Министерство образования и науки Российской Федерации

Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

И.И. Кручинин
(к.т.н. доцент)

лекция «Методы восстановления регрессии при обучении по прецедентам»
по курсу «Введение в машинное обучение»

Калуга - 2018

Краткое содержание:

1. Метод наименьших квадратов
2. Теорема Гаусса – Маркова

3. Непараметрическая регрессия, ядерное сглаживание, формула Надарая – Ватсона
4. Оптимизация ширины окна. Алгоритм локально взвешенного сглаживания
5. Проблема краевых эффектов
6. Линейная регрессия
7. Сингулярное разложение
8. Проблема мультиколлинеарности
9. Гребневая регрессия
10. Лассо Табширани
11. Линейная монотонная регрессия
12. Метод главных компонент
13. Преобразование Карунена – Лоэва
14. Нелинейные методы восстановления регрессии, метод Ньютона – Рафсона
15. Нелинейные одномерные преобразования признаков, метод BackFitting
16. Обобщенные линейные модели
17. Неквадратичные функции потерь
18. Логистическая регрессия
19. Метод опорных векторов в задачах регрессии
20. Выбор «лучшей» регрессионной модели

Методы восстановления регрессии

Задачу обучения по прецедентам при $Y = \mathbb{R}$ принято называть задачей восстановления регрессии. Основные обозначения остаются прежними. Задано пространство объектов X и множество возможных ответов Y. Существует неизвестная целевая зависимость $y^*: X \to Y$, значения которой известны только на объектах обучающей выборки $X^\ell = (x_i, y_i)_{i=1}^\ell$, $y_i = y^*(x_i)$. Требуется построить алгоритм, который в данной задаче принято называть «функцией регрессии» $a: X \to Y$, аппроксимирующий целевую зависимость y^*.

Метод наименьших квадратов

Пусть задана модель регрессии — параметрическое семейство функций $g(x,\alpha)$, где $\alpha \in \mathbb{R}^p$ — вектор параметров модели. Определим функционал качества аппроксимации целевой зависимости на выборке X^ℓ как сумму квадратов ошибок:

$$Q(\alpha, X^\ell) = \sum_{i=1}^{\ell} \big(g(x_i, \alpha) - y_i\big)^2 . \qquad (5.1)$$

Обучение по методу наименьших квадратов (МНК) состоит в том, чтобы найти вектор параметров α^*, при котором достигается минимум среднего квадрата ошибки на заданной обучающей выборке X^ℓ:

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^p} Q(\alpha, X^\ell). \qquad (5.2)$$

Стандартный способ решения этой оптимизационной задачи — воспользоваться необходимым условием минимума. Если функция $g(x,\alpha)$ достаточное число раз дифференцируема по α, то в точке минимума выполняется система pуравнений относительно pнеизвестных:

$$\frac{\partial Q}{\partial \alpha}(\alpha, X^{\ell}) = 2\sum_{i=1}^{\ell}\big(g(x_i,\alpha) - y_i\big)\frac{\partial g}{\partial \alpha}(x_i,\alpha) = 0 \quad . \qquad (5.3)$$

МНК-оценка вектора параметров

Классическим вариантом оценивания параметров по наблюдениям, осложненным шумами, является метод наименьших квадратов (МНК). В этом случае наблюдения заданы в дискретные моменты времени, причем m=1, так что $z°(k)$ – скалярная величина. Матрица $C(k)$ превращается в матрицу-строку. Для того чтобы сохранить традиционные для МНК обозначения, примем

$\{C(k) = (a_{k1}, a_{k2}, ..., a_{kn}),\ k = 1,2, ..., N,$

$z°(k) = l_k,$

$r(k) = r_k.$

Таким образом уравнения наблюдений принимают вид

$l_k = a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n + r_k,\ k = 1, 2, ..., N.$

Пусть

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \cdots & a_{Nn} \end{pmatrix},\ l = \begin{pmatrix} l_1 \\ l_2 \\ \cdot \\ \cdot \\ l_N \end{pmatrix},\ r = \begin{pmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ r_N \end{pmatrix},\ x = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_3 \end{pmatrix}.$$

В матричном виде уравнение принимает вид

$$l = Ax + r.$$

Допустим, что мы располагаем оценкой $\hat{x}$, но погрешность r, естественно, нам неизвестна. Тогда можно предвычислить наблюдения

$$z^c(k) = a_{k1}\hat{x}_1 + a_{k2}\hat{x}_2 + ... + a_{kn}\hat{x}_n,$$

или в матричной форме

$\hat{l} = A\hat{x}.$

Теперь можно построить остаточные разности

$v = l - \hat{l},$

указывающие на рассогласование наблюдений $(°)$ и вычисленных значений (c). Задача МНК-оценивания состоит в таком выборе вектора параметров $\hat{x}$, чтобы минимизировать критерий качества, заданный в виде квадратичной формы

$$I = \sum_{i=1}^{N}\sum_{j=1}^{N} q_{ij}v_iv_j,\ q_{ij} = q_{ji}.$$

В матричном виде этот критерий имеет вид

$I = v^T Q v,$

где

$$Q = \left\| q_{ij} \right\|.$$

Здесь Q – пока не определенная весовая матрица, обладающая свойством симметрии.

Дифференцируя I по ξ и приравнивая результат нулю, получим нормальные уравнения

$$\begin{aligned} &\frac{\partial I}{\partial \xi} = \frac{\partial}{\partial \xi}\left[(l - A\xi)^T Q(l - A\xi)\right] = -2A^T Q(l - A\xi) = 0, \\ &A^T QA\xi = A^T Ql. \end{aligned} \qquad (10.1)$$

Следовательно, МНК-оценка вектора параметров имеет вид

$$\xi = \left(A^T QA\right)^{-1} A^T Ql \qquad (10.2)$$

Погрешность оценки равна

$$\left(A^T QA\right)^{-1} A^T Qr.$$

Определим ковариационную матрицу погрешности оценки

$$\mathrm{P}_x = \left\langle (\xi - x)(\xi - x)^T \right\rangle = \left(A^T QA\right)^{-1} A^T Q\left\langle rr^T \right\rangle QA\left(A^T QA\right)^{-1} =$$

$$= \left(A^T QA\right)^{-1} A^T QRQA\left(A^T QA\right)^{-1},$$

где R – ковариационная матрица ошибок измерений: $R(k-l) = \left\| \langle r_k r_l \rangle \right\|$. Полагая измерения независимыми, мы получим

$$\langle r_k r_l \rangle = \begin{cases} \sigma_k^2, & k = l, \\ 0, & k \neq l, \end{cases}$$

поэтому R – диагональная матрица

$$R = \begin{pmatrix} \sigma_1^2 & 0 & 0 & \ldots & 0 \\ 0 & \sigma_2^2 & 0 & \ldots & 0 \\ 0 & 0 & \sigma_3^2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \sigma_N^2 \end{pmatrix},$$

элементами которой являются дисперсии ошибок измерений. Очевидно, что для

$$Q = R^{-1} = \begin{pmatrix} \sigma_1^{-2} & 0 & 0 & \ldots & 0 \\ 0 & \sigma_2^{-2} & & \ldots & 0 \\ 0 & 0 & \sigma_3^{-2} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & \sigma_N^{-2} \end{pmatrix} \qquad (10.3)$$

будем иметь

$$\begin{aligned} &QR = E, \\ &\mathrm{P}_x = \left(A^T QA\right)^{-1}. \end{aligned} \qquad (10.4)$$

В частном, но довольно распространенном случае, когда измерения равноточны, ковариационная матрица ошибки измерений имеет вид

$$R = \sigma_0^2 E,$$

Так что

$$\mathrm{P}_x = \sigma_0^2\left(A^T A\right)^{-1}.$$

Величина σ_0^2 называется *дисперсией единицы веса*, диагональные элементы матрицы P_x – *дисперсиями погрешностей составляющих вектора x.*

Формула (10.2) показывает, что весовая матрица Q может быть определена с точностью до постоянного множителя. Действительно, заменяя Q на $Q' = \sigma_0^2 Q$, получим тотже результат

$$\hat{x} = \left(A^T Q' A\right)^{-1} A^T Q' l = \left(A^T \sigma_0^2 Q A\right)^{-1} A^T \sigma_0^2 Q l = \left(A^T Q A\right)^{-1} A^T Q l.$$

Обозначим через $p_k = \sigma_0^2 / \sigma_k^2$ вес измерения в момент $t = t_k$. Весовая матрица (10.3) будет иметь вид

$$Q = \begin{pmatrix} p_1 & 0 & ... & 0 \\ & p_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & p_k \end{pmatrix}, \qquad (10.5)$$

а ковариационная матрица оценки $\hat{x}$ –

$$\mathrm{P}_x = \sigma_0^2 \left(A^T Q A\right)^{-1} - \qquad (10.6)$$

будет отличаться от вида (10.5) множителем σ_0^2.

Для априорной оценки величину дисперсии единицы веса σ_0^2 нужно задать. Однако при достаточно большом объеме измерений оценку этой величины можно определить и апостериорно, т.е. после обработки данных измерений. Приведем эту формулу без вывода

$$\hat{\sigma}_0^2 = \frac{\sum_{k=1}^{N} p_k v_k^2}{N - n} .$$

Отличие $\hat{\sigma}_0^2$ от σ_0^2 будет говорить о том, что

1) число избыточных наблюдений $N - n$ недостаточно велико,
2) неудачно выбрана система весов наблюдений, дисперсии σ_k^2 не отражают реальную точность,
3) измерения содержат система тические ошибки.

Наконец, результат МНК-оценивания записывают в виде

$$x = \hat{x} \pm \varepsilon_x,$$

где $\varepsilon_x = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_n)^T$ – средняя квадратическая ошибка вектора x,

$$\varepsilon_j = \sqrt{\frac{\sum_{k=1}^{n} p_k v_k^2}{N - n} q_{jj}} ,$$

q_{jj} – j-й диагональный элемент матрицы $\left(A^T Q A\right)^{-1}$.

Теорема Гаусса–Маркова

Теорема Гаусса–Маркова утверждает, что МНК-оценка является наилучшей в том смысле, что ее ковариационная матрица является наименьшей.

Квадратная матрица C_1 больше другой квадратной матрицы C_2, если разность $\mathrm{C}_1 - \mathrm{C}_2$ – положительно определенная матрица. Напомним, что матрица C является положительно определенной, если для любого вектора y имеет место неравенство $y^T C y > 0$.

Итак, МНК-оценка x имеет вид

$$\text{€} = \left(A^T QA\right)^{-1} A^T Ql.$$

Обозначим $\mathrm{B}_0 = \left(A^T QA\right)^{-1} A^T Q$. Тогда $\text{€} = \mathrm{B}_0 l$. Эта оценка несмещенная, ибо при $l = Ax + r$ получим

$$\text{€} = x + \left(A^T QA\right)^{-1} A^T Qr = x + \mathrm{B}_0 r,$$

$$\langle \text{€} \rangle = x + \left(A^T QA\right)^{-1} A^T Q\langle r \rangle = x,$$

ибо $\langle r \rangle = 0$. Предположим, что существует другая несмещенная оценка $\tilde{x}$, для которой выполняется равенство

$$\tilde{x} = B_1 l = B_1 (Ax + r).$$

Для несмещенности необходимо

$$\langle \tilde{x} \rangle = B_1 \langle l \rangle = B_1 Ax + B_1 \langle r \rangle = B_1 Ax,$$

т.е $B_1 A. = E$. Итак, матрица B_1 должна быть псевдообратной матрицей по отношению к матрице A. Аналогично $B_0 A = E$, поэтому $(B_1 - B_0)A = 0$.

Вычислим ковариационную матрицу оценки $\tilde{x}$:

$$\mathrm{P}_{\tilde{x}} = \left\langle (\tilde{x} - x)(\tilde{x} - x)^T \right\rangle = \left\langle B_1 rr^T B_1^T \right\rangle = B_1 RB_1^T,$$

где R – ковариационная матрица шумов измерения. Аналогично

$$\mathrm{P}_{\text{€}} = B_0 RB_0^T.$$

Образуем разность и выполним небольшие преобразования

$$\mathrm{P}_{\tilde{x}} - \mathrm{P}_{\text{€}} = B_1 RB_1^T - B_0 RB_0^T = [B_0 - (B_1 - B_0)]R\left[B_0^T + \left(B_1^T - B_0^T\right)\right] - B_0 RB_0^T =$$

$$= (B_1 - B_0)R(B_1 - B_0)^T + B_0 R\left(B_1^T - B_0^T\right) + (B_1 - B_0)RB_0^T.$$

Однако

$$(B_1 - B_0)RB_0^T = (B_1 - B_0)RQA\left(A^T QA\right)^{-1} = 0,$$

так как $RQ = \sigma_0^2 E$, а $(B_1 - B_0)A = 0$. Матрица $B_0 R\left(B_1^T - B_0^T\right)$ также равна нулю, так как она является транспонированной к нулевой матрице $(B_1 - B_0)RB_0^T$. Итак,

$$\mathrm{P}_{\tilde{x}} - \mathrm{P}_{\text{€}} = (B_1 - B_0)R(B_1 - B_0)^T.$$

Полученная матрица неотрицательно определена. Это легко доказать. Из теории случайных векторов известно, что ковариационная матрица случайного вектора положительно определена. Рассмотрим следующий случайный вектор

$$\xi = (B_1 - B_0)r.$$

Ковариационная матрица этого вектора имеет вид

$$\mathrm{P}_{\xi} = \left\langle \xi\xi^T \right\rangle \left\langle (B_1 - B_0)rr^T (B_1 - B_0)^T \right\rangle = (B_1 - B_0)R(B_1 - B_0)^T \geq 0.$$

Матрица P_{ξ} принимает минимальное значение, равное нулю при $B_1 = B_0$. Следовательно,

$$\mathrm{P}_{\tilde{x}} - \mathrm{P}_{\text{€}} \geq 0, \ \mathrm{P}_{\text{€}} \leq \mathrm{P}_{\tilde{\mathrm{x}}}.$$

Полученное неравенство и доказывает теорему.

Пример 1. Решить систему уравнений с помощью метода наименьших квадратов:

$$
\begin{aligned}
x_1 &- x_2 = 1{,}8 + 0{,}075\\
x_1 &+ x_2 = 0{,}1 - 0{,}025\\
x_1 &+ 2x_2 = -1{,}1 - 0{,}150\\
2x_1 &+ x_2 = 1{,}0 - 0{,}175\\
2x_1 &+ 3x_2 = -1{,}0 - 0{,}175\\
3x_1 &+ 2x_2 = 1{,}2 + 0{,}150
\end{aligned}
$$

Точные значения x_1 и x_2 известны и равны соответственно 1,000 и –1,000. Выборочная дисперсия погрешностей правой части системы равна

$$\sigma_0^2 = \frac{1}{6}\left[(-0{,}2)^2 + (0{,}1)^2 + (-0{,}1)^2 + (0{,}0)^2 + (0{,}0)^2 + (0{,}2)^2\right] = 0{,}017, \quad \sigma_0 = 0{,}13.$$

Следуя схеме МНК, находим нормальные уравнения

$$\begin{pmatrix} 20 & 16 \\ 16 & 20 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4{,}4 \\ -3{,}5 \end{pmatrix},$$

откуда

$$x_1 = 1{,}000,$$

$$x_2 = -0{,}975.$$

Определим остаточные разности $v_i = l_i - (a_{i1}x_1 + a_{i2}x_2)$. Значения их приведены справа от системы уравнений. Апостериорную оценку дисперсии единицы веса вычислим по формуле

$$\sigma_0^2 = \frac{\sum v_i^2}{6-2} = 0{,}022, \quad \sigma_0 = 0{,}15.$$

Отличие апостериорной оценки от априорной объясняется недостаточным объемом данных, хотя оно, по-видимому, не значимо. Ковариационная матрица системы имеет вид

$$P_x = \sigma_0^2\begin{pmatrix} 20 & 16 \\ 16 & 20 \end{pmatrix}^{-1} = \sigma_0^2\begin{pmatrix} 0{,}1389 & -0{,}1111 \\ -0{,}1111 & 0{,}1389 \end{pmatrix}.$$

Следовательно,

$$x_1 = 1{,}000 \pm \sigma_0\sqrt{0{,}1389} = 1{,}00 \pm 0{,}05,$$

$$x_2 = -0{,}975 \pm \sigma_0\sqrt{0{,}1389} = -0{,}98 \pm 0{,}05.$$

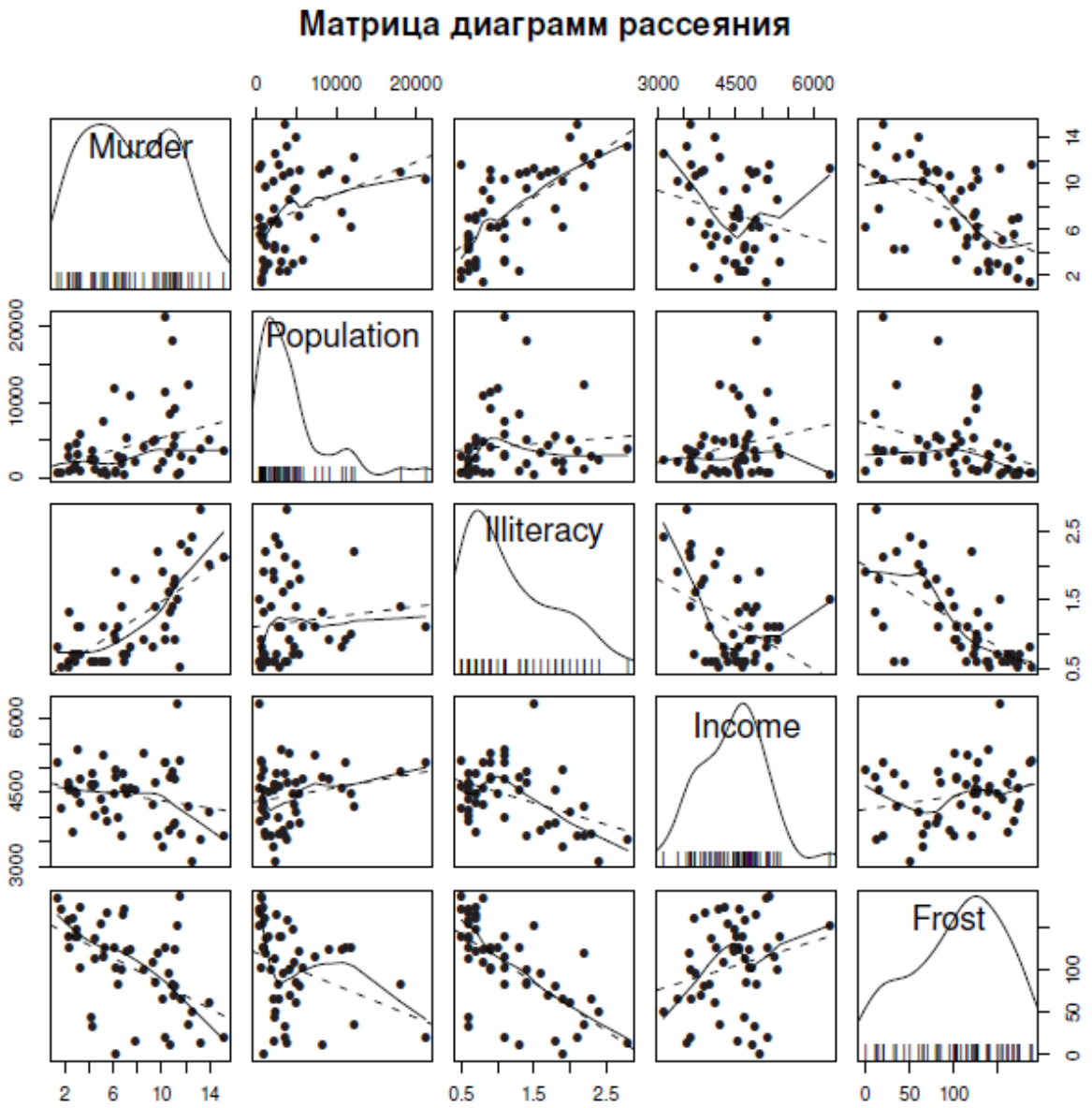


Рис. 8.4. Матрица диаграмм рассеяния значений зависимой и независимых переменных для данных о штатах, включающая регрессионные прямые и сглаживающие линии, а также распределения переменных (диаграмма ядерной оценки функции плотности и график-щетка)

Непараметрическая регрессия: ядерное сглаживание

Значение $a(x)$ вычисляется для каждого объекта xпо нескольким ближайшим к нему объектам обучающей выборки. Чтобы можно было говорить о «близости» объектов, на множестве Xдолжна быть задана функция расстояния $\rho(x,x')$.

Формула Надарая–Ватсона

Возьмём самую простую модель регрессии, какая только возможна — константу $g(x,\alpha) = \alpha$, $\alpha \in R$. Но при этом, чтобы не получить тривиального решения, введём веса объектов $w_i(x)$, зависящие от того объекта x, в котором мы собираемся вычислять значение $a(x) = g(x,\alpha)$.

Можно сказать и так, что обучение регрессионной модели будет производиться отдельно в каждой точке xпространства объектов X.

Чтобы вычислить значение $a(x) = \alpha$для произвольного$x \in X$, воспользуемся методом наименьших квадратов:

$$Q(\alpha; X^\ell) = \sum_{i=1}^{\ell} w_i(x)\big(\alpha - y_i\big)^2 \to \min_{\alpha \in \mathbb{R}}.$$

Зададим веса w_iобучающих объектов так, чтобы они убывали по мере увеличения расстояния $\rho(x,x_i)$. Для этого введём невозрастающую, гладкую, ограниченную функцию K: $[0,\infty) \to [0,\infty)$, называемую ядром:

$$w_i(x) = K\left(\frac{\rho(x, x_i)}{h}\right).$$

Параметр hназывается шириной ядра или шириной окна сглаживания. Чем меньше h, тем быстрее будут убывать веса $w_i(x)$ по мере удаления x_iот x.

Приравняв нулю производную $\frac{\partial Q}{\partial \alpha} = 0$, получим формулу ядерного сглаживания Надарая–Ватсона:

$$a_h(x; X^\ell) = \frac{\sum\limits_{i=1}^{\ell} y_i w_i(x)}{\sum\limits_{i=1}^{\ell} w_i(x)} = \frac{\sum\limits_{i=1}^{\ell} y_i K\left(\frac{\rho(x,x_i)}{h}\right)}{\sum\limits_{i=1}^{\ell} K\left(\frac{\rho(x,x_i)}{h}\right)}. \qquad (5.4)$$

Эта формула интуитивно очевидна: значение $a(x)$ есть среднее y_iпо объектам x_i, ближайшим к x.

В одномерном случае $X = \mathrm{R}^1$ метрика задаётся как $\rho(x,x_i) = |x - x_i|$. При этом строгим обоснованием формулы (5.4) служит следующая теорема, аналогичная Теореме 2.3 о непараметрическом восстановлении плотности.

Теорема 5.1 ([25]). *Пусть выполнены следующие условия:*

1) выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$ простая, получена из распределения $p(x, y)$;

2) ядро $K(r)$ удовлетворяет ограничениям $\int_0^\infty K(r)\,dr < \infty$ и $\lim\limits_{r\to\infty} rK(r) = 0$;

3) восстанавливаемая зависимость, определяемая плотностью $p(y|x)$, удовлетворяет при любом $x \in X$ ограничению $\mathsf{E}(y^2|x) = \int_Y y^2 p(y|x)\,dy < \infty$;

4) последовательность h_ℓ такова, что $\lim\limits_{\ell\to\infty} h_\ell = 0$ и $\lim\limits_{\ell\to\infty} \ell h_\ell = \infty$.

Тогда имеет место сходимость по вероятности: $a_{h_\ell}(x; X^\ell) \xrightarrow{P} \mathsf{E}(y|x)$ в любой точке $x \in X$, в которой $\mathsf{E}(y|x)$, $p(x)$ и $\mathsf{D}(y|x)$ непрерывны и $p(x) > 0$.

Таким образом, для широкого класса ядер оценка Надарая–Ватсона сходится к ожидаемому значению восстанавливаемой зависимости при неограниченном увеличении длины выборки ℓ и одновременном уменьшении ширины окна h.

Выбор ядра и ширины окна

Ядерное сглаживание — это довольно простой метод с точки зрения реализации. Обучение алгоритма $a_h(x;X^\ell)$ сводится к запоминанию выборки, подбору ядра Kи ширины окна h.

Выбор ядра Kмало влияет на точность аппроксимации, но определяющим образом влияет на степень гладкости функции $a_h(x)$. В одномерном случае функция $a_h(x)$ столько же раз дифференцируема, сколько и ядро $K(r)$. Часто используемые ядра показаны на Рис. 3. Для ядерного сглаживания чаще всего $K_G(r) = \exp\left(-\frac{1}{2}r^2\right)$ $K_Q(r) = (1-r^2)^2\left[|r|<1\right]$ берут гауссовское ядро или квартическое.

Если ядро $K(r)$ финитно, то есть $K(r) = 0$ при $r>1$, то ненулевые веса получат только те объекты x_i, для которых $\rho(x,x_i) < h$. Тогда в формуле (5.4) достаточно суммировать только по ближайшим соседям объекта x. В одномерном случае $X = \mathrm{R}^1$ для эффективной реализации этой идеи выборка должна быть упорядочена по возрастанию x_i. В общем случае необходима специальная структура данных, позволяющая быстро находить множество ближайших соседей для любого объекта x.

Выбор ширины окна hрешающим образом влияет на качество восстановления зависимости. При слишком узком окне ($h\to 0$) функция $a_h(x)$ стремится пройти через все точки выборки, реагируя на шум и претерпевая резкие скачки. При слишком широком окне функция чрезмерно сглаживается и в пределе $h\to\infty$ вырождается в константу. Таким образом, должно существовать оптимальное значение ширины окна $h*$ — компромисс между точностью описания выборки и гладкостью аппроксимирующей функции.

Проблема локальных сгущений возникает, когда объекты выборки распределены неравномерно в пространстве X. В областях локальных сгущений оптимальна меньшая ширина окна, чем в областях разреженности. В таких случаях используется окно переменной ширины $h(x)$, зависящей от объекта x. Соответственно, веса вычисляются по формуле $w_i(x) = K\left(\frac{\rho(x,x_i)}{h(x)}\right)$.

Самый простой способ — взять в качестве ширины окна $h(x)$ расстояние от объекта xдо его k+ 1-го соседа: $h_k(x) = \rho(x, x_x^{(k+1)})$. Недостаток этого способа в том, что функция $h_k(x)$ является непрерывной, но не гладкой, поэтому у функций $w_i(x)$ и $a_{hk}(x)$ будут разрывные первые производные, даже если ядро гладкое. Для устранения этого недостатка можно сгладить саму функцию $h_k(x)$ по узлам равномерной сетки, при постоянной ширине окна и каком-либо гладком ядре, скажем, K_Q.

Оптимизация ширины окна. Чтобы оценить при данномhили kточность локальной аппроксимации в точке x_i, саму эту точку необходимо исключить из обучающей выборки. Если этого не делать, минимум ошибки будет достигаться при $h\to 0$. Такой способ оценивания называется скользящим контролем с исключением объектов по одному (leave-one-out, LOO):

$$\mathrm{LOO}(h, X^\ell) = \sum_{i=1}^{\ell} \big(a_h\big(x_i; X^\ell \backslash \{x_i\}\big) - y_i\big)^2 \to \min_h,$$

где минимизация осуществляется по ширине окна hили по числу соседей k.

Проблема выбросов: робастная непараметрическая регрессия

Оценка Надарайя–Ватсона крайне чувствительна к большим одиночным выбросам. Идеяобнаружениявыбросовзаключается в том, чточембольшевеличина

Алгоритм 5.1. LOWESS — локально взвешенное сглаживание.

Вход:
X^ℓ — обучающая выборка;
Выход:
коэффициенты $\gamma_i,\ \ i = 1, \ldots, \ell$;

1: инициализация: $\gamma_i := 1,\ \ i = 1, \ldots, \ell$;
2: **повторять**
3: вычислить оценки скользящего контроля на каждом объекте:

$$a_i := a_h\big(x_i; X^\ell \backslash \{x_i\}\big) = \frac{\sum\limits_{j=1,\, j\neq i}^{\ell} y_j \gamma_j K\big(\frac{\rho(x_i, x_j)}{h(x_i)}\big)}{\sum\limits_{j=1,\, j\neq i}^{\ell} \gamma_j K\big(\frac{\rho(x_i, x_j)}{h(x_i)}\big)}, \quad i = 1, \ldots, \ell$$

4: вычислить коэффициенты γ_i:
$\gamma_i := \tilde{K}\big(|a_i - y_i|\big); \quad i = 1, \ldots, \ell;$
5: **пока** коэффициенты γ_i не стабилизируются;

выбросом, $|a_h(x_i; X^\ell \backslash \{x_i\}) - y_i|$ и тем меньше должен быть его вес. Эти соображения приводят к идееошибки ε_i=, тем в большей степени прецедент (x_i,y_i) является
домножить веса $w_i(x)$ на коэффициенты $\gamma_i = \tilde{K}(\varepsilon_i)$, где $\tilde{K}$ — ещё одно ядро, вообще говоря, отличное от $K(r)$.

Коэффициенты γ_i, как и ошибки ε_i, зависят от функции a_h, которая, в свою очередь, зависит от γ_i. Разумеется, это не «порочный круг», а хороший повод для организации итерационного процесса, см. Алгоритм 5.1. На каждой итерации строится функция a_h, затем уточняются весовые множители γ_i. Как правило, этот процесс сходится довольно быстро. Он называется локально взвешенным сглаживанием (locallyweightedscatterplotsmoothing, LOWESS) .

Методы восстановления регрессии, устойчивые к шуму в исходных данных, называют робастными, что означает «разумный, здравый» (robust).

Возможны различные варианты задания ядра $\tilde{K}(\varepsilon)$.

Жёсткая фильтрация: строится вариационный ряд ошибок $\varepsilon^{(1)} \leqslant \cdots \leqslant \varepsilon^{(\ell)}$, и отбрасывается некоторое количество tобъектов с наибольшей ошибкой. Это соответствует ядру $\tilde{K}(\varepsilon) = \big[\varepsilon \leqslant \varepsilon^{(\ell-t)}\big]$.

Мягкая фильтрация : используется квартическое ядро $\tilde{K}(\varepsilon) = K_Q\big(\frac{\varepsilon}{6\,\mathrm{med}\{\varepsilon_i\}}\big)$, где $\mathrm{med}\{\varepsilon_i\}$ — медиана вариационного ряда ошибок.

Проблема краевых эффектов

В одномерном случае $X = \mathbb{R}^1$ часто наблюдается значительное смещение аппроксимирующей функции $a_h(x)$ от истинной зависимости $y^*(x)$ вблизи минимальных и максимальных значений x_i. Смещение возникает, когда объекты выборки x_iрасполагаются только по одну сторону (а не вокруг) объекта x. Чем больше размерность пространства объектов, тем чаще возникает такая ситуация.

Для решения этой проблемы зависимость аппроксимируется в окрестности точки $x \in X$не константой $a(u) = \alpha$, а линейной функцией $a(u) = \alpha(u - x) + \beta$.

Введём для краткости сокращённые обозначения $w_i = w_i(x)$, $d_i = x_i - x$и запишем задачу наименьших квадратов:

$$Q(\alpha, \beta; X^\ell) = \sum_{i=1}^{\ell} w_i\big(\alpha d_i + \beta - y_i\big)^2 \to \min_{\alpha,\beta \in \mathbb{R}}.$$

Приравнивая нулю производные $\frac{\partial Q}{\partial \alpha} = 0$ и $\frac{\partial Q}{\partial \beta} = 0$, получим систему линейных уравнений 2×2, решение которой даёт аналог формулы Надарая–Ватсона:

$$a_h(x; X^\ell) = \frac{\sum\limits_{i=1}^{\ell} w_i d_i^2 \sum\limits_{i=1}^{\ell} w_i y_i - \sum\limits_{i=1}^{\ell} w_i d_i \sum\limits_{i=1}^{\ell} w_i d_i y_i}{\sum\limits_{i=1}^{\ell} w_i \sum\limits_{i=1}^{\ell} w_i d_i^2 - \Big(\sum\limits_{i=1}^{\ell} w_i d_i\Big)^2}.$$

В многомерном случае $X = \mathbb{R}^n$для вычисления коэффициентов в линейной форме $a(u) = \alpha^{\mathrm{T}}(u - x) + \beta$приходится решать задачу многомерной линейной регрессии (см. ниже). Причём она должна решаться заново для каждой точки $x \in X$, что сопряжено с большим объёмом вычислений.

Линейная регрессия

Пусть каждому объекту соответствует его признаковое описание $\big(f_1(x), \ldots, f_n(x)\big)$, где $f_j: X \to \mathbb{R}$ — числовые признаки, $j = 1,\ldots,n$. Линейной моделью регрессии называется линейная комбинация признаков с коэффициентами $\alpha \in \mathbb{R}^n$:

$$g(x, \alpha) = \sum_{j=1}^{n} \alpha_j f_j(x).$$

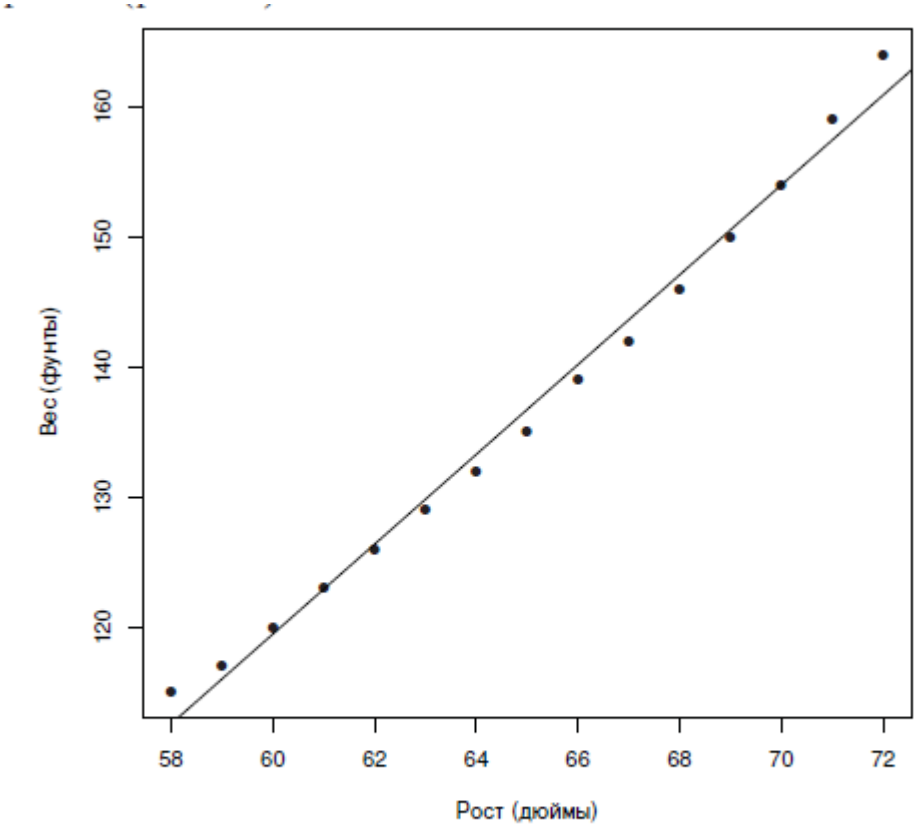


Рис. 8.1. Диаграмма рассеяния с регрессионной прямой для значений веса, предсказанных по значениям роста

Введём матричные обозначения: $F = \left(f_j(x_i)\right)_{\ell \times n}$ — матрица объекты–признаки; $y = \left(y_i\right)_{\ell \times 1}$ — целевой вектор; $\alpha = \left(\alpha_j\right)_{n \times 1}$ — вектор параметров. В матричных обозначениях функционал Q принимает вид

$$Q(\alpha) = \|F\alpha - y\|^2.$$

Запишем необходимое условие минимума (5.3) в матричном виде:

$$\frac{\partial Q}{\partial \alpha}(\alpha) = 2F^{\mathsf{T}}(F\alpha - y) = 0,$$

откуда следует $F^{\mathsf{T}}F\alpha = F^{\mathsf{T}}y$. Эта система линейных уравнений относительно α называется нормальной системой для задачи наименьших квадратов. Если матрица $F^{\mathsf{T}}F$ размера $n \times n$ невырождена, то решением нормальной системы является вектор

$$\alpha^* = (F^\intercal F)^{-1} F^\intercal y = F^+ y.$$

Матрица $F^+ = (F^\intercal F)^{-1} F^\intercal$ называется *псевдообратной* для прямоугольной матрицы F. Подставляя найденное решение в исходный функционал, получаем

$$Q(\alpha^*) = \|P_F y - y\|^2,$$

где $P_F = FF^+ = F(F^\intercal F)^{-1} F^\intercal$ — *проекционная матрица*.

Решение имеет простую геометрическую интерпретацию. Произведение $P_F y$ есть проекция целевого вектора y на линейную оболочку столбцов матрицы F. Разность $(P_F y - y)$ есть проекция целевого вектора y на ортогональное дополнение этой линейной оболочки. Значение функционала $Q(\alpha^*) = \|P_F y - y\|^2$ есть квадрат длины перпендикуляра, опущенного из y на линейную оболочку. Таким образом, МНК находит кратчайшее расстояние от y до линейной оболочки столбцов F.

Известно большое количество численных методов решения нормальной системы. Наибольшей популярностью пользуются методы, основанные на ортогональных разложениях матрицы *F*. Эти методы эффективны, обладают хорошей численной устойчивостью и позволяют строить различные модификации и обобщения.

Сингулярное разложение

Произвольную$\ell \times n$-матрицу ранга *n*можно представить в виде сингулярного разложения (singularvaluedecomposition, SVD)

$F = VDU^\intercal$,

обладающего рядом замечательных свойств (позже мы докажем Теорему 5.2, из которой эти свойства будут вытекать как следствия):
вектор $F\alpha*$ — МНК-аппроксимацию целевого вектора *y*:

1) $n \times n$-матрица D диагональна, $D = \operatorname{diag}\left(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}\right)$, где $\lambda_1, \ldots, \lambda_n$ — общие ненулевые собственные значения матриц $F^\intercal F$ и $FF^\intercal$.

2) $\ell \times n$-матрица $V = (v_1, \ldots, v_n)$ ортогональна, $V^\intercal V = I_n$, столбцы v_j являются собственными векторами матрицы $FF^\intercal$, соответствующими $\lambda_1, \ldots, \lambda_n$;

3) $n \times n$-матрица $U = (u_1, \ldots, u_n)$ ортогональна, $U^\intercal U = I_n$, столбцы u_j являются собственными векторами матрицы $F^\intercal F$, соответствующими $\lambda_1, \ldots, \lambda_n$;

Имея сингулярное разложение, легко записать псевдообратную матрицу:

$$F^+ = (UDV^\intercal VDU^\intercal)^{-1} UDV^\intercal = UD^{-1}V^\intercal = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j v_j^\intercal;$$

вектор МНК-решения:

$$\alpha^* = F^+ y = UD^{-1}V^\intercal y = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j (v_j^\intercal y); \tag{5.5}$$

вектор $F\alpha^*$ — МНК-аппроксимацию целевого вектора y:

$$F\alpha^* = P_F y = (VDU^\intercal) UD^{-1}V^\intercal y = VV^\intercal y = \sum_{j=1}^{n} v_j (v_j^\intercal y); \tag{5.6}$$

и норму вектора коэффициентов:

$$\|\alpha^*\|^2 = y^\intercal VD^{-1}U^\intercal UD^{-1}V^\intercal y = y^\intercal VD^{-2}V^\intercal y = \sum_{j=1}^{n} \frac{1}{\lambda_j} (v_j^\intercal y)^2. \tag{5.7}$$

Итак, если есть сингулярное разложение, то обращать матрицы уже не нужно. Однако вычисление сингулярного разложения практически столь же трудоёмко, как и обращение. Эффективные численные алгоритмы, вычисляющие SVD, реализованы во многих стандартных математических пакетах.

Проблема мультиколлинеарности

Если ковариационная матрица $\Sigma = F^\intercal F$ имеет неполный ранг, то её обращение невозможно. Тогда приходится отбрасывать линейно зависимые признаки или применять описанные ниже методы — регуляризацию или метод главных компонент. На практике чаще встречается проблема мультиколлинеарности — когда матрица Σ имеет полный ранг, но близка к некоторой матрице неполного ранга. Тогда говорят, что Σ — матрица неполного псевдоранга или что она плохо обусловлена. Геометрически это означает, что объекты выборки сосредоточены вблизи линейного подпространства меньшей размерности $m<n$. Признаком мультиколлинеарности является наличие у матрицы Σ собственных значений, близких к нулю. Число обусловленности матрицы Σ есть

$$\mu(\Sigma) = \|\Sigma\| \|\Sigma^{-1}\| = \frac{\max\limits_{u:\ \|u\|=1} \|\Sigma u\|}{\min\limits_{u:\ \|u\|=1} \|\Sigma u\|} = \frac{\lambda_{\max}}{\lambda_{\min}},$$

где λ_{max} и λ_{min} — максимальное и минимальное собственные значения матрицы Σ, все нормы евклидовы. Матрица считается плохо обусловленной, если $\mu(\Sigma)$ &10^2 ...10^4. Обращение такой матрицы численно неустойчиво. При умножении обратной матрицы на вектор, $z = \Sigma^{-1}u$, относительная погрешность усиливается в $\mu(\Sigma)$ раз:

$$\frac{\|\delta z\|}{\|z\|} \leqslant \mu(\Sigma)\frac{\|\delta u\|}{\|u\|}.$$

Именно это и происходит с МНК-решением в случае плохой обусловленности. В формуле (5.7) близкие к нулю собственные значения оказываются в знаменателе, в результате увеличивается разброс коэффициентов $\alpha*$, появляются большие по абсолютной величине положительные и отрицательные коэффициенты. МНК-решение становится неустойчивым — малые погрешности измерения признаков или ответов у обучающих объектов могут существенно повлиять на вектор решения $\alpha*$, а погрешности измерения признаков у тестового объекта x— на значения функции регрессии $g(x,\alpha*)$. Мультиколлинеарность влечёт не только неустойчивость и переобучение, но и неинтерпретируемость коэффициентов, так как по абсолютной величине коэффициента α_jстановится невозможно судить о степени важности признака f_j.

Гребневая регрессия

Для решения проблемы мультиколлинеарности добавим к функционалу Qрегуляризатор, штрафующий большие значения нормы вектора весов $\|\alpha\|$:

$$Q_\tau(\alpha) = \|F\alpha - y\|^2 + \tau\|\alpha\|^2,$$

где τ— неотрицательный параметр. В случае мультиколлинеарности имеется бесконечно много векторов α, доставляющих функционалу Qзначения, близкие к минимальному. Штрафное слагаемое выполняет рольрегуляризатора, благодаря которому среди них выбирается решение с минимальной нормой. Приравнивая нулю производную $Q_\tau(\alpha)$ по параметру α, находим:

$$\alpha_\tau^* = (F^\intercal F + \tau I_n)^{-1}F^\intercal y.$$

Таким образом, перед обращением матрицы к ней добавляется «гребень» — диагональная матрица τI_n. Отсюда и название метода — гребневая регрессия (ridgeregression). При этом все её собственные значения увеличиваются на τ, а собственные векторы не изменяются. В результате матрица становится хорошо обусловленной, оставаясь в то же время «похожей» на исходную. Аналогичный приём мы уже упоминали в разделе 2.3.3 в связи с обращением ковариационной матрицы Σв линейном дискриминанте Фишера.

Выразим регуляризованное МНК-решение через сингулярное разложение:

$$\alpha_\tau^* = (UD^2U^\intercal + \tau I_n)^{-1}UDV^\intercal y = U(D^2 + \tau I_n)^{-1}DV^\intercal y = \sum_{j=1}^{n}\frac{\sqrt{\lambda_j}}{\lambda_j + \tau}u_j(v_j^\intercal y).$$

Теперь найдём регуляризованную МНК-аппроксимацию целевого вектора y:

$$F\alpha_\tau^* = VDU^\intercal\alpha_\tau^* = V\,\mathrm{diag}\left(\frac{\lambda_j}{\lambda_j + \tau}\right)V^\intercal y = \sum_{j=1}^{n}\frac{\lambda_j}{\lambda_j + \tau}v_j(v_j^\intercal y). \qquad (5.8)$$

Как и прежде в (5.6), МНК-аппроксимация представляется в виде разложения целевого вектора yпо базису собственных векторов матрицы $FF^\intercal$. Только теперь проекции на собственные векторы сокращаются, умножаясь на $\frac{\lambda_j}{\lambda_j+\tau} \in (0,1)$. В сравнении с (5.7) уменьшается и норма вектора коэффициентов:

$$\|\alpha_\tau^*\|^2 = \|D^2(D^2+\tau I_n)^{-1}D^{-1}V^\intercal y\|^2 = \sum_{j=1}^{n}\frac{1}{\lambda_j+\tau}(v_j^\intercal y)^2 < \sum_{j=1}^{n}\frac{1}{\lambda_j}(v_j^\intercal y)^2 = \|\alpha^*\|^2.$$

Отсюда ещё одно название метода — сжатие (shrinkage) или сокращение весов (weightdecay) .

Понятие эффективной размерности. Из формул видно, что по мере увеличения параметра τвектор коэффициентовα_τ^* становится всё более устойчивым и жёстко определённым. Фактически, происходит понижение эффективной размерности решения — это второй смысл термина «сжатие».

$$\operatorname{tr} F(F^\intercal F)^{-1}F^\intercal = \operatorname{tr}(F^\intercal F)^{-1}F^\intercal F = \operatorname{tr} I_n = n.$$

При использовании регуляризации эффективная размерность принимает значение от 0 до n, не обязательно целое, и убывает при возрастании τ:

$$n_{\text{эфф}} = \operatorname{tr} F(F^\intercal F+\tau I_n)^{-1}F^\intercal = \operatorname{tr}\operatorname{diag}\left(\frac{\lambda_j}{\lambda_j+\tau}\right) = \sum_{j=1}^{n}\frac{\lambda_j}{\lambda_j+\tau} < n.$$

Проблема выбора константы регуляризации. При $\tau\to 0$ регуляризованное решение стремится к МНК-решению:$\alpha_\tau^* \to \alpha^*$. При $\tau\to\infty$ чрезмерная регуляризации приводит к вырожденному решению: $\alpha_\tau^* \to 0$. Оба крайних случая нежелательны, поэтому оптимальным является некоторое промежуточное значение $\tau*$. Для его нахождения можно применять скользящий контроль. Зависимость оценки скользящего контроля от параметра τ, как правило, имеет характерный минимум.

Скользящий контроль — вычислительно трудоёмкая процедура. Известна практическая рекомендация брать τ в отрезке $[0.1, 0.4]$, если столбцы матрицы F заранее стандартизованы (центрированы и нормированы). Ещё одна эвристика — выбрать τ так, чтобы число обусловленности приняло заданное не слишком большое значение: $M_0 = \mu(F^\intercal F+\tau I_n) = \frac{\lambda_{\max}+\tau}{\lambda_{\min}+\tau}$, откуда следует рекомендация $\tau^* \approx \lambda_{\max}/M_0$.

Лассо Тибширани

Ещё один метод регуляризации внешне похож на гребневую регрессию, но приводит к качественно иному поведению вектора коэффициентов. Вместо добавления штрафного слагаемого к функционалу качества вводится ограничение-неравенство, запрещающее слишком большие абсолютные значения коэффициентов:

$$\begin{cases} Q(\alpha) = \|F\alpha - y\|^2 \to \min\limits_{\alpha}; \\ \sum\limits_{j=1}^{n}|\alpha_j| \leqslant \varkappa; \end{cases} \tag{5.9}$$

где $\varkappa$ — параметр регуляризации.

При больших значениях к ограничение (5.9) становится строгим неравенством, и решение совпадает с МНК-решением. Чем меньше к, тем больше коэффициентов α_jобнуляются. Происходит отбор (селекция) признаков, поэтому параметр к называют ещё селективностью. Образно говоря, параметр к зажимает вектор коэффициентов, лишая его избыточных степеней свободы. Отсюда и названиеметода — лассо (LASSO, least absolute shrinkage and selection operator) .

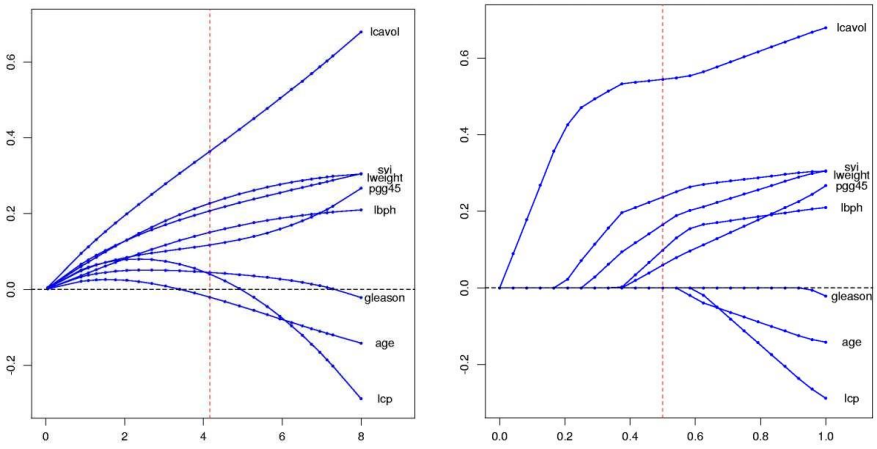
Чтобы понять, почему лассо осуществляет отбор признаков, приведём задачу квадратичного программирования (5.9) к каноническому виду. Заменим каждую переменную α_jразностью двух новых неотрицательных переменных: $\alpha_j = \alpha_j^+ - \alpha_j^-$. Функционал Qостанется квадратичным по новым переменным, ограничение (5.9) примет линейный вид, и добавится $2n$ограничений-неравенств:

$$\sum_{j=1}^{n} \alpha_j^+ + \alpha_j^- \leqslant \varkappa; \quad \alpha_j^+ \geqslant 0; \quad \alpha_j^- \geqslant 0 \qquad .$$

Чем меньше к, тем больше ограничений обращаются в равенства $\alpha_j^+ = \alpha_j^- = 0$, что соответствует обнулению коэффициента α_jи исключению j-го признака.

Сравнение лассо и гребневой регрессии. Оба метода успешно решают проблему мультиколлинеарности. Гребневая регрессия использует все признаки, стараясь «выжать максимум» из имеющейся информации. Лассо производит отбор признаков, что предпочтительнее, если среди признаков есть шумовые или измерения признаков связаны с ощутимыми затратами.

Ослабление регуляризации ведёт к уменьшению ошибки на обучении и увеличению нормы вектора коэффициентов. При этом ошибка на контроле в какой-то момент проходит через минимум, и далее только возрастает — это и есть переобучение.



(а) зависимость $\{\alpha_j\}$ от $\sigma = 1/\tau$ (б) зависимость $\{\alpha_j\}$ отк

Рис. 16. Зависимость коэффициентов линейной модели от параметра $\sigma = 1/\tau$для гребневой регрессии и от параметра к для лассо Тибширани, по реальным данным задачи UCI.cancer [44].

Линейная монотонная регрессия

В некоторых приложениях возникает линейная модель регрессии с неотрицательными коэффициентами. Например, заранее может быть известно, что чем больше значение признака f_j, тем больше должен быть отклик y. При построении линейной композиции алгоритмов регрессии или прогнозирования роль признаков играют базовые алгоритмы (см. главу ??).

Естественно полагать, что если базовые алгоритмы настраиваются на один и тот же целевой вектор y, то они должны учитываться в композиции с положительными весами.

Возникает задача минимизации квадратичного функционала $Q(\alpha,X^\ell)$ с nограничениями типа неравенств:

$$\begin{cases} Q(\alpha) = \|F\alpha - y\|^2 \to \min\limits_{\alpha}; \\ \alpha_j \geqslant 0; \quad j = 1,\ldots,n. \end{cases}$$

Это опять-таки, задача квадратичного программирования с линейными ограничениями. Когда ограничение $\alpha_j>0$ становится активным, то есть обращается в равенство, признак f_j, фактически, исключается из модели регрессии. В линейной композиции это соответствует исключению j-го базового алгоритма из композиции.

Метод главных компонент

Ещё одно решение проблемы мультиколлинеарности заключается в том, чтобы подвергнуть исходные признаки некоторому функциональному преобразованию, гарантировав линейную независимость новых признаков, и, возможно, сократив их количество, то есть уменьшив размерность задачи.

В методе главных компонент (principalcomponentanalysis, PCA) строится минимальное число новых признаков, по которым исходные признаки восстанавливаются линейным преобразованием с минимальными погрешностями. PCAотносится к методам обучения без учителя (unsupervisedlearning), поскольку матрица «объекты–признаки» Fпреобразуется без учёта целевого вектора y.

Важно отметить, что PCA подходит и для регрессии, и для классификации, и для многих других типов задач анализа данных, как вспомогательное преобразование, позволяющее определить эффективную размерность исходных данных.

Постановка задачи. Пусть имеется nчисловых признаков $f_j(x)$, $j= 1,...,n$. Как обычно, будем отождествлять объекты обучающей выборки и их признаковые описания: $x_i \equiv \big(f_1(x_i),\ldots,f_n(x_i)\big),\ i = 1,\ldots,\ell$. Рассмотрим матрицу F, строки которой соответствуют признаковым описаниям обучающих объектов:

$$F_{\ell\times n} = \begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{pmatrix} = \begin{pmatrix} x_1 \\ \ldots \\ x_\ell \end{pmatrix}.$$

Обозначим через$z_i = \big(g_1(x_i),\ldots,g_m(x_i)\big)$ признаковые описания тех же объектов в новом пространстве Z= R^mменьшей размерности, $m<n$:

$$G_{\ell\times m} = \begin{pmatrix} g_1(x_1) & \ldots & g_m(x_1) \\ \ldots & \ldots & \ldots \\ g_1(x_\ell) & \ldots & g_m(x_\ell) \end{pmatrix} = \begin{pmatrix} z_1 \\ \ldots \\ z_\ell \end{pmatrix}.$$

Потребуем, чтобы исходные признаковые описания можно было восстановить по новым описаниям с помощью некоторого линейного преобразования, определяемого матрицей U= $(u_{js})_{n\times m}$:

$$\hat{f}_j(x) = \sum_{s=1}^{m} g_s(x) u_{js}, \quad j = 1, \ldots, n, \quad x \in X,$$

или в векторной записи: $\hat{x} = zU^{\intercal}$. Восстановленное описание $\hat{x}$ не обязано в точности совпадать с исходным описанием x, но их отличие на объектах обучающей выборки должно быть как можно меньше при выбранной размерности m. Будем искать одновременно и матрицу новых признаковых описаний G, и матрицу линейного преобразования U, при которых суммарная невязка восстановленных описаний минимальна:

$$\Delta^2(G,U) = \sum_{i=1}^{\ell} \|\hat{x}_i - x_i\|^2 = \sum_{i=1}^{\ell} \|z_i U^{\intercal} - x_i\|^2 = \|GU^{\intercal} - F\|^2 \to \min_{G,U}, \qquad (5.10)$$

где все нормы евклидовы. Напомним, что $\|A\|^2 = \mathrm{tr}AA^{\intercal} = \mathrm{tr}A^{\intercal}A$, где tr — операция следа матрицы.

Будем предполагать, что матрицы G и U невырождены: $\mathrm{rk}G = \mathrm{rk}U = m$. Иначе существовало бы представление $G^{-}U^{-\intercal} = GU^{\intercal}$ с числом столбцов в матрице G^{-}, меньшим m. Поэтому интересны лишь случаи, когда $m \leqslant \mathrm{rk}F$.

Исчерпывающее решение задачи (5.10) даёт следующая теорема.

Теорема 5.2. Если $m \leqslant \mathrm{rk}F$, то минимум $\Delta^2(G,U)$ достигается, когда столбцы матрицы U есть собственные векторы $F^{\intercal}F$, соответствующие m максимальным собственным значениям. При этом $G = FU$, матрицы U и G ортогональны.

Доказательство. Запишем необходимые условия минимума:

$$\begin{cases} \partial\Delta^2/\partial G = (GU^\intercal - F)U = 0; \\ \partial\Delta^2/\partial U = G^\intercal(GU^\intercal - F) = 0. \end{cases}$$

Поскольку искомые матрицы G и U невырождены, отсюда следует

$$\begin{cases} G = FU(U^\intercal U)^{-1}; \\ U = F^\intercal G(G^\intercal G)^{-1}. \end{cases} \tag{5.11}$$

Функционал $\Delta^2(G,U)$ зависит только от произведения матриц $GU^\intercal$, поэтому решение задачи (5.10) определено с точностью до произвольного невырожденного преобразования R: $GU^\intercal = (GR)(R^{-1}U^\intercal)$. Распорядимся свободой выбора R так, чтобы матрицы $U^\intercal U$ и $G^\intercal G$ оказались диагональными. Покажем, что это всегда возможно.

Пусть $\tilde{G}\tilde{U}^\intercal$ — произвольное решение задачи (5.10).

Матрица $\tilde{U}^\intercal\tilde{U}$ симметричная, невырожденная, положительно определенная, поэтому существует невырожденная матрица $S_{m\times m}$ такая, что $S^{-1}\tilde{U}^\intercal\tilde{U}S^{-1\intercal} = I_m$.

Матрица $S^\intercal\tilde{G}^\intercal\tilde{G}S$ симметричная и невырожденная, поэтому существует ортогональная матрица $T_{m\times m}$ такая, что $T^\intercal(S^\intercal\tilde{G}^\intercal GS)T = \operatorname{diag}(\lambda_1,\ldots,\lambda_m) \equiv \Lambda$ — диагональная матрица. По определению ортогональности $T^\intercal T = I_m$.

Преобразование $R = ST$ невырождено. Положим $G = \tilde{G}R$, $U^\intercal = R^{-1}\tilde{U}^\intercal$. Тогда

$$G^\intercal G = T^\intercal(S^\intercal\tilde{G}^\intercal\tilde{G}S)T = \Lambda;$$
$$U^\intercal U = T^{-1}(S^{-1}\tilde{U}^\intercal\tilde{U}S^{-1\intercal})T^{-1\intercal} = (T^\intercal T)^{-1} = I_m.$$

В силу $GU^\intercal = \tilde{G}\tilde{U}^\intercal$ матрицы G и U являются решением задачи (5.10) и удовлетворяют необходимому условию минимума. Подставим матрицы G и U в (5.11). Благодаря диагональности $G^\intercal G$ и $U^\intercal U$ соотношения существенно упростятся:

$$\begin{cases} G = FU; \\ U\Lambda = F^\intercal G. \end{cases}$$

Подставим первое соотношение во второе, получим $U\Lambda = F^\intercal FU$. Это означает, что столбцы матрицы Uобязаны быть собственными векторами матрицы $F^\intercal F$, а диагональные элементы $\lambda_1,\ldots,\lambda_m$— соответствующими им собственными значениями.

Аналогично, подставив второе соотношение в первое, получим $G\Lambda = FF^\intercal G$, то есть столбцы матрицы Gявляются собственными векторами $FF^\intercal$, соответствующими тем же самым собственным значениям.

Подставляя G и U в функционал $\Delta^2(G,U)$, находим:

$$\begin{aligned} \Delta^2(G,U) &= \|F - GU^\intercal\|^2 = \operatorname{tr}(F^\intercal - UG^\intercal)(F - GU^\intercal) = \operatorname{tr} F^\intercal(F - GU^\intercal) = \\ &= \operatorname{tr} F^\intercal F - \operatorname{tr} F^\intercal GU^\intercal = \|F\|^2 - \operatorname{tr} U\Lambda U^\intercal = \\ &= \|F\|^2 - \operatorname{tr}\Lambda = \sum_{j=1}^{n}\lambda_j - \sum_{j=1}^{m}\lambda_j = \sum_{j=m+1}^{n}\lambda_j, \end{aligned}$$

где $\lambda_1,\ldots,\lambda_n$ — все собственные значения матрицы $F^\intercal F$. Минимум Δ^2 достигается, когда $\lambda_1,\ldots,\lambda_m$ — наибольшие m из n собственных значений. ■

Собственные векторы $u_1,\ldots,u_m$, отвечающие максимальным собственным значениям, называют главными компонентами.

Из Теоремы 5.2 вытекают следующие свойства метода главных компонент.

Связь с сингулярным разложением. Если $m = n$, то $\Delta^2(G, U) = 0$. В этом случае представление $F = GU^\intercal$ является точным и совпадает с сингулярным разложением: $F = GU^\intercal = VDU^\intercal$, если положить $G = VD$ и $\Lambda = D^2$. При этом матрица V ортогональна: $V^\intercal V = I_m$. Остальные свойства сингулярного разложения, перечисленные на стр. 85, непосредственно вытекают из Теоремы 5.2.

Если $m < n$, то представление $F \approx GU^\intercal$ является приближённым. Сингулярное разложение матрицы $GU^\intercal$ получается из сингулярного разложения матрицы F путём отбрасывания (обнуления) $n - m$ минимальных собственных значений.

Преобразование Карунена–Лоэва. Диагональность матрицы $G^\intercal G = \Lambda$ означает, что новые признаки $g_1, \ldots, g_m$ не коррелируют на обучающих объектах. Ортогональное преобразование U называют *декоррелирующим* или преобразованием *Карунена–Лоэва*. Если $m = n$, то прямое и обратное преобразование вычисляются с помощью одной и той же матрицы U: $F = GU^\intercal$ и $G = FU$.

Задача наименьших квадратов в новом признаковом пространстве имеет вид

$$\|G\beta - y\|^2 \to \min_\beta.$$

Поскольку U ортогональна, $G\beta = GU^\intercal U\beta = GU^\intercal\alpha \approx F\alpha$, где $\alpha = U\beta$. Это означает, что задача наименьших квадратов в новом пространстве соответствует замене матрицы F на её приближение $GU^\intercal$ в исходной задаче наименьших квадратов.

Интересно отметить, что новый вектор коэффициентов β связан со старым α тем же линейным преобразованием U: $\beta = U^\intercal U\beta = U^\intercal\alpha$.

В новом пространстве МНК-решение не требует явного обращения матрицы, поскольку $G^\intercal G$ диагональна:

$$\beta^* = \Lambda^{-1}G^\intercal y = D^{-1}V^\intercal y;$$
$$G\beta^* = VD\beta^* = VV^\intercal y.$$

Для вектора $\alpha* = U\beta*$ МНК-решение выглядит так же, как и раньше, с той лишь разницей, что в суммах (5.5)–(5.7) надо взять первые m<= nслагаемых, а оставшиеся $n-m$просто отбросить.

Интересно сравнить метод главных компонент и гребневую регрессию. Оба сводятся к модификации сумм (5.5)–(5.7). Гребневая регрессия сокращает коэффициенты при всех слагаемых, а метод главных компонент обнуляет коэффициенты при последних слагаемых. Возможно, имеют смысл и комбинации этих двух методов.

Эффективная размерность. Главные компоненты содержат основную информацию о матрице F. Число главных компонент mназывают также эффективной размерностью задачи. На практике её определяют следующим образом. Все собственные значения матрицы $F^\intercal F$упорядочиваются по убыванию: λ_1 >... >λ_n>0. Задаётся пороговое значение $\varepsilon\in$ [0,1], достаточно близкое к нулю, и определяется наименьшее целое m, при котором относительная погрешность приближения матрицы Fне превышает ε:

$$E(m) = \frac{\|GU^\intercal - F\|^2}{\|F\|^2} = \frac{\lambda_{m+1} + \cdots + \lambda_n}{\lambda_1 + \cdots + \lambda_n} \leqslant \varepsilon.$$

Величина $E(m)$ показывает, какая доля информации теряется при замене исходных признаковых описаний длины nна более короткие описания длины m. Метод главных

компонент особенно эффективен в тех случаях, когда $E(m)$ оказывается малым уже при малых значениях m.

Если задать число εиз априорных соображений не представляется возможным, прибегают к критерию «крутого обрыва». На графике $E(m)$ отмечается то значение m, при котором происходит резкий скачок: $E(m-1) \gg E(m)$, при условии, что $E(m)$ уже достаточно мало´.

Визуализация многомерных данных. Метод главных компонент часто используется для представления многомерной выборки данных на двумерном графике. Для этого полагают $m=2$ и полученные пары значений $\big(g_1(x_i), g_2(x_i)\big),\ i = 1,\ldots,\ell$, наносят как точки на график. Проекция на главные компоненты является наименее искаженной из всех линейных проекций многомерной выборки на какую-либо пару осей. Как правило, в осях главных компонент удаётся увидеть наиболее существенные особенности исходных данных, даже несмотря на неизбежные искажения. В частности, можно судить о наличии кластерных структур и выбросов. Две оси g_1 и g_2 отражают «две основные тенденции» в данных. Иногда их удаётся интерпретировать, если внимательно изучить, какие точки на графике являются «самыми левыми», «самыми правыми», «самыми верхними» и «самыми нижними». Этот вид анализа не позволяет делать точные количественные выводы и обычно используется с целью понимания данных. Аналогичную роль играют многомерное шкалирование (см. ??) и карты Кохонена (см. 7.2.2).

Нелинейные методы восстановления регрессии

Предположение о том, что модель регрессии линейна по параметрам, удобно для построения численных методов, но не всегда хорошо согласуется со знаниями о предметной области. В этом параграфе рассматриваются случаи, когда модель регрессии нелинейна по параметрам, когда в линейную модель добавляются нелинейные преобразования исходных признаков или целевого признака, а также когда вводится неквадратичная функция потерь.

Общая идея во всех этих случаях одна: нелинейная задача сводится к решению последовательности более простых линейных задач.

Нелинейная модель регрессии

Пусть задана нелинейная модель регрессии $f(x,\alpha)$ и требуется минимизировать функционал качества по вектору параметров $\alpha \in \mathrm{R}^p$:

$$Q(\alpha, X^\ell) = \sum_{i=1}^{\ell} \big(f(x_i,\alpha) - y_i\big)^2 .$$

Для выполнения численной минимизации функционала Qвоспользуемся методом Ньютона–Рафсона. Выберем начальное приближение $\alpha^0 = (\alpha_1^0,\ldots,\alpha_p^0)$ и организуем итерационный процесс

$$\alpha^{t+1} := \alpha^t - h_t\big(Q''(\alpha^t)\big)^{-1} Q'(\alpha^t),$$

где $Q'(\alpha^t)$ — градиент функционала Qв точке α^t, $Q''(\alpha^t)$ — гессиан (матрица вторых производных) функционала Qв точке α^t, h_t— величина шага, который можно регулировать, а в простейшем варианте просто полагать равным единице.

Запишем компоненты градиента:

$$\frac{\partial}{\partial \alpha_j} Q(\alpha) = 2 \sum_{i=1}^{\ell} \big(f(x_i, \alpha) - y_i\big) \frac{\partial f}{\partial \alpha_j}(x_i, \alpha)$$.

Запишем компоненты гессиана:

$$\frac{\partial^2}{\partial \alpha_j \partial \alpha_k} Q(\alpha) = 2 \sum_{i=1}^{\ell} \frac{\partial f}{\partial \alpha_j}(x_i, \alpha) \frac{\partial f}{\partial \alpha_k}(x_i, \alpha) - \underbrace{2 \sum_{i=1}^{\ell} \big(f(x_i, \alpha) - y_i\big) \frac{\partial^2 f}{\partial \alpha_j \partial \alpha_k}(x_i, \alpha)}.$$

при линеаризации полагается равным 0

Поскольку функция *f*задана, градиент и гессиан легко вычисляются численно. Основная сложность метода Ньютона–Рафсона заключается в обращении гессиана на каждой итерации.

Необходимо отметить, что дифференциальное исчисление функций многих переменных — важный раздел анализа, имеющий немало приложений в физике, инженерии и прикладной математике. Существенное количество практических задач формулируется в терминах функций от двух переменных — явном выражении поверхностей в пространстве $\mathbb{R}^3$.

Якобианом векторного поля $f : \mathbb{R}^m \to \mathbb{R}^n, \forall x \in \mathbb{R}^m f(x) = (f_1(x), ..., f_m(x))$, дифференцируемого в точке x и непрерывного в некоторой её окрестности $U(x) \in \mathbb{R}^m$ называют линейный оператор **J**, описывающий наилучшее линейное приближение функции в некоторой окрестности точки xи имеющий матрицу вида:

$$J_f(x) = \begin{Vmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \dots & \frac{\partial f_1}{\partial x_m}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) & \dots & \frac{\partial f_2}{\partial x_m}(x) \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_m}{\partial x_1}(x) & \frac{\partial f_m}{\partial x_2}(x) & \dots & \frac{\partial f_m}{\partial x_m}(x) \end{Vmatrix}$$

— так называемую **матрицу Якоби** (матрица касательного отображения). Для скалярного поля матрица Якоби имеет вид:

$$J_f(x) = \begin{Vmatrix} \frac{\partial f}{\partial x_1}(x) & \frac{\partial f}{\partial x_2}(x) & \dots & \frac{\partial f}{\partial x_m}(x) \end{Vmatrix}$$

Гессианом скалярного поля $f : \mathbb{R}^m \to \mathbb{R}$, дважды дифференцируемого по всем аргументам в точке $x = (x^1, ..., x^m) \in \mathbb{R}^m$, называют симметрическую квадратичную форму $H(x) = \sum_{i=1}^{m} \sum_{j=1}^{m} h_{ij} x_i x_j$, описывающую наилучшее квадратичное приближение функции в некоторой окрестности точки x и имеющую матрицу вида:

$$\mathbf{H}_f(x) = \begin{Vmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_m}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_m}(x) \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x_m \partial x_1}(x) & \frac{\partial^2 f}{\partial x_m \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_m^2}(x) \end{Vmatrix}$$

Для того, чтобы функция $f : U(x_0) \to \mathbb{R}$, дважды дифференцируемая по всем аргументам в точке $x_0 = (x_0^1, \ldots, x_0^m) \in \mathbb{R}^m$, в ней имела экстремум достаточно, чтобы её Гессиан был знакоопределён, причем, положительная определённость влечёт наличие в точке **строгого локального минимума**, отрицательная определённость — **строгого локального максимума**.

Более эффективной с вычислительной точки зрения является следующая модификация этого метода. Если функция *f* достаточно гладкая (дважды непрерывно дифференцируема), то её можно линеаризовать в окрестности текущего значения вектора коэффициентов α^t:

$$f(x_i, \alpha) = f(x_i, \alpha^t) + \sum_{j=1}^{p} \frac{\partial f}{\partial \alpha_j}(x_i, \alpha_j)(\alpha_j - \alpha_j^t).$$

Заменим в гессиане функцию *f* на её линеаризацию. Это всё равно, что положить второе слагаемое в гессиане равным нулю. Тогда не нужно будет вычислять вторые производные $\frac{\partial^2 f}{\partial \alpha_j \partial \alpha_k}(x_i, \alpha)$. Этот метод называют методом Ньютона–Гаусса. В остальном он ничем не отличается от метода Ньютона–Рафсона.

Введём матричные обозначения: $F_t = \left(\frac{\partial f}{\partial \alpha_j}(x_i, \alpha^t)\right)_{i=1,\ell}^{j=1,p}$ — матрица первых производных размера $\ell \times p$ на t-й итерации; $f_t = \left(f(x_i, \alpha^t)\right)_{i=1,\ell}$ — вектор значений аппроксимирующей функции на t-й итерации. Тогда формула t-й итерации метода Ньютона– Гаусса в матричной записи примет вид:

$$\alpha^{t+1} := \alpha^t - h_t \underbrace{(F_t^\intercal F_t)^{-1} F_t^\intercal (f^t - y)}_{\delta}.$$

В правой части записано решение стандартной задачи многомерной линейной регрессии $\|F_t\delta - (f^t - y)\|^2 \to \min$. Таким образом, в методе Ньютона–Гаусса нелинейная регрессия сводится к последовательности линейных регрессионных задач. Скорость сходимости у него практически такая же, как и у метода Ньютона–Рафсона (оба являются методами второго порядка), но вычисления несколько проще и выполняются стандартными методами линейной регрессии.

Нелинейные одномерные преобразования признаков

На практике встречаются ситуации, когда линейная модель регрессии представляется необоснованной, но предложить адекватную нелинейную модель $f(x,\alpha)$

Алгоритм 5.2. Метод настройки с возвращениями (backfitting).

Вход:
F, y — матрица «объекты–признаки» и вектор ответов;
Выход:
$\varphi_j(x)$ — функции преобразования признаков, в общем случае нелинейные.

нулевое приближение:
$\alpha :=$ решение задачи МЛР с признаками $f_j(x)$;
$\varphi_j(x) := \alpha_j f_j(x), \ \ j = 1, \ldots, n;$
повторять
  для $j = 1, \ldots, n$
    $z_i := y_i - \sum\limits_{k=1, k\neq j}^{n} \varphi_k(f_k(x_i)), \ \ i = 1, \ldots, \ell;$
    $\varphi_j := \arg\min\limits_{\varphi} \sum\limits_{i=1}^{\ell} \big(\varphi(f_j(x)) - z_i\big)^2;$
    $Q_j := \sum\limits_{i=1}^{\ell} \big(\varphi_j(f_j(x)) - z_i\big)^2;$
пока значения Q_j не стабилизируются

также не удаётся. Тогда в качестве компромисса строится модель вида

$$f(x, \alpha) = \sum_{j=1}^{n} \varphi_j(f_j(x)),$$

где $\varphi_j : \mathbb{R} \to \mathbb{R}$ — некоторые преобразования исходных признаков, в общем случае нелинейные. Задача состоит в том, чтобы подобрать неизвестные одномерные преобразования φ_j, при которых достигается минимум квадратичного функционала (5.1).

Метод настройки с возвращениями предложен Хасти и Тибширани в 1986 году . Схема реализации показана в Алгоритме 5.2.

Метод основан на итерационном уточнении функций ϕ_j. На первом шаге они полагаются линейными, $\phi_j(x) = \alpha_j f_j(x)$, и неизвестные коэффициенты α_jнастраиваются методами многомерной линейной регрессии. На каждом последующем шаге выбирается одна из функций ϕ_j, все остальные фиксируются, и выбранная функция строится заново. Для этого решается стандартная задача наименьших квадратов

$$Q(\varphi_j, X^\ell) = \sum_{i=1}^{\ell} \Big(\varphi_j(f_j(x_i)) - \underbrace{\big(y_i - \textstyle\sum_{k=1, k\neq j}^{n} \varphi_k(f_k(x_i))\big)}_{z_i = \text{const}(\varphi_j)} \Big)^2 \to \min_{\varphi_j}$$

с обучающей выборкой $Z_j^\ell = \big(f_j(x_i), z_i\big)_{i=1}^{\ell}$. Для решения данной задачи годятся любые одномерные методы: ядерное сглаживание, сплайны, полиномиальная или Фурье-аппроксимация.

Обобщённые линейные модели

Рассмотрим другую ситуацию, когда модель регрессии $f(x,\alpha)$ линейна, но известна нелинейная функция связи $g(f)$ между выходом модели fи целевым признаком y. Задача аппроксимации ставится, исходя из принципа наименьших квадратов:

$$Q(\alpha, X^\ell) = \sum_{i=1}^{\ell} \Big(g\big(\underbrace{\textstyle\sum_{j=1}^{n} \alpha_j f_j(x_i)}_{z_i}\big) - y_i \Big)^2 \to \min_{\alpha \in \mathbb{R}^n},$$

где $g(f)$ — заданная непрерывно дифференцируемая функция.

Допустим, имеется некоторое приближение вектора коэффициентов α. Линеаризуем функцию $g(z)$ в окрестности каждого из ℓ значений z_i:

$$g(z) = g(z_i) + g'(z_i)(z - z_i).$$

Тогда функционал Qаппроксимируется функционалом $\tilde{Q}$, квадратичным по вектору коэффициентов α:

$$\tilde{Q}(\alpha, X^\ell) = \sum_{i=1}^{\ell} \Big(g(z_i) + g'(z_i)\big(\textstyle\sum_{j=1}^{n} \alpha_j f_j(x_i) - z_i\big) - y_i \Big)^2 =$$

$$= \sum_{i=1}^{\ell} \underbrace{\big(g'(z_i)\big)^2}_{w_i} \Big(\textstyle\sum_{j=1}^{n} \alpha_j f_j(x_i) - \underbrace{\big(z_i + \frac{y_i - g(z_i)}{g'(z_i)}\big)}_{\tilde{y}_i} \Big)^2 \to \min_{\alpha \in \mathbb{R}^n}.$$

Линеаризованная задача сводится к стандартной многомерной линейной регрессии с весами объектов w_iи модифицированным целевым признаком $\tilde{y}$. Решение этой задачи принимается за следующее приближение вектора коэффициентов α. Итерации повторяются до тех пор, пока вектор коэффициентов αили значение функционала $Q(\alpha)$ не перестанет существенно изменяться.

Неквадратичные функции потерь

Функция потерь $L(a,y)$ характеризует величину потери от ответа $a \in Y$при точном ответе $y \in Y$. Она задаётся априори, и благодаря ей задача обучения алгоритма $a(x)$ по выборке $X^\ell = (x_i, y_i)_{i=1}^{\ell}$ сводится к минимизации суммарных потерь:

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i), y(x_i)\big) \to \min_{a:\, X \to Y}.$$

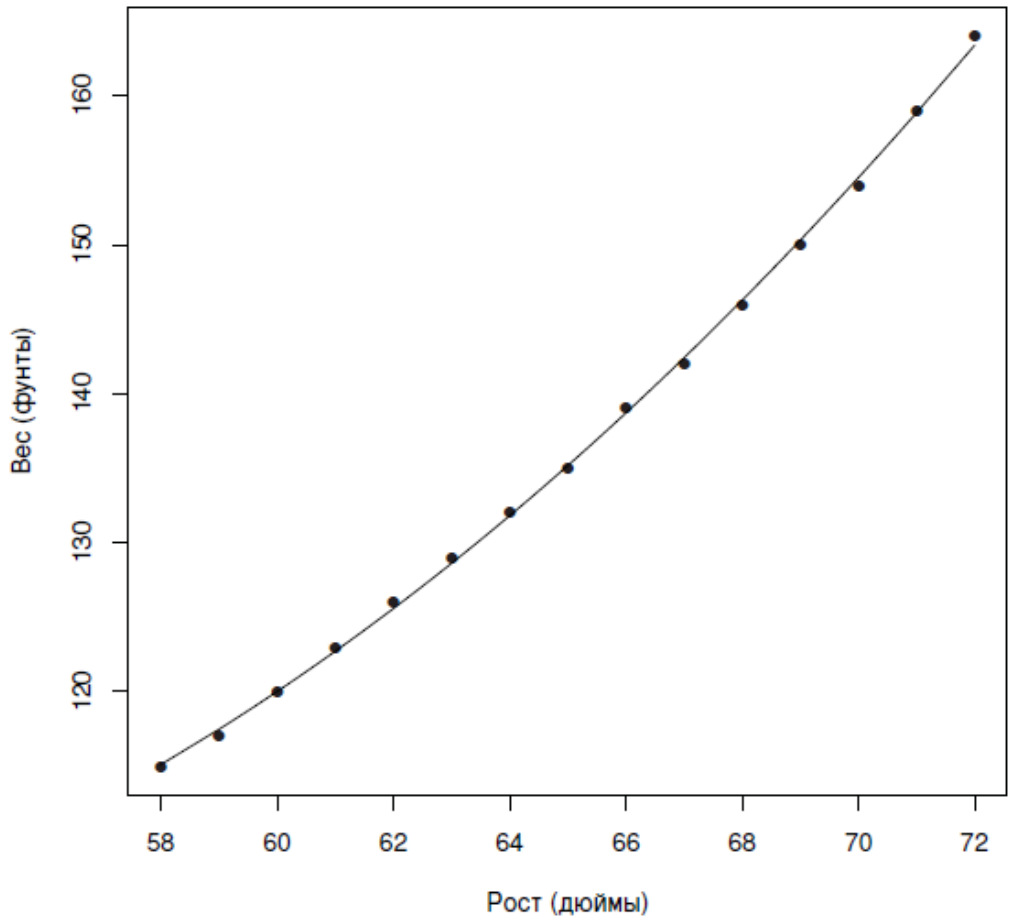


Рис. 8.2. Квадратичная регрессия для предсказаний значений веса по значениям роста

Если функция потерь квадратична, $L(a,y) = (a - y)^2$, то минимизация Qсоответствует методу наименьших квадратов, который был рассмотрен выше. При неквадратичных функциях потерь применяются численные методы оптимизации. Мы не будем подробно останавливаться на методах, а ограничимся перечислением ситуаций, в которых возникают функции потерь, отличные от квадратичных.

Ненормальный шум. Как было показано в Примере 1.4, вид функции потерь связан с априорными предположениями о распределении шума. В частности, квадратичная функция потерь соответствует гауссовскому шуму. Если распределение шума не гауссовское, то функция потерь окажется неквадратичной.

Проблемно-зависимые функции потерь. Во многих прикладных задачах минимизация ошибки предсказания $|a - y|$ или максимизация правдоподобия являются не самыми естественными критериями качества алгоритма.

Пример 5.1. При планировании закупок в розничной сети решается регрессионная задача прогнозирования потребительского спроса. Строится алгоритм $a(x)$, который отвечает на вопрос, сколько единиц данного товара купят в данном магазине в ближайшее время (скажем, в течение следующей недели). Квадрат отклонения $(a - y)^2$ прогноза aот реального спроса

yэкономического смысла не имеет. Гораздо удобнее измерять потери в рублях. Потери от заниженного прогноза $a<y$связаны с недополученной прибылью и прямо пропорциональны величине отклонения: $L(a,y) = = c_1|a-y|$, где c_1 — коэффициент торговой наценки. Потери от завышенного прогноза $a>y$связаны с замораживанием средств, затовариванием склада, а в худшем случае — с истечением срока годности и списанием товара. В первом приближении эти потери также прямо пропорциональны отклонению, но с другим коэффициентом: $L(a,y) = c_2|a- y|$. Коэффициенты c_1 и c_2 известны для каждого магазина и каждого товара. Таким образом, в данной задаче более обоснованной оказывается не квадратичная, а кусочно-линейная несимметричная функция потерь.

Пример 5.2. При создании автоматических систем биржевой торговли строится алгоритм $a(x)$, прогнозирующий в момент времени x_iцену финансового инструмента на следующий момент x_{i+1}. В данном случае квадрат отклонения $(a-y)^2$ особого интереса не представляет. Экономический смысл имеет величина прибыли, которую можно получить, играя на бирже с применением алгоритма $a(x)$. Допустим, мы покупаем 1 акцию, если алгоритм предсказывает повышение, и продаём 1 акцию, если он предсказывает понижение. В следующий момент времени совершаем противоположную операцию (на языке трейдеров «закрываем позицию»), и тут же принимаем следующее решение согласно алгоритму $a(x)$, и так далее. Суммарная прибыль, заработанная в течение ℓ последовательных моментов времени $x_1,\ldots,x_\ell$, равна

$$Q(a) = \sum_{i=1}^{\ell-1} \operatorname{sign}\big(a(x_i) - y(x_i)\big)\big(y(x_{i+1}) - y(x_i)\big).$$

Обучение алгоритма aсводится к максимизации функционала $Q(a)$ на обучающей последовательности цен $y(x_1),\ldots,y(x_{\ell+1})$. Разности цен $w_i = |y(x_{i+1}) - y(x_i)|$ играют роль весов объектов — чем больше величина скачка w_i, тем важнее правильно спрогнозировать направление скачка. Итак, в данной задаче содержательно обоснованной оказалась взвешенная кусочно-постоянная функция потерь.

Робастная регрессия. Чтобы функционал $Q(a, X^\ell)$ был нечувствителен к выбросам, вводится ограниченная сверху функция потерь, например, функция Мешалкина $\mathscr{L}(a,y) = 1 - \exp\big(-\frac{1}{2\sigma}(a-y)^2\big)$, где σ — параметр, равный дисперсии «обычного» шума, не связанного с большими выбросами. Задача минимизации функционала $Q(a, X^\ell)$ с такой функцией потерь уже не может быть решена средствами линейной алгебры; приходится применять численные методы оптимизации, например, метод сопряжённых градиентов.

Логистическая регрессия и итерационный взвешенный МНК

Напомним, что неквадратичная функция потерь используется также в логистической регрессии. Там минимизируемый функционал имеет вид

$$Q(w) = \sum_{i=1}^{\ell} \ln\Big(1 + \exp\big(-w^\intercal x_i y_i\big)\Big) = -\sum_{i=1}^{\ell} \ln \sigma\big(w^\intercal x_i y_i\big) \to \min_w,$$

где $\sigma(z) = (1 + e^{-z})^{-1}$ — сигмоидная функция.

Стандартная техника настройки параметров wзаключается в применении метода Ньютона-Рафсона для минимизации нелинейного функционала $Q(w)$. В качестве нулевого приближения можно взять «наивное» решение задачи классификации как задачи многомерной линейной регрессии, в которой ответы принимают только два значения, $y_i \in \{-1,+1\}$. Затем

начинается итерационный процесс, на t-м шаге которого уточняется вектор коэффициентов w^{t+1}:

$$w^{t+1} := w^t - h_t\big(Q''(w^t)\big)^{-1}Q'(w^t),$$

где $Q'(w^t)$ — вектор первых производных (градиент) функционала $Q(w)$ в точке w^t, $Q''(w^t)$ — матрица вторых производных (гессиан) функционала $Q(w)$ в точке w^t, h_t — величина шага, который можно положить равным 1, но более тщательный его подбор способен увеличить скорость сходимости.

Найдём выражения для градиента и гессиана.

Обозначим $\sigma_i = \sigma(y_i w^\intercal x_i)$ и заметим, что производная логистической функции есть $\sigma'(z) = \sigma(z)(1-\sigma(z))$.

Элементы градиента (вектора первых производных) функционала $Q(w)$:

$$\frac{\partial Q(w)}{\partial w_j} = -\sum_{i=1}^{\ell}(1-\sigma_i)y_i f_j(x_i), \quad j = 1,\dots,n.$$

Элементы гессиана (матрицы вторых производных) функционала $Q(w)$:

$$\frac{\partial^2 Q(w)}{\partial w_j \partial w_k} = -\frac{\partial}{\partial w_k}\sum_{i=1}^{\ell}(1-\sigma_i)y_i f_j(x_i) =$$

$$= \sum_{i=1}^{\ell}(1-\sigma_i)\sigma_i f_j(x_i) f_k(x_i), \quad j = 1,\dots,n, \quad k = 1,\dots,n.$$

Введём матричные обозначения:

$F_{\ell\times n} = \big(f_j(x_i)\big)$ — матрица признаковых описаний объектов;

$\Gamma_{\ell\times\ell} = \operatorname{diag}\big(\sqrt{(1-\sigma_i)\sigma_i}\,\big)$ — диагональная матрица весов объектов;

$\tilde{F} = \Gamma F$ — взвешенная матрица признаковых описаний объектов;

$\tilde{y}_i = y_i\sqrt{(1-\sigma_i)/\sigma_i}$, $\tilde{y} = (\tilde{y}_i)_{i=1}^{\ell}$ — взвешенный вектор ответов.

В этих обозначениях произведение матрицы, обратной к гессиану, на вектор градиента принимает следующий вид:

$$\big(Q''(w)\big)^{-1}Q'(w) = -(F^\intercal\Gamma^2 F)^{-1}F^\intercal\Gamma\tilde{y} = -(\tilde{F}^\intercal\tilde{F})^{-1}\tilde{F}^\intercal\tilde{y} = -\tilde{F}^+\tilde{y}.$$

Алгоритм 5.3. IRLS — итерационный взвешенный метод наименьших квадратов

Вход:
F, y — матрица «объекты–признаки» и вектор ответов;
Выход:
w — вектор коэффициентов линейной комбинации.

нулевое приближение — обычный МНК:
$w := (F^\intercal F)^{-1} F^\intercal y$;
для $t := 1, 2, 3, \ldots$
  $z := Fw$;
  $\gamma_i := \sqrt{(1-\sigma(z_i))\sigma(z_i)}$ для всех $i = 1, \ldots, \ell$;
  $\tilde{F} := \operatorname{diag}(\gamma_1, \ldots, \gamma_\ell) F$;
  $\tilde{y}_i := y_i \sqrt{(1-\sigma(z_i))/\sigma(z_i)}$ для всех $i = 1, \ldots, \ell$;
  выбрать градиентный шаг h_t;
  $w := w + h_t (\tilde{F}^\intercal \tilde{F})^{-1} \tilde{F}^\intercal \tilde{y}$;
  если $\sigma(z_i)$ мало изменились относительно предыдущей итерации **то**
    прервать итерации, выйти из цикла;
конец цикла по t.

Полученное выражение совпадает с решением задачи наименьших квадратов для многомерной линейной регрессии со взвешенными объектами и модифицированными ответами:

$$Q(w) = \|\tilde{F}w - \tilde{y}\|^2 = \sum_{i=1}^{\ell} \underbrace{(1-\sigma_i)\sigma_i}_{\gamma_i} \Big(w^\intercal x - \underbrace{y_i \sqrt{(1-\sigma_i)/\sigma_i}}_{\tilde{y}_i} \Big)^2 \to \min_w .$$

Таким образом, решение задачи классификации сводится к последовательности регрессионных задач, для каждой из которых веса объектов и ответы пересчитываются заново. Отсюда и название — метод наименьших квадратов с итерационнымперевзвешиванием (iterativelyreweightedleastsquares, IRLS)

Понять смысл этого пересчёта совсем нетрудно. Во-первых, заметим, что величина σ_iравна вероятности правильного ответа алгоритма w^tна объекте x_i. Поэтому вес γ_iмаксимален для пограничных объектов, у которых эта вероятность близка к $\frac{1}{2}$. Увеличение точности настройки на этих объектах способствует уменьшению неопределённости классификации. Во-вторых, по мере увеличения вероятности ошибки алгоритма w^tна объекте x_iмодифицированный ответ $\tilde{y}_i$возрастает по модулю. Это приводит к повышению точности настройки алгоритма w^{t+1} на тех объектах, которые оказались «наиболее трудными» для алгоритма w^tна предыдущей итерации.

Метод опорных векторов в задачах регрессии

Мы уже рассматривали задачи многомерной линейной регрессии, предполагая, что $X = \mathbb{R}^n$, $Y = \mathbb{R}$, алгоритм имеет вид $a(x) = (w, x_i) - w_0$ $a(x) = \langle w, x \rangle - w_0$, , и для настройки параметров $w \in \mathbb{R}^n$и $w_0 \in \mathbb{R}$ минимизируется квадратичный функционал. В случае гребневой регрессии вводится ещё и штрафное слагаемое,

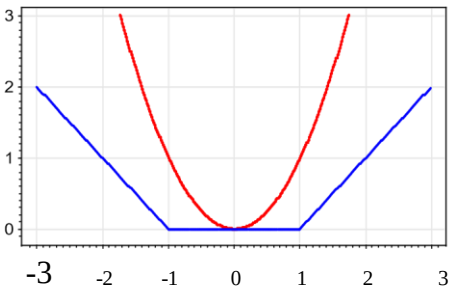


Рис. 17. Функции потерь в задачах регрессии: кусочно-линейная $|z|_\varepsilon$ при $\varepsilon = 1$ и квадратичная z^2.

предотвращающее бесконтрольное увеличение коэффициентов w:

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} (\langle w, x_i \rangle - w_0 - y_i)^2 + \tau \|w\|^2 \to \min_{w, w_0},$$

где τ — параметр регуляризации. Выбор именно квадратичной функции потерь обусловлен удобством решения задачи наименьших квадратов.

Однако в некоторых случаях более естественно использовать кусочно-линейную функцию ε-чувствительности, $|z|_\varepsilon = (|z| - \varepsilon)_+$ показанную на Рис 17, которая не считает за ошибки отклонения $a(x_i)$ от y_i, меньшие ε. Предполагается, что значение параметра ε задаёт эксперт, исходя из априорных соображений.

С этой функцией потерь функционал принимает вид

$$Q_\varepsilon(a, X^\ell) = \sum_{i=1}^{\ell} |\langle w, x_i \rangle - w_0 - y_i|_\varepsilon + \tau \langle w, w \rangle^2 \to \min_{w, w_0}. \qquad (5.12)$$

Легко обнаруживается сходство данной задачи с задачей классификации (4.18). Покажем, что минимизация (5.12) эквивалентна некоторой задаче квадратичного программирования с линейными ограничениями типа неравенств. При этом также возникает двойственная задача, зависящая только от двойственных переменных; также достаточно оставить в выборке только опорные объекты; также решение выражается через скалярные произведения объектов, а не сами объекты; и также можно использовать ядра. Иными словами, SVM-регрессия отличается от SVMклассификации только в технических деталях, основные идеи остаются теми же.

Положим $C = \frac{1}{2\tau}$. Введём дополнительные переменные ξ_i^+ и ξ_i^-, значения которых равны потере при завышенном и заниженном ответе $a(x_i)$ соответственно:

$$\xi_i^+ = (a(x_i) - y_i - \varepsilon)_+, \qquad \xi_i^- = (-a(x_i) + y_i - \varepsilon)_+, \qquad i = 1, \ldots, \ell.$$

Тогда задача минимизации (5.12) может быть переписана в эквивалентной форме как задача квадратичного программирования с линейными ограничениями-неравенствами относительно переменных w_i, w_0, ξ_i^+ и ξ_i^-:

$$\begin{cases} \frac{1}{2} \langle w, w \rangle + C \sum_{i=1}^{\ell} (\xi_i^+ + \xi_i^-) \to \min_{w, w_0, \xi^+, \xi^-} \\ y_i - \varepsilon - \xi_i^- \leqslant \langle w, x_i \rangle - w_0 \leqslant y_i + \varepsilon + \xi_i^+, \quad i = 1, \ldots, \ell \\ \xi_i^- \geqslant 0, \quad \xi_i^+ \geqslant 0, \quad i = 1, \ldots, \ell. \end{cases} \qquad (5.13)$$

;

Как и в предыдущих случаях, лагранжиан этой задачи выражается через двойственные переменные λ^+_i, λ^-_i, $i = 1,\ldots,\ell$, а скалярные произведения $\langle x_i, x_j\rangle$ можно заменить ядром $K(x_i,x_j)$. Опуская выкладки, представим результат:

$$\begin{cases} \mathscr{L}(\lambda^+,\lambda^-) = -\varepsilon\sum\limits_{i=1}^{\ell}(\lambda_i^- + \lambda_i^+) + \sum\limits_{i=1}^{\ell}(\lambda_i^- - \lambda_i^+)y_i - \\ \qquad\qquad - \frac{1}{2}\sum\limits_{i,j=1}^{\ell}(\lambda_i^- - \lambda_i^+)(\lambda_j^- - \lambda_j^+)K(x_i,x_j) \;\to\; \max\limits_{\lambda^+,\lambda^-}; \\ 0 \leqslant \lambda_i^+ \leqslant C, \quad 0 \leqslant \lambda_i^- \leqslant C, \quad i = 1,\ldots,\ell; \\ \sum\limits_{i=1}^{\ell}(\lambda_i^- + \lambda_i^+) = 0. \end{cases}$$

Все объекты x_i, $i = 1,\ldots,\ell$ делятся на следующие пять типов:

1. $|a(x_i) - y_i| < \varepsilon$; $\lambda_i^+ = \lambda_i^- = \xi_i^+ = \xi_i^- = 0$.

Ответ алгоритма $a(x_i)$ находится внутри отрезка $[y_i - \varepsilon, y_i + \varepsilon]$ и считается верным. Объект x_i не является опорным — вектор весов w не изменился бы, если бы этого объекта изначально не было в выборке.

2. $a(x_i) = y_i + \varepsilon$; $0 < \lambda_i^+ < C$; $\lambda_i^- = 0$; $\xi_i^+ = \xi_i^- = 0$.
3. $a(x_i) = y_i - \varepsilon$; $0 < \lambda_i^- < C$; $\lambda_i^+ = 0$; $\xi_i^+ = \xi_i^- = 0$.
4. $a(x_i) > y_i + \varepsilon$; $\lambda_i^+ = C$; $\lambda_i^- = 0$; $\xi_i^+ = a(x_i) - y_i - \varepsilon > 0$; $\xi_i^- = 0$.
5. $a(x_i) < y_i - \varepsilon$; $\lambda_i^- = C$; $\lambda_i^+ = 0$; $\xi_i^- = y_i - a(x_i) - \varepsilon > 0$; $\xi_i^+ = 0$.

Объекты типов 2–5 являются опорными и учитываются при определении вектора весов. При этом только на объектах типов 4 и 5 возникает ненулевая ошибка.

Уравнение регрессии также выражается через двойственные переменные:

$$a(x) = \sum_{i=1}^{\ell}(\lambda_i^- - \lambda_i^+)K(x_i,x) - w_0;$$

где параметр w_0 определяется из ограничений-неравенств, которые становятся равенствами на опорных объектах типа 2 и 3:

$$\langle w, x_i\rangle - w_0 = \begin{cases} y_i + \varepsilon, & \text{если } x_i \text{ — объект типа 2;} \\ y_i - \varepsilon, & \text{если } x_i \text{ — объект типа 3.} \end{cases}$$

Как и раньше, чтобы избежать численной неустойчивости, имеет смысл взять медиану множества значений w_0, вычисленных по всем опорным векторам.

В этом методе есть два управляющих параметра. Параметр точности ε задаётся из априорных соображений. Параметр регуляризации C подбирается, как правило, по скользящему контролю, что является вычислительно трудоёмкой процедурой.

Выбор «лучшей» регрессионной модели

Составляя уравнение регрессии, мы в неявном виде сталкиваемся с выбором из большого числа возможных моделей. Следует ли включить все исследуемые переменные или удалить те, которые не вносят значительного вклада в предсказание значений зависимой переменной? Нужно ли добавлять полиномиальные члены и/или эффекты взаимодействия, чтобы улучшить соответствие модели данным? Выбор окончательной регрессионной модели всегда подразумевает компромисс между точностью предсказания и экономностью. При прочих равных условиях из двух моделей с одинаковой предсказательной силой можно отдать предпочтение наиболее простой. Слово «лучшей» взято в кавычки, поскольку не существует единственного критерия, который можно использовать при совершении выбора. Окончательное решение основывается на мнении исследователя.

Две вложенные модели можно сравнить по степени соответствия данным при помощи программных средств статистического моделирования (язык R функция anova()). Вложенная модель (nestedmodel) – это модель, все члены которой входят в другую модель. Можно проверить, будет ли модель без выбранных переменных предсказывать значения зависимой переменной так же хорошо, как и модель, в которую они включены.

Если результат теста незначим (p = .994), мы заключаем, что выбранные переменные не увеличивают предсказательную силу модели, так что мы правильно решили исключить их. Информационный критерий Акаике (AkaikeInformationCriterion, AIC) – это другой способ сравнения моделей. При расчете этого критерия учитывается статистическое соответствие модели данным и число необходимых для достижения этого соответствия параметров. Предпочтение нужно отдавать моделям с *меньшими* значениями AIC, указывающими на хорошее соответствие данным при использовании меньшего числа параметров. Этот критерий вычисляется при помощи функции AIC()

Выбор переменных

Существуют два распространенных способа формировать окончательный набор независимых переменных из большего числа имеющихся переменных – это пошаговый метод и регрессия по всем подмножествам.

Пошаговая регрессия

При пошаговом выборе переменные добавляются в модель или удаляются из нее по одной, пока не будет достигнуто заданное значение критерия для остановки процесса. Например, при методе *пошагового включения* (forwardstepwise) переменные по одной добавляются в модель, пока добавление новых переменных не перестанет ее улучшать. При *пошаговом исключении* (backwardstepwise) вы начинаете с модели, включающей все независимые переменные, а потом удаляете их по одной до тех пор, пока модель не начнет ухудшаться. При *комбинированном методе* (stepwisestepwise) совмещены оба упомянутых подхода. Переменные добавляются по одной, однако на каждом шаге происходит переоценка модели, и те переменные, которые не вносят значительного вклада, удаляются. Независимая переменная может быть включена в модель и удалена из нее несколько раз, пока не будет достигнуто окончательное решение.

Результат применения метода пошаговой регрессии зависит от критериев включения или удаления переменных. При помощи функции stepAIC() из пакета MASS можно провести все три типа пошаговой регрессии с использованием точного критерия AIC. В следующем программном коде метод регрессии с пошаговым исключением применен для решения задачи по множественной регрессии.

Мы начинаем с модели, включающей все четыре независимые переменные. В столбце AIC приведено значение одноименного критерия для модели, из которой удалена указанная в соответствующей строке переменная. Значение AIC для строки <none> (никакой) – это значение критерия для модели, из которой не удалено никаких переменных. На первом шаге удалена переменная Frost, что привело к уменьшению AIC с 97.75 до 95.75. На втором шаге удалена переменная Income, при этом значение AIC снизилось до 93.76. Удаление остальных переменных увеличивает значение критерия, поэтому процесс остановлен.

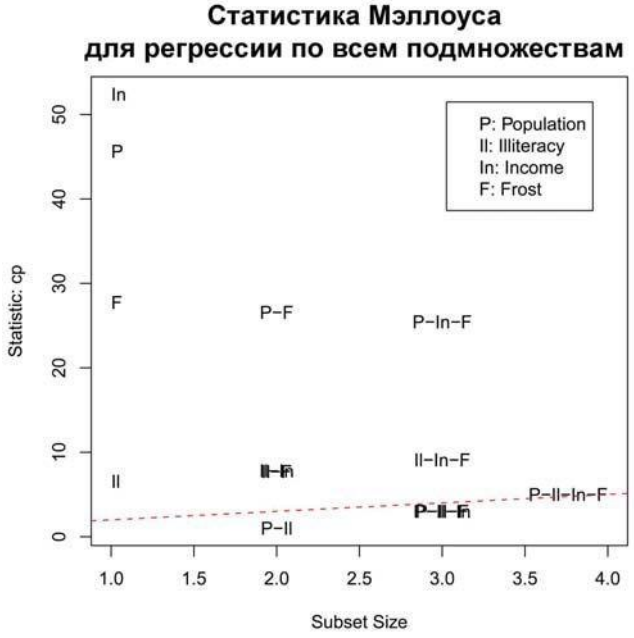
Пошаговая регрессия – спорный подход. Хотя с его помощью можно найти хорошую модель, нет гарантии, что она будет лучшей, поскольку не рассмотрены все возможные модели. Попытка обойти это ограничение делается при использовании *регрессии по всем подмножествам*.

Регрессия по всем подмножествам

В ходе регрессии по всем подмножествам исследуются все возможные модели. Вы можете просмотреть все полученные результаты или вывести на экран определенное число лучших моделей для каждого подмножества (одна независимая переменная, две и т. д.). Например, при значении параметра nbest=2 выводятся на экран две лучшие модели для одной независимой переменной, потом две лучшие модели для двух независимых переменных, затем – для трех, заканчивая двумя лучшими моделями, в которые входят все независимые переменные.

Регрессия по всем подмножествам проводится при помощи функции regsubsets() из пакета leaps. В качестве критерия «лучшей» модели можно выбрать коэффициент детерминации, скорректированный коэффициент детерминации или Ср-статистику Мэллоуса (MallowsCpstatistic).

Как вы уже знаете, коэффициент детерминации – это доля дисперсии зависимой переменной, объясненная независимыми переменными. Скорректированный коэффициент детерминации учитывает число параметров модели. Дело в том, что коэффициент детерминации всегда увеличивается при добавлении независимых переменных. Когда число независимых переменных достаточно велико (по сравнению с объемом выборки), соответствие модели данным может быть переоценено. Скорректированный коэффициент детерминации создан в попытке дать более устойчивую оценку коэффициента детерминации для генеральной совокупности. Статистика Мэллоуса также используется в качестве критерия «лучшей» модели. Считается, что для хорошей модели эта статистика должна принимать значения, близкие к числу параметров модели (включая свободный член).



Регрессионный анализ – это название, под которым скрываются разнообразные статистические методологии. Это в значительной степени интерактивный подход, который включает подбор моделей, проверку выполнения допущений, лежащих в их основе, модификацию и данных, и моделей, а также повторный подбор моделей для достижения окончательного результата.

Список литературы

[1] Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: классификация и снижение размерности. — М.: Финансы и статистика, 1989.

[2] Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: исследование зависимостей. — М.: Финансы и статистика, 1985.

[3] Белецкий Н. Г. Применение комитетов для многоклассовой классификации // Численный анализ решения задач линейного и выпуклого программирования. — Свердловск, 1983.— С. 156–162.

[4] Вайнцвайг М. Н. Алгоритм обучения распознаванию образов «кора» // Алгоритмы обучения распознаванию образов / Под ред. В. Н. Вапник. — М.: Советское радио, 1973.— С. 110–116.

[5] Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы. — М.: Физматлит, 2006.— 320 с.

[6] Грэхем Р., Кнут Д., Паташник О. Конкретная математика.— М.: Мир, 1998.

[7] Fu¨rnkranz J., Flach P. A. Roc ‘n’ rule learning-towards a better understanding of covering algorithms // Machine Learning.— 2005.— Vol. 58, no. 1. — Pp. 39–77. http://dblp.uni-trier.de/db/journals/ml/ml58.html#FurnkranzF05.

[8] Hidber C. Online association rule mining // SIGMOD Conf. — 1999.— Pp. 145–156. http://citeseer.ist.psu.edu/hidber98online.html.

[9] Hipp J., Gu¨ntzer U., Nakhaeizadeh G. Algorithms for association rule mining — a general survey and comparison // SIGKDD Explorations.— 2000.— Vol. 2, no. 1.— Pp. 58–64.