M_1$.

M^n — все термы из M и все их тензорные степени порядка n и ниже.

$I(M)$ — все операторы внутри M имеют обычный арифметический смысл.

Заметим, что формулы можно использовать и в качестве параметров других функций R. Например, функция

```
> x <- seq(-pi, pi, len = 100)
> plot(sin(x) ~ x, type = "l")
```

рисует график синуса.

Объект, возвращаемый функцией `lm`, имеет различные поля. Вот некоторые из них:

`coefficients` — вектор коэффициентов $\hat{\beta}_j$

`residuals` — вектор разностей $y - X\hat{\beta}$

`rank` — ранг матрицы X

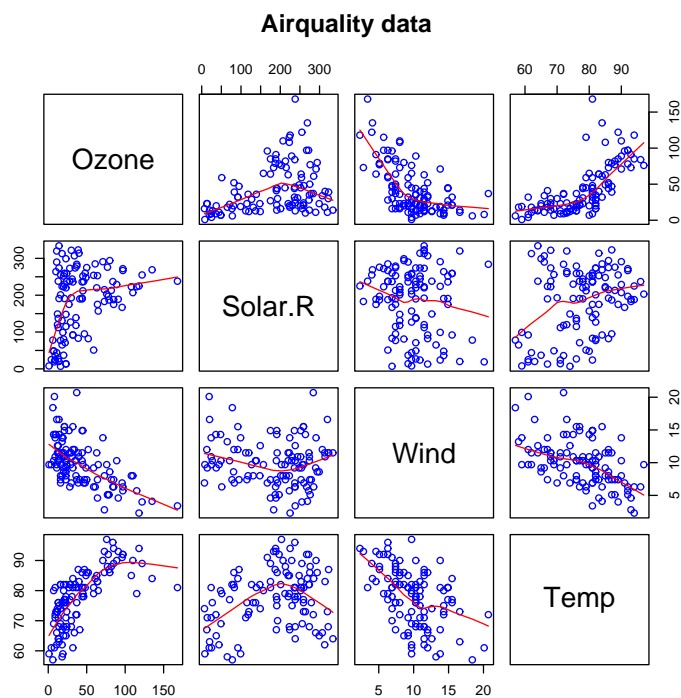


Рис. 1.1. Диаграммы рассеяния для каждой пары переменных Ozone, Solar.R, Wind, Temp

`fitted.values` — вычисленные значения $\mathbf{X}\hat{\beta}$
`qr` — QR -разложение
`qraux` — вспомогательная информация, необходимая для восстановления QR -разложения (для нахождения QR -разложения используется функция `qr`)
`pivot` — направляющие элементы при отыскании функции `qr`
`tol` — допуск, используемый при отыскании
`model` — матрица \mathbf{X}

1.1.2. Пример

Фрейм данных `airquality` из пакета `datasets` содержит ежедневные замеры показателей качества воздуха в Нью-Йорке с мая по сентябрь 1973 г.: содержание озона `Ozone`, солнечное излучение `Solar.R`, средняя скорость ветра `Wind`, максимальная дневная температура `Temp`. Подробности см. в документации к пакету.

Сначала нарисуем диаграммы рассеяния для каждой пары переменных:

```

> library(datasets)
> air <- airquality[, c("Ozone", "Solar.R", "Wind",
                        "Temp")]

```

```

> air <- air[!is.na(air$Solar.R) & !is.na(air$Ozone), ]
> pairs(air, panel = panel.smooth,
        main = "Airquality data", col = "blue")

```

Результат приведен на рис. 1.1.

Теперь рассмотрим регрессионную модель

$$\text{Ozone} = \beta_0 + \beta_1 \cdot \text{Solar.R} + \beta_2 \cdot \text{Wind} + \beta_3 \cdot \text{Temp}$$

```

> fit = lm(Ozone ~ ., data = air)
> fit

```

Call:

```
lm(formula = Ozone ~ ., data = air)
```

Coefficients:

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

Таким образом, найдены следующие значения $\hat{\beta}_0 = -64.34208$, $\hat{\beta}_1 = 0.05982$, $\hat{\beta}_2 = -3.33359$, $\hat{\beta}_3 = 1.65209$. Больше информации можно получить с помощью функции *summary*:

```

> summary(fit)

```

Call:

```
lm(formula = Ozone ~ ., data = air)
```

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

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. В таблице в первом столбце приведены значения $\hat{\beta}_j$, во втором — их стандартные ошибки $se \hat{\beta}_j$, в третьем — значения для t -статистики (стандартные коэффициенты): $t_j = \hat{\beta}_j / se \hat{\beta}_j$ и в четвертом — соответствующие p -value. Остаточная дисперсия (остаточная

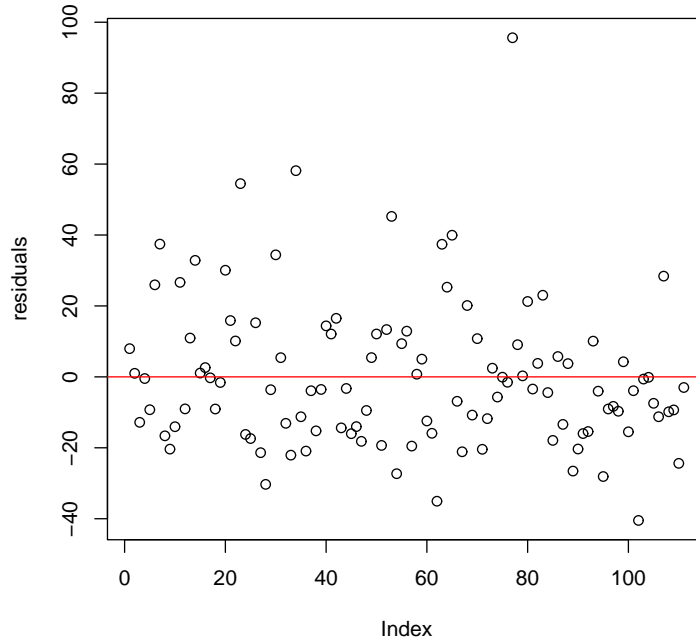


Рис. 1.2. График остатков

стандартная ошибка) равна $\hat{\sigma} = 21.18$. Задача имеет $N-p-1 = 111-3-1 = 107$ степеней свободы. Коэффициент детерминации равен $R^2 = 0.6059$. «Подправленный» (adjusted) коэффициент детерминации, равный 0.5948, учитывает возможную перепогонку данных. Для сравнения модели с `Ozone = const` вычисляется значение статистики $F = 54.83$. Данная статистика должна иметь распределение Фишера $F_{3,107}$. Полученное p -value меньше машинного эпсилон.

Можно нарисовать график остатков:

```
> plot(fit$residuals, ylab = "residuals", log = "")
> abline(0, 0, col = "red")
```

Результат приведен на рис. 1.2.

Запрограммируйте методы отыскания коэффициентов $\hat{\beta}_j$, $\hat{\beta}_j$, t_j , $\hat{\sigma}$ и т. д. Сравните их с полученными с помощью функции `lm` значениями. Например, для нахождения коэффициента детерминации R^2 можно воспользоваться следующей «программой»:

```
> (cor(air$Ozone, fit$fitted.values))^2
```

Для имеющихся данных рассмотрите другие линейные модели. Например,

$$\text{Ozone} = \beta_0 + \beta_1 \cdot \text{Solar.R} + \beta_2 \cdot \text{Wind} + \beta_{31} \cdot \text{Temp} + \beta_{32} \cdot \text{Temp}^2$$

```
> fit <- lm(Ozone ~ Solar.R + Wind + poly(Temp, 2),  
           data = air)
```

Попробуйте подобрать более подходящую модель.

1.1.3. Регуляризация

```
lm.ridge(formula, data, lambda = 0,...)
```

`formula` — символическое описание модели поиска

`data` — база, из которой берутся данные для загрузки в формулу. Необходимо указывать, если в формуле есть данные из этой базы.

`lambda` — параметр регуляризации (скаляр или вектор)

1.1.4. Задания для лабораторной работы

- 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 $k \in \{0, 1, \dots, p\}$ выбрать подмножество признаков мощности k , минимизирующее $RSS(\beta)$. Используя полученный алгоритм, выберите оптимальное подмножество признаков для данных из файла `reglab2.txt`. Объясните свой выбор.
- 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 регрессионную модель (возможно нелинейную), отражающую зависимость веса аллигатора от его длины.