Лекции по алгоритмам восстановления регрессии

К. В. Воронцов

21 декабря 2007 г.

Материал находится в стадии разработки, может содержать ошибки и неточности. Автор будет благодарен за любые замечания и предложения, направленные по адресу voron@ccas.ru. Перепечатка любых фрагментов данного материала без согласия автора является плагиатом.

Содержание

1 Восстановление 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 $y^*\colon X \to Y$, 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 $a\colon X \to Y$, аппроксимирующий целевую зависимость y^*.

§1.1 Метод наименьших квадратов

Пусть модель алгоритмов задана в виде параметрического семейства функций $f(x, \alpha)$, где $\alpha \in \mathbb{R}^p$ — вектор параметров модели.

Определим функционал качества аппроксимации целевой зависимости на выборке X^ℓ как сумму квадратов ошибок:

$$Q(\alpha, X^\ell) = \sum_{i=1}^{\ell} w_i \big(f(x_i, \alpha) - y_i\big)^2, \tag{1.1}$$

где w_i — вес, выражающий степень важности i-го объекта. Этот функционал называют также *остаточной суммой квадратов* (residual sum of squares, RSS).

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:

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^p} Q(\alpha, X^\ell). \tag{1.2}$$

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

$$\frac{\partial Q}{\partial \alpha}(\alpha, X^\ell) = 2 \sum_{i=1}^{\ell} w_i \big(f(x_i, \alpha) - y_i\big) \frac{\partial f}{\partial \alpha}(x_i, \alpha) = 0. \tag{1.3}$$

Численное решение этой системы принимается за искомый вектор параметров α^*. Метод наименьших квадратов широко используется благодаря интуитивной ясности и удобству реализации. Кроме того, при определённых дополнительных предположениях МНК эквивалентен методу максимума правдоподобия (ММП).

Теорема 1.1. *Пусть целевая зависимость y^* описывается вероятностной моделью регрессии с гауссовским шумом, то есть при некотором векторе параметров α*

$$y^*(x_i) = f(x_i, \alpha) + \varepsilon_i, \quad i = 1, \ldots, \ell,$$

где ε_i — независимые нормальные случайные величины с нулевым математическим ожиданием и дисперсией σ_i^2. Тогда решения МНК и ММП, совпадают, причём веса объектов обратно пропорциональны дисперсии шума, $w_i = \sigma_i^{-2}$.

Доказательство.

Запишем функцию правдоподобия для нормального случайного вектора $(\varepsilon_1, \ldots, \varepsilon_\ell)$ с независимыми компонентами:

$$L(\varepsilon_1, \ldots, \varepsilon_\ell | \alpha, X^\ell) = \prod_{i=1}^{\ell} \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_i^2} \big(f(x_i, \alpha) - y_i\big)^2 \right).$$

Запишем логарифм функции правдоподобия:

$$\ln L(\varepsilon_1, \ldots, \varepsilon_\ell | \alpha, X^\ell) = \text{const}(\alpha) - \frac{1}{2} \sum_{i=1}^{\ell} \frac{1}{\sigma_i^2} \big(f(x_i, \alpha) - y_i\big)^2.$$

Максимизация $\ln L(\alpha, X^\ell)$ по α эквивалентна минимизации среднего квадрата ошибки $Q(\alpha, X^\ell)$ с весами $w_i = \sigma_i^{-2}$. ■

Теорема 1.1 проясняет содержательный смысл весов объектов. Чем выше дисперсия σ_i, тем менее точной может быть настройка на i-ом объекте. Если дисперсия шума одинакова для всех объектов, то веса можно положить равными 1.

В Теореме 1.1 предполагается, что модель $f(x, \alpha)$ лишена систематических погрешностей и описывает искомую зависимость с точностью до некоррелированного гауссовского шума. Нарушения этих условий могут приводить к существенному изменению функционала Q, и применение МНК для его минимизации уже будет некорректно. Например, предположение, что шум подчиняется распределению Лапласа приводит к *методу наименьших модулей*:

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

Предположение, что шум представляется смесью двух распределений, одно из которых нормальное, а второе имеет большую дисперсию и низкую априорную вероятность, приводит к широкому классу *робастных методов*, устойчивых к наличию *редких выбросов* — резко выделяющихся ошибочных значений y_i.

§1.2 Непараметрическая регрессия: ядерное сглаживание

Непараметрическое восстановление регрессии основано на той же идее, что и непараметрическое восстановление плотности распределения, рассмотренное в **??**. Значение $a(x)$ вычисляется для каждого объекта x по нескольким ближайшим к нему объектам обучающей выборки. Чтобы можно было говорить о «близости» объектов, на множестве X должна быть задана функция расстояния $\rho(x, x')$.

1.2.1 Метод наименьших квадратов и формула Надарая–Ватсона

Будем рассматривать случай $Y = \mathbb{R}$.

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

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

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

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

$$a_h(x; X^\ell) = \frac{\sum_{i=1}^{\ell} y_i w_i(x)}{\sum_{i=1}^{\ell} w_i(x)} = \frac{\sum_{i=1}^{\ell} y_i K\left(\frac{\rho(x,x_i)}{h}\right)}{\sum_{i=1}^{\ell} K\left(\frac{\rho(x,x_i)}{h}\right)}. \tag{1.4}$$

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

В одномерном случае $X = \mathbb{R}^1$ метрика задаётся как $\rho(x, x_i) = |x - x_i|$. При этом строгим обоснованием формулы (1.4) служит следующая теорема, аналогичная Теореме **??** о непараметрическом восстановлении плотности.

Теорема 1.2 ([2]). *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.

1.2.2 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 K и ширины окна h.

Выбор ядра K мало 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. **??** и в Таблице **??**. Для ядерного сглаживания чаще всего берут гауссовское ядро $K_G(r) = \exp\left(-\frac{1}{2}r^2\right)$ или квартическое $K_Q(r) = (1 - r^2)^2\big[|r| < 1\big]$.

Если ядро $K(r)$ финитно, то есть $K(r) = 0$ при $r \geqslant 1$, то ненулевые веса получат только те объекты x_i, для которых $\rho(x, x_i) < h$. Тогда в формуле (1.4) достаточно суммировать только по ближайшим соседям объекта x. В одномерном случае

$X = \mathbb{R}^1$ 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 h решающим 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, Рис. **??**. 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 $w_i(x) = K\big(\frac{\rho(x,x_i)}{h(x)}\big)$.

Самый простой способ — взять в качестве ширины окна $h(x)$ расстояние от объекта x до его $k+1$-го соседа: $h_k(x) = \rho(x, x_{k+1,x})$. Недостаток этого способа в том, что функция $h_k(x)$ является кусочно-непрерывной, следовательно, функции $w_i(x)$ и $a_{h_k}(x)$ в общем случае имеют разрывные первые производные, причём независимо от степени гладкости ядра.

Оптимизация ширины окна. Чтобы оценить при данном h или k точность 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 h или по числу соседей k.

1.2.3 Проблема выбросов: робастная непараметрическая регрессия

Оценка Надарайя–Ватсона крайне чувствительна к большим одиночным выбросам, что показано на Рис. **??**. На практике легко идентифицируются только грубые ошибки, возникающие, например, в результате сбоя оборудования или невнимательности персонала при подготовке данных. В общем случае можно лишь утверждать, что чем больше величина ошибки

$$\varepsilon_i = \big|a_h\big(x_i; X^\ell \backslash \{x_i\}\big) - y_i\big|,$$

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

Алгоритм 1.1. 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:
$$a_i := a_h\big(x_i; X^\ell \backslash \{x_i\}\big) = \frac{\sum_{j=1,\, j\neq i}^{\ell} y_j \gamma_j K\big(\frac{\rho(x_i,x_j)}{h(x_i)}\big)}{\sum_{j=1,\, j\neq i}^{\ell} \gamma_j K\big(\frac{\rho(x_i,x_j)}{h(x_i)}\big)}, \quad i = 1, \ldots, \ell$$
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;

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 1.1. 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* (locally weighted scatter plot smoothing, LOWESS) [3].

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 t объектов с наибольшей ошибкой. Это соответствует ядру $\tilde{K}(\varepsilon) = \big[\varepsilon \leqslant \varepsilon^{(\ell-t)}\big]$.

Мягкая фильтрация [3]: используется квартическое ядро $\tilde{K}(\varepsilon) = K_Q\big(\frac{\varepsilon_i}{6\,\mathrm{med}\{\varepsilon_i\}}\big)$, где $\mathrm{med}\{\varepsilon_i\}$ — медиана вариационного ряда ошибок.

1.2.4 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, см. Рис **??**. Смещение возникает, когда объекты выборки x_i располагаются 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.

Введём для краткости сокращённые обозначения $w_i = w_i(x),\ d_i = x_i - x$.

Рассмотрим одномерный случай $a(u) = \alpha(u - x) + \beta$. Поскольку $a(x) = \beta$, достаточно найти только коэффициент β. 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 $a(u) = \alpha^\intercal(u - x) + \beta$ приходится решать задачу многомерной линейной регрессии, см. §1.3. Причём она должна решаться заново для каждой точки $x \in X$, что сопряжено с большим объёмом вычислений.

§1.3 Многомерная линейная регрессия

Пусть имеется набор n вещественнозначных признаков $f_j(x)$, $j = 1, \ldots, n$. Решение системы (1.3) существенно упрощается, если модель алгоритмов линейна по параметрам $\alpha \in \mathbb{R}^n$:

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

Введём матричные обозначения: матрицу информации F, целевой вектор y, вектор параметров α и диагональную матрицу весов W:

$$F = \begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \ldots \\ y_\ell \end{pmatrix}, \quad \alpha = \begin{pmatrix} \alpha_1 \\ \ldots \\ \alpha_n \end{pmatrix}, \quad W = \begin{pmatrix} \sqrt{w_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{w_\ell} \end{pmatrix}.$$

В матричных обозначениях функционал среднего квадрата ошибки принимает вид

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

Функционал с произвольными весами легко приводится к функционалу с единичными весами путём несложной предварительной обработки данных $F' = WF$, $y' = Wy$:

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

Поэтому в дальнейшем будем рассматривать только задачу с единичными весами.

Полиномиальная регрессия. Если $X = \mathbb{R}$, а признаками являются всевозможные степени $f_j(x) = x^{j-1}$, то говорят о *полиномиальной регрессии*:

$$f(x, \alpha) = \sum_{j=1}^{p} \alpha_j x^{j-1}.$$

В этом случае матрица F является *матрицей Вандермонда*:

$$F = \begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{p-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_\ell & x_\ell^2 & \ldots & x_\ell^{p-1} \end{pmatrix}.$$

Полиномиальная регрессия традиционно применяется для интерполяции и аппроксимации (сглаживания) функций одного переменного. В последнее время для этих целей предпочитают использовать непараметрические методы: ядерное сглаживание, сплайны или вейвлеты, отличающиеся лучшей численной устойчивостью и возможностью контролировать точность аппроксимации.

Криволинейная регрессия. Следующим частным случаем является *криволинейная регрессия*, когда исходные признаки $f_1, \ldots, f_n$ подвергаются некоторым преобразованиям $\varphi_1, \ldots, \varphi_k$, в общем случае нелинейным:

$$f(x, \alpha) = \sum_{j=1}^{k} \alpha_j \varphi_j\big(f_1(x), \ldots, f_n(x)\big).$$

Формально постановка задачи остаётся той же, если функции $\varphi_j\big(f_1(x), \ldots, f_n(x)\big)$ рассматривать как новые признаки $\varphi_j(x)$. Матрица $F_{\ell \times k}$ называется *обобщённой матрицей Вандермонда*:

$$F = \begin{pmatrix} \varphi_1(x_1) & \ldots & \varphi_k(x_1) \\ \ldots & \ldots & \ldots \\ \varphi_1(x_\ell) & \ldots & \varphi_k(x_\ell) \end{pmatrix}.$$

1.3.1 Нормальная система уравнений

Запишем необходимое условие минимума (1.3) в матричном виде:

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

откуда следует

$$F^\intercal F\alpha = F^\intercal y.$$

Эта система линейных уравнений относительно α называется *нормальной системой* для задачи наименьших квадратов. Матрица $F^\intercal F$ имеет размер $n \times n$ и совпадает с ковариационной матрицей набора признаков $f_1, \ldots, f_n$. Если она невырождена, то решением 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 $Q(\alpha^*) = \|P_F y - y\|$ есть длина 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.

1.3.2 Сингулярное разложение

Если число признаков не превышает число объектов, $n \leqslant \ell$, и среди столбцов матрицы F нет линейно зависимых, то F можно представить в виде *сингулярного разложения* (singular value decomposition, SVD)

$$F = VDU^\intercal,$$

обладающего рядом замечательных свойств (позже мы докажем Теорему 1.3, из которой эти свойства будут вытекать как следствия):

1) $\ell \times n$ матрица V ортогональна, $V^\intercal V = I_n$, и составлена из n собственных векторов матрицы $FF^\intercal$, соответствующих ненулевым собственным значениям;

2) $n \times n$ матрица U ортогональна, $U^\intercal U = I_n$, и составлена из собственных векторов матрицы $F^\intercal F$;

3) $n \times n$ матрица D диагональна, $D = \operatorname{diag}\bigl(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}\bigr)$, где $\lambda_1, \ldots, \lambda_n$ — собственные значения матриц $F^\intercal F$ и $FF^\intercal$.

Имея сингулярное разложение, можно выписать решение задачи наименьших квадратов в явном виде, не прибегая к трудоёмкому обращению матриц:

$$F^+ = (UDV^\intercal VDU^\intercal)^{-1}UDV^\intercal = (DU^\intercal)^{-1}(UD)^{-1}(UD)V^\intercal = UD^{-1}V^\intercal.$$

Обращение диагональной матрицы тривиально: $D^{-1} = \operatorname{diag}\bigl(\frac{1}{\sqrt{\lambda_1}}, \ldots, \frac{1}{\sqrt{\lambda_n}}\bigr)$.

Теперь легко выписать вектор МНК-решения:

$$\alpha^* = F^+ y = UD^{-1}V^\intercal y,$$

и МНК-аппроксимацию целевого вектора y:

$$F\alpha^* = P_F y = (VDU^\intercal)UD^{-1}V^\intercal y = VV^\intercal y = \sum_{j=1}^{n} v_j (v_j^\intercal y). \tag{1.5}$$

Эта формула означает, что МНК-аппроксимация $F\alpha^*$ есть проекция целевого вектора y на линейную оболочку собственных векторов матрицы $FF^\intercal$.

Ещё полезно вычислить норму вектора коэффициентов:

$$\|\alpha^*\|^2 = \|D^{-1}V^\intercal y\|^2 = \sum_{j=1}^{n} \frac{1}{\lambda_j}(v_j^\intercal y)^2. \tag{1.6}$$

Кажущаяся простота решения (1.5) компенсируется тем, что вычисление сингулярного разложения столь же трудоёмко, как и обращение матрицы $F^\intercal F$. Эффективные численные алгоритмы, выполняющие эту работу, в частности, QR-алгоритм, реализованы во многих стандартных математических пакетах.

1.3.3 Стандартизация данных

В прикладной задаче признаки могут иметь различный «физический» смысл и размерности. Если масштабы измерения признаков резко отличаются, то при вычислении F^+ возможно накопление ошибок, связанное с разрядностью представления чисел в компьютере. Для повышения численной устойчивости алгоритма выполняют предварительную *стандартизацию* данных в матрице $F = (f_{ij})$:

$$f_{ij} := (f_{ij} - \bar{f}_j)/\sigma_j, \quad j = 1, \ldots, n, \quad i = 1, \ldots, \ell, \tag{1.7}$$

где $\bar{f}_j = \frac{1}{\ell}\sum_{i=1}^{\ell} f_{ij}$ — выборочное среднее, $\sigma_j^2 = \frac{1}{\ell}\sum_{i=1}^{\ell}(f_{ij} - \bar{f}_j)^2$ — выборочная дисперсия j-го признака. После обучения по стандартизованным данным преобразование (1.7) придётся применять и к любому объекту x, подаваемому на вход построенного алгоритма $a(x) = f(x, \alpha^*)$.

Отметим, что до стандартизации $F^\intercal F$ является ковариационной матрицей, а после стандартизации — корреляционной.

1.3.4 Проблема мультиколлинеарности

Если ковариационная матрица $\Sigma = F^\intercal F$ имеет неполный ранг, то её обращение невозможно. На практике шансы «наткнуться» на матрицу неполного ранга исчезающе малы, хотя бы вследствие измерительных и вычислительных погрешностей. Гораздо чаще встречается случай *мультиколлинеарности*, когда матрица Σ имеет полный ранг, но близка к некоторой матрице неполного ранга. Тогда говорят, что Σ — *матрица неполного псевдоранга*, а также что матрица Σ *плохо обусловлена*. Столбцы такой матрицы *почти линейно зависимы*, то есть условие линейной зависимости выполняется не точно, а приближённо. То же самое можно сказать и о матрице F. 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 $\lambda_{\max}$ и $\lambda_{\min}$ — максимальное и минимальное собственные значения матрицы Σ, все нормы евклидовы. Матрицу принято считать плохо обусловленной, если $\mu(\Sigma)$ превышает $10^3..10^6$. Обращение такой матрицы численно неустойчиво. Умножение обратной матрицы на вектор может усиливать его относительную погрешность в $\mu(\Sigma)$ раз: если $z = \Sigma^{-1}u$, то

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

Негативные последствия мультиколлинеарности для линейной регрессии:

- Из формулы (1.6) следует, что увеличивается разброс коэффициентов α, появляются большие положительные и большие отрицательные коэффициенты. По абсолютной величине коэффициента α_j становится невозможно судить о степени важности признака f_j. Коэффициенты утрачивают интерпретируемость.

- Повышается неустойчивость решения. Малые изменения данных, такие как добавление нового обучающего объекта или шумовые искажения значений признаков или ответов, способны существенно изменить вектор коэффициентов.

- В результате понижается обобщающая способность алгоритма.

Три стратегии сокращения размерности для линейной регрессии. Далее будут рассмотрены три основных подхода к устранению мультиколлинеарности.

- *Регуляризация.* Накладываются дополнительные ограничения на норму вектора коэффициентов α. Это приводит к гребневой регрессии (см. 1.3.5) или методу лассо (см. 1.3.6). В первом случае задействуются все признаки, но снижается эффективная размерность пространства. Во втором случае у части признаков обнуляются коэффициенты α_j, что равносильно их исключению из модели.

- *Преобразование признаков.* Исходные n признаков с помощью некоторых преобразований переводятся в меньшее число m новых признаков. В частности, линейные преобразования приводят к методу главных компонент (см. 1.3.8).

- *Отбор признаков.* Производится явный перебор всевозможных подмножеств признаков. В **??** рассматриваются общие методы отбора признаков, применимые и к нелинейной регрессии, и к задачам классификации. Для линейной регрессии удаётся строить эффективные методы, совмещающие перебор подмножеств с оптимизацией коэффициентов. К таким методам относятся, опять-таки, лассо и ортогонализация Грама–Шмидта (см. 1.3.9).

1.3.5 Гребневая регрессия

Для решения проблемы мультиколлинеарности припишем к функционалу Q дополнительное слагаемое, штрафующее 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 Q значения, близкие к минимальному. Штрафное слагаемое выполняет роль регуляризатора, 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* (ridge regression). Добавление гребня к матрице $F^\intercal F$ увеличивает все её собственные значения на τ, но не изменяет её собственных векторов. В результате матрица становится хорошо обусловленной, оставаясь в то же время «похожей» на исходную. Аналогичный приём применялся в разделе **??** при обращении ковариационной матрицы Σ в линейном дискриминанте Фишера.

Выразим регуляризованное МНК-решение через сингулярное разложение:

$$\alpha_\tau^* = (UD^2U^\intercal + \tau I_n)^{-1}UDV^\intercal y = U(D^2 + \tau I_n)^{-1}DV^\intercal y = U\,\mathrm{diag}\left(\frac{\sqrt{\lambda_j}}{\lambda_j + \tau}\right)V^\intercal y.$$

Теперь найдём регуляризованную МНК-аппроксимацию целевого вектора y:

$$F\alpha_\tau^* = VDU^\intercal\alpha_\tau^* = V\,\mathrm{diag}\left(\frac{\lambda_j}{\lambda_j + \tau}\right)V^\intercal y = \sum_{j=1}^{n} v_j(v_j^\intercal y)\frac{\lambda_j}{\lambda_j + \tau}. \tag{1.8}$$

Как и прежде в (1.5), МНК-аппроксимация представляется в виде разложения целевого вектора y по 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 (1.6) уменьшается и норма вектора коэффициентов:

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

Отсюда ещё одно название метода — *сжатие* (shrinkage) или *сокращение весов* (weight decay) [5].

По мере увеличения параметра τ вектор коэффициентов α_τ^* становится всё более устойчивым и жёстко определённым. Фактически, происходит понижение *эффективной размерности* решения. Можно показать, что роль размерности играет след проекционной матрицы. Действительно, в нерегуляризованном случае имеем

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

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

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

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 τ^*. Для его нахождения применяется скользящий контроль, обычно hold-out или k-fold CV, см. раздел **??** или, более подробно, **??**. Зависимость этого функционала от параметра τ, как правило, имеет характерный минимум.

Скользящий контроль — очень долгая процедура. На практике параметр τ назначают в диапазоне от 0.1 до 0.4, если столбцы матрицы F были заранее стандартизованы. Ещё одна эвристика — выбрать τ так, чтобы число обусловленности приняло заданное не слишком большое значение M_0:

$$\mu(F^\intercal F + \tau I_n) = \frac{\lambda_{\max} + \tau}{\lambda_{\min} + \tau} = M_0,$$

откуда следует рекомендация $\tau^* \approx \lambda_{\max}/M_0$.

1.3.6 Лассо Тибширани

Ещё один метод регуляризации внешне очень 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:

$$\begin{cases} Q(\alpha) = \|F\alpha - y\|^2 \to \min\limits_{\alpha}; \\ \sum\limits_{j=1}^{n} |\alpha_j| \leqslant \varkappa; \end{cases} \tag{1.9}$$

где $\varkappa$ — параметр регуляризации. При больших значениях $\varkappa$ ограничение (1.9) обращается в строгое неравенство, и решение совпадает с методом наименьших квадратов. Чем меньше $\varkappa$, тем больше коэффициентов α_j принимают нулевое значение. Образно говоря, параметр $\varkappa$ зажимает вектор коэффициентов, заставляя его отказываться от лишних степеней свободы. Отсюда и название метода — *лассо* (LASSO, least absolute shrinkage and selection operator) [6].

Итак, лассо осуществляет отбор информативных признаков, хотя изначально задача так не ставилась. Чтобы понять, почему это происходит, приведём задачу математического программирования к каноническому виду. Введём вместо каждой переменной α_j две неотрицательные переменные: $\alpha_j = \alpha_j^+ - \alpha_j^-$, $\alpha_j^+ \geqslant 0$, $\alpha_j^- \geqslant 0$. Тогда минимизируемый функционал останется квадратичным по новым переменным, а ограничение (1.9) примет линейный вид:

$$\sum_{j=1}^{n} \alpha_j^+ - \alpha_j^- \leqslant \tau.$$

При уменьшении τ всё большее число ограничений-неравенств становятся активными, превращаясь в строгие равенства $\alpha_j^+ = \alpha_j^- = 0$, что соответствует обнулению коэффициента α_j и исключению j-го признака.

Сравнение лассо и гребневой регрессии. Оба метода успешно решают проблемы мультиколлинеарности, переобучения, и уменьшают разброс коэффициентов. Гребневая регрессия использует все признаки, стараясь «выжать максимум» из всей имеющейся информации. Лассо производит отбор признаков, что предпочтительнее, когда среди признаков имеются шумовые, или измерения признаков связаны с ощутимыми затратами.

На Рис. **??** левый ряд графиков соответствует гребневой регрессии, правый ряд — лассо. В верхнем ряду графиков показаны зависимости коэффициентов регрессии α_j от параметров регуляризации τ^{-1} или $\varkappa$. В нижнем ряду графиков показаны зависимости среднеквадратичной ошибки на обучении $Q(a^*, X^\ell)$ и на контроле $Q(a^*, X^k)$, а также евклидовой нормы вектора коэффициентов $\|\alpha^*\|$ от тех же параметров регуляризации τ и $\varkappa^{-1}$. Из графиков хорошо видно, что ослабление регуляризации ведёт к монотонному 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.

Существуют оптимальные значения параметров регуляризации, при которых алгоритм имеет наилучшую обобщающую способность. Для их подбора используют контрольную выборку, скользящий контроль или другие *внешние критерии*, см. **??**.

1.3.7 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 отклик y». Возникает задача минимизации квадратичного функционала $Q(\alpha, X^\ell)$ при ограничениях-неравенствах

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

Метод решения данной задачи основан на применении теоремы Куна-Таккера. Когда ограничение $\alpha_j \geqslant 0$ становится активным, то есть обращается в равенство, признак f_j, фактически, исключается из уравнения регрессии.

1.3.8 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* (principal component analysis, PCA) строится минимальное число новых признаков, по которым исходные признаки восстанавливаются линейным преобразованием с минимальными погрешностями. PCA относится к методам *обучения без учителя* (unsupervised learning), поскольку преобразование строится только по матрице «объекты–признаки» F без учёта целевого вектора y.

Пусть имеется n исходных числовых признаков $f_j(x)$, $j = 1, \ldots, n$. 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 $Z = \mathbb{R}^m$ меньшей 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 матрицу новых признаковых описаний G и матрицу линейного преобразования U, при которых суммарная невязка восстановленных описаний минимальна:

$$\Delta^2(G,U) = \sum_{i=1}^{\ell} \|\hat{x}_i - x_i\|^2 = \sum_{i=1}^{\ell} \|z_i U^\intercal - x_i\|^2 = \|GU^\intercal - F\|^2 \to \min_{G,U}, \tag{1.10}$$

где нормы векторов и матриц понимаются в смысле суммы квадратов элементов. Напомним, что $\|A\|^2 = \operatorname{tr} AA^\intercal = \operatorname{tr} A^\intercal A$, где tr — операция следа матрицы.

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

Исчерпывающее решение задачи (1.10) даёт следующая теорема.

Теорема 1.3. *Если $m = \operatorname{rg} G \leqslant \operatorname{rg} F$, 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

$$\begin{cases} G = FU(U^\intercal U)^{-1}; \\ U = F^\intercal G(G^\intercal G)^{-1}. \end{cases} \tag{1.11}$$

Функционал $\Delta^2(G,U)$ зависит только от произведения матриц $GU^\intercal$, поэтому решение задачи (1.10) 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 (1.10).

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 $T^\intercal(S^\intercal\tilde{G}^\intercal\tilde{G}S)T = \operatorname{diag}(\lambda_1,\ldots,\lambda_m) \equiv \Lambda$ — 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 (1.10) и удовлетворяют необходимому условию минимума. Подставим матрицы G и U в (1.11).

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 U обязаны 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 G являются 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 1.3 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. 9, непосредственно вытекают из Теоремы 1.3.

Преобразование Карунена–Лоэва. Диагональность матрицы $G^\intercal G = \Lambda$ означает, что новые признаки $g_1, \ldots, g_m$ не коррелируют на обучающих объектах. Поэтому ортогональное преобразование U называют *декоррелирующим*, а также преобразованием *Карунена–Лоэва*. 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, $G\beta = GU^\intercal U\beta = GU^\intercal\alpha$, 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:

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

Для вектора $\alpha^* = U\beta^*$ решение выглядит аналогично (1.5), только теперь матрицы U и V имеют меньшее число столбцов $m \leqslant n$.

Эффективная размерность выборки. Главные компоненты содержат основную информацию о матрице F. Если точность приближения $F \approx GU^\intercal$ вполне удовлетворяет, то остальные собственные векторы можно отбросить, то есть считать неинформативными. Число главных компонент m называют *эффективной размерностью* выборки. На практике её определяют следующим образом. Все собственные значения матрицы $F^\intercal F$ упорядочиваются по убыванию: $\lambda_1 \geqslant \ldots \geqslant \lambda_n \geqslant 0$. Задаётся пороговое значение $\varepsilon \in [0, 1]$, достаточно близкое к нулю, и определяется наименьшее целое m, при котором относительная погрешность приближения матрицы F не 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 n на 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 ε из 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 мало́.

Другие методы *синтеза признаков* (feature extraction) рассматриваются в **??**.

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 вряд ли подходит для получения точных количественных результатов. Обычно он используется на этапе предварительного обследования данных с целью понимания данных и выработки стратегии дальнейшего анализа. Аналогичную роль играют карты сходства (см. **??**) и карты Кохонена (см. **??**).

1.3.9 Метод ортогонализации Грама-Шмидта

Рассмотрим ортогональное разложение $F = GR$, где R — верхняя треугольная матрица размера $n \times n$, G — ортогональная $\ell \times n$-матрица, $G^\intercal G = I_n$. Для любой матрицы F существует бесконечно много разложений указанного вида. Имея одно из них, легко выразить псевдообратную матрицу F^+ через G и R:

$$F^+ = (R^\intercal G^\intercal G R)^{-1} R^\intercal G^\intercal = R^{-1} R^{-1\intercal} R^\intercal G^\intercal = R^{-1} G^\intercal.$$

Для вычисления псевдообратной F^+ достаточно построить какое-нибудь ортогональное разложение матрицы F, обратить верхнюю треугольную матрицу R и умножить её на $G^\intercal$. Этот метод во многом удобнее явного обращения матрицы.

Для построения такого разложения воспользуемся процессом ортогонализации Грама-Шмидта. Запишем матрицы F и G по столбцам:

$$F = (f_1, \ldots, f_n);$$
$$G = (\tilde{g}_1, \ldots, \tilde{g}_n).$$

Волной здесь и далее обозначается операция нормирования вектора:

$$\tilde{v} = v / \|v\|.$$

Процесс ортогонализации Грама-Шмидта строит ортогональные векторы $g_1, \ldots, g_n$, линейная оболочка которых совпадает с линейной оболочкой $f_1, \ldots, f_n$:

$$\begin{aligned} g_1 &= f_1; \\ g_2 &= f_2 - \tilde{g}_1 \tilde{g}_1^\intercal f_2; \\ &\ldots \\ g_j &= f_j - \tilde{g}_1 \tilde{g}_1^\intercal f_j - \cdots - \tilde{g}_{j-1} \tilde{g}_{j-1}^\intercal f_j. \end{aligned} \tag{1.12}$$

Легко проверить, что для всех k, j из $\{1, \ldots, n\}$, $k \neq j$, векторы $\tilde{g}_k$ и $\tilde{g}_j$ ортогональны. Доказательство этого факта можно найти в любом учебнике по линейной алгебре.

Лемма 1.4. *На j-м шаге процесса, $j = 1, \ldots, n$, матрица $F_j = (f_1, \ldots, f_j)$ представима в виде ортогонального разложения $F_j = G_j R_j$, где*

$G_j = (\tilde{g}_1, \ldots, \tilde{g}_j)$ *— ортонормированная матрица;*

$R_j = \begin{pmatrix} r_{11} & \ldots & r_{1j} \\ & \ddots & \vdots \\ 0 & & r_{jj} \end{pmatrix}$ *— верхняя треугольная матрица,* $r_{ij} = \begin{cases} \tilde{g}_i^\intercal f_j, & i < j; \\ \|g_j\|, & i = j. \end{cases}$

Доказательство.

С учётом введённых обозначений равенство (1.12) принимает вид

$$f_j = \tilde{g}_1 r_{1j} + \cdots + \tilde{g}_{j-1} r_{j-1,j} + \tilde{g}_j r_{jj},$$

что совпадает с матричной записью $F_j = G_j R_j$. Матрица G_j является ортонормированной согласно предыдущей лемме. ■

По окончании процесса (1.12) получаем ортогональное разложение $F = G_n R_n$.

С вычислительной точки зрения процесс Грама-Шмидта удобен тем, что на каждом шаге матрицы G_j и R_j получаются путём дописывания справа по одному столбцу к матрицам G_{j-1} и R_{j-1} соответственно. При этом предыдущие столбцы не изменяются (если не считать изменением дописывание нулей снизу к матрице R_j — при разумной программной реализации эти нули всё равно не хранятся).

В следующей лемме утверждается, что обратная матрица $T_j = R_j^{-1}$ также является верхней треугольной и формируется путём дописывания столбцов справа.

Лемма 1.5. *Пусть матрицы R_j невырождены и в блочной записи имеют вид*

$$R_1 = (r_{11});$$
$$R_j = \begin{pmatrix} R_{j-1} & r_j \\ 0 & r_{jj} \end{pmatrix}, \quad j = 2, \ldots, n,$$

где r_{jj} — скаляр, r_j — вектор-столбец размера $(j-1)$. Тогда матрицы $T_j = R_j^{-1}$ могут быть вычислены по рекуррентной формуле

$$T_1 = (t_{11});$$
$$T_j = \begin{pmatrix} T_{j-1} & t_j \\ 0 & t_{jj} \end{pmatrix}, \quad j = 2, \ldots, n,$$

где $t_{jj} = 1/r_{jj}$ — скаляр, $t_j = -t_{jj} T_{j-1} r_j$ — вектор-столбец размера $(j-1)$.

Доказательство.

Утверждение доказывается по индукции. Очевидно, $R_1 T_1 = I_1$. Непосредственным перемножением убеждаемся, что если $R_{j-1} T_{j-1}$ — единичная матрица размера $(j-1)$, то $R_j T_j$ — единичная матрица размера j. ■

Замечание 1.1. Обеспечить невырожденность матрицы R_j в процессе ортогонализации очень просто. Допустим, матрица R_{j-1} невырождена. Поскольку R_j — верхняя треугольная, вырожденность может возникнуть только в том случае, если $r_{jj} = 0$. Такое возможно только при $g_j = 0$, а это означает, что вектор f_j линейно зависит от векторов $\{f_1, \ldots, f_{j-1}\}$. Если в ходе процесса r_{jj} оказывается равным нулю, то коэффициент α_j обнуляется и j-й признак далее не учитывается, как будто его вообще не существовало. Если r_{jj} не равен, но близок к нулю, может возникнуть проблема неустойчивости решения, поскольку на r_{jj} приходится делить. На практике признак f_j исключают, например, по такому условию: $r_{jj} < \delta \max\limits_{i<j} r_{ii}$, где δ имеет порядок $10^{-2}..10^{-5}$.

Назовём вектор $\alpha_j = F_j^+ y$ *текущим вектором коэффициентов на j-м шаге.* Этот вектор имеет размерность j. По окончании процесса $\alpha_n = \alpha^*$.

Лемма 1.6. *Пусть выполняются условия предыдущей леммы. Тогда на j-м шаге процесса вектор α_j может быть вычислен по рекуррентной формуле*

$$\alpha_1 = t_{11}(y^\intercal \tilde{g}_1),$$
$$\alpha_j = \begin{pmatrix} \alpha_{j-1} + t_j(y^\intercal \tilde{g}_j) \\ t_{jj}(y^\intercal \tilde{g}_j) \end{pmatrix}, \quad j = 2, \ldots, n.$$

Доказательство.
На первом шаге процесса имеем $\alpha_1 = T_1 G_1^\intercal y = t_{11}(y^\intercal \tilde{g}_1)$. На следующих шагах

$$\alpha_j = T_j G_j^\intercal y = \begin{pmatrix} T_{j-1} & t_j \\ 0 & t_{jj} \end{pmatrix} \begin{pmatrix} G_{j-1}^\intercal \\ \tilde{g}_j^\intercal \end{pmatrix} y = \begin{pmatrix} T_{j-1} G_{j-1}^\intercal y + t_j \tilde{g}_j^\intercal y \\ t_{jj} \tilde{g}_j^\intercal y \end{pmatrix}, \quad j = 2, \ldots, n,$$

откуда следует требуемая рекуррентная формула. ■

Назовём величину $Q_j = \min\limits_{\alpha} \|y - F_j\alpha\|^2 = \|y - F_j\alpha_j\|^2$ *текущим значением функционала* Q *на* j*-м шаге*. Оно равно кратчайшему расстоянию от y до линейной оболочки столбцов F_j. По окончании процесса $Q_n = Q(\alpha^*)$. Следующая лемма показывает, что текущее значение Q_j от шага к шагу только уменьшается.

Лемма 1.7. *Значения Q_j могут быть вычислены в ходе ортогонализации по рекуррентной формуле*

$$Q_0 = \|y\|^2;$$
$$Q_j = Q_{j-1} - (y^\intercal \tilde{g}_j)^2, \quad j = 1, \ldots, n.$$

Доказательство.
Воспользуемся разложением псевдообратной матрицы $F_j^+ = T_j G_j^\intercal$:

$$\begin{aligned} Q_j &= \|y - F_j F_j^+ y\|^2 = \|y - G_j R_j R_j^{-1} G_j^\intercal y\|^2 = \\ &= (y - G_j G_j^\intercal y)^\intercal (y - G_j G_j^\intercal y) = y^\intercal y - y^\intercal G_j G_j^\intercal y = \\ &= \|y\|^2 - y^\intercal (\tilde{g}_1 \tilde{g}_1^\intercal + \cdots + \tilde{g}_j \tilde{g}_j^\intercal) y = \|y\|^2 - \sum_{s=1}^{j} (y^\intercal \tilde{g}_s)^2, \end{aligned}$$

откуда следует требуемая рекуррентная формула. ■

Совершенно аналогично доказывается

Лемма 1.8. *Текущий вектор невязок $\varepsilon_j = y - F_j\alpha_j$ на j-м шаге процесса ортогонализации вычисляется по рекуррентной формуле*

$$\varepsilon_0 = y;$$
$$\varepsilon_j = \varepsilon_{j-1} - \tilde{g}_j (y^\intercal \tilde{g}_j), \quad j = 1, \ldots, n.$$

Модифицированная ортогонализация Грама-Шмидта. Если вместо $r_{ij} := \tilde{g}_i^\intercal f_j$ вычислять $r_{ij} := \tilde{g}_i^\intercal g_j$, то формально результат не изменится, поскольку

$$\tilde{g}_i^\intercal g_j = \tilde{g}_i^\intercal \left(f_j - \sum_{s=1}^{i-1} \tilde{g}_s r_{sj} \right) = \tilde{g}_i^\intercal f_j - \sum_{s=1}^{i-1} \underbrace{\tilde{g}_i^\intercal \tilde{g}_s}_{0} r_{sj} = \tilde{g}_i^\intercal f_j.$$

Данная модификация повышает численную устойчивость алгоритма. Это объясняется тем, что вектор g_j обладает минимальной нормой среди всех векторов вида $f_j - \sum_{s=1}^{i-1} \beta_s \tilde{g}_s$, где β_s — произвольные коэффициенты [1]. Поэтому при скалярном умножении на g_j ошибки накапливаются существенно медленнее.

Прежде чем переходить к следующей модификации, запишем основную часть алгоритма ортогонализации, вычисляющую G_j и R_j.

Алгоритм 1.2. Ортогонализация Грама-Шмидта.

1: инициализация: $g_j := f_j,\ \ j := 1, \ldots, n$;
2: **для** $j := 1, \ldots, n$
3: **для** $i := 1, \ldots, j-1$
4: $r_{ij} := \tilde{g}_i^{\intercal} g_j$; (вычисление i-й компоненты вектор-столбца $r_j \in \mathbb{R}^{j-1}$);
5: $g_j := g_j - \tilde{g}_i r_{ij}$; (ортогонализация g_j относительно g_i);
6: $r_{jj} := \|g_j\|$;

Изменим порядок ортогонализации столбцов. До сих пор мы ортогонализовали столбец g_j относительно предыдущих столбцов $g_1, \ldots, g_{j-1}$. Но можно сделать и по-другому — ортогонализовать все последующие столбцы $g_{j+1}, \ldots, g_n$ относительно g_j:

$$g_i := g_i - \tilde{g}_j(\tilde{g}_j^{\intercal} g_i), \quad i = j+1, \ldots, n.$$

Тогда в начале j-го шага все столбцы $g_j, \ldots, g_n$ по построению будут ортогональны всем столбцам $g_1, \ldots, g_{j-1}$. При этом подматрицы G_j, R_j, T_j и вектор α_j останутся такими же, как и до модификации.

Описанная модификация обладает важным преимуществом. Теперь на каждом шаге можно выбрать столбец $g_m \in \{g_j, \ldots, g_n\}$, добавление которого наиболее выгодно. Чтобы не менять обозначений, будем полагать, что перед добавлением столбцы g_j и g_m переставляются местами (при реализации придётся сохранять соответствие между старой и новой нумерацией признаков, но мы не будем останавливаться на столь мелких технических деталях).

Возможны альтернативные критерии выбора добавляемого столбца:

1) столбец с максимальной нормой $\|g_m\| \to \max_m$, что соответствует выбору столбца f_m, максимально некоррелированного с $g_1, \ldots, g_{j-1}$; применение этого критерия решает проблему вырожденности R_j (см. Замечание 1.1); здесь существенно, чтобы матрица F была заранее стандартизована;

2) столбец, наиболее коррелированный с вектором ответов: $\frac{y^{\intercal} g_m}{\|g_m\|} \to \max_m$; его добавление ведёт к скорейшему убыванию функционала Q;

3) столбец, добавление которого ведёт к наименьшему увеличению нормы вектора коэффициентов $\|\alpha_j\|$, что соответствует применению регуляризации;

4) столбец, после добавления которого значение функционала качества Q на независимой контрольной выборке $X^k = \{x'_1, \ldots, x'_k\}$ окажется минимальным, что соответствует применению скользящего контроля (hold-out CV).

Наконец, можно использовать совокупность критериев, выбирая тот столбец, добавление которого выгодно с нескольких точек зрения.

В Алгоритме 1.3 применяются первые два критерия.

Алгоритм состоит из основного цикла по j, в котором поочерёдно добавляются столбцы. На шаге 3 принимается решение, какой из оставшихся столбцов f_m добавить, затем он меняется местами со столбцом f_j. Шаги 5–8 обновляют текущие значения функционала Q_j, обратной матрицы T_j и коэффициентов α_j. На шаге 9 проверяется условие останова: если достаточная точность аппроксимации уже достигнута, то добавлять оставшиеся столбцы не имеет смысла. Таким образом, Алгоритм 1.3

Алгоритм 1.3. Решение линейной задачи наименьших квадратов путём ортогонализации Грама-Шмидта с последовательным добавлением признаков

Вход:
$F = (f_1, \ldots, f_n)$ — матрица информации;
y — вектор ответов;
δQ — параметр критерия останова.

Выход:
α_j — вектор коэффициентов линейной комбинации;
Q_j — минимальное значение функционала.

1: инициализация:
$Q_0 := \|y\|^2$; $g_j := f_j$; $Z_j := \|g_j\|^2$; $D_j := y^\intercal g_j$; $j := 1, \ldots, n$;
2: **для** $j := 1, \ldots, n$
3: выбор $m \in \{j, \ldots, n\}$ по критериям $Z_m \to \max_m$ и $(D_m^2/Z_m) \to \max_m$;
4: перестановка местами столбцов:
$g_j \rightleftarrows g_m$, $f_j \rightleftarrows f_m$, $r_j \rightleftarrows r_m$;
5: $r_{jj} := \sqrt{Z_m}$; нормировка: $\tilde{g}_j := g_j / r_{jj}$;
6: вычисление текущего значения функционала:
$d_j := D_j / r_{jj}$; (эффективное вычисление $d_j := y^\intercal \tilde{g}_j$);
$Q_j := Q_{j-1} - d_j^2$;
7: обращение верхней треугольной матрицы $T_j = R_j^{-1}$:
$t_{jj} := 1/r_{jj}$; $t_j := -t_{jj} T_{j-1} r_j$; (вектор-столбец t_j длины $j-1$);
$T_j := \begin{pmatrix} T_{j-1} & t_j \\ 0 & t_{jj} \end{pmatrix}$;
8: вычисление текущего вектора коэффициентов:
$\alpha_j := \begin{pmatrix} \alpha_{j-1} + t_j d_j \\ t_{jj} d_j \end{pmatrix}$;
9: **если** $Q_j < \delta Q$ **то**
10: прекратить добавление столбцов; **выход**;
11: **для** $i := j+1, \ldots, n$
12: $r_{ji} := \tilde{g}_j^\intercal g_i$; (компоненты вектор-столбца r_i);
13: $g_i := g_i - \tilde{g}_j r_{ji}$; (ортогонализация g_i относительно g_j);
14: $Z_i := Z_i - r_{ji}^2$; (теперь $Z_i = \|g_i\|^2$);
15: $D_i := D_i - d_j r_{ji}$; (теперь $D_i = y^\intercal g_i$);
16: **конец** цикла по j.

осуществляет *отбор информативных признаков* (features selection). Шаги 11–15 реализуют вложенный цикл, в котором все последующие столбцы ортогонализуются относительно только что добавленного столбца. Заодно обновляются значения квадратов норм столбцов $Z_j = \|g_j\|^2$ и скалярных произведений $D_j = y^\intercal g_j$, необходимые для эффективного выбора признака f_m на шаге 3 в следующей итерации.

Отбор признаков и проблема переобучения. Если качество работы алгоритма вне обучающей выборки оказывается существенно хуже качества, достигнутого на обучающих данных, то говорят об *эффекте переобучения* или *переподгонки*.

Одна из возможных причин переобучения — наличие шумовых признаков, которые не несут никакой информации о восстанавливаемой зависимости. На практике такие признаки встречаются довольно часто, а в некоторых задачах их большинство. Причина в том, что на этапе формирования данных полезность признаков ещё не очевидна; поэтому измеряются все доступные характеристики объектов.

Включение шумового признака в регрессионную модель может только ухудшить её качество. Тем не менее, согласно лемме 1.7, добавление любого линейно независимого признака строго уменьшает остаточную сумму квадратов. Критерии 3) и 4) позволяют более надёжно отличать информативные признаки от шумовых.

1.3.10 Робастная регрессия

Метод наименьших квадратов довольно чувствителен к большим одиночным выбросам, связанным с грубыми ошибками в данных. Согласно теореме 1.1, в основе МНК лежит предположение о независимости и нормальности распределения ошибок. Методы, нечувствительные к искажениям предполагаемого распределения, называются *робастными*.

Простой эвристический способ построения робастного МНК состоит в просеивании (*цензурировании*) выборки. Задача решается несколько раз. После каждого раза из обучающей выборки исключается некоторая доля объектов, имеющих слишком большие невязки $\varepsilon_i = f(x_i, \alpha) - y_i$. Итерации продолжаются до тех пор, пока удаётся выделять объекты с большими невязками. На практике двух итераций, как правило, бывает достаточно. Максимальная доля отсеиваемых объектов задаётся исходя из содержания задачи. Например, если выбросы действительно обусловлены грубыми ошибками измерений, то на гистограмме распределения невязок соответствующие точки легко отделяются с помощью статистических или эвристических критериев.

Эта идея легко обобщается на тот случай, когда нет чёткого критерия для выделения выбросов. В разделе 1.2.3 уже был рассмотрен метод робастной непараметрической регрессии — локально взвешенное сглаживание. Это итерационный процесс, в котором объекты с бóльшими отклонениями получали меньший вес, после чего регрессия строилась заново. Итерации повторялись до тех пор, пока веса не стабилизируются. В случае параметрической регрессии пересчёт весов можно организовать точно также, и мы даже не будем останавливаться на деталях реализации.

Ещё один способ заключается в том, чтобы вместо квадратичной функции потерь ввести ограниченную сверху функцию, которая в окрестности нуля ведёт себя как квадратичная, а на бесконечности стремится к горизонтальной асимптоте. Такая функция потерь не различает большие и сверхбольшие отклонения. Классический пример — функция Мешалкина: $\mathscr{L}(a, y) = 1 - \exp\bigl(-\frac{1}{2\sigma}(a-y)^2\bigr)$, где σ — параметр, равный дисперсии «обычного» шума, не связанного с большими выбросами.

Задача минимизации функционала $Q(a)$ с такой функцией потерь уже не может быть решена средствами линейной алгебры; приходится применять численные методы оптимизации, например, метод сопряжённых градиентов.

§1.4 Нелинейные обобщения линейной 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 линейных задач.

1.4.1 Нелинейная модель регрессии

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

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

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

Запишем компоненты градиенты:

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

Запишем компоненты гессиана:

$$\frac{\partial^2}{\partial \alpha_j \partial \alpha_k} Q(\alpha) = 2 \sum_{i=1}^{\ell} \frac{\partial f}{\partial \alpha_j}(x_i, \alpha) \frac{\partial f}{\partial \alpha_k}(x_i, \alpha) - \underbrace{2 \sum_{i=1}^{\ell} \big(f(x_i, \alpha) - y_i\big) \frac{\partial^2 f}{\partial \alpha_j \partial \alpha_k}(x_i, \alpha)}_{\text{при линеаризации полагается равным 0}}.$$

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

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:

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

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 $\|F_t\delta - (f^t - y)\|^2 \to \min\limits_{\delta}$. 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.

1.4.2 Нелинейные одномерные преобразования признаков

На практике встречаются ситуации, когда линейная модель регрессии представляется необоснованной, но предложить адекватную нелинейную модель $f(x,\alpha)$ также не удаётся. Тогда в качестве компромисса строится модель вида

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

где $\varphi_j\colon \mathbb{R} \to \mathbb{R}$ — некоторые преобразования исходных признаков, в общем случае нелинейные. Задача состоит в том, чтобы одновременно подобрать и коэффициенты линейной модели α_j, и неизвестные одномерные преобразования φ_j, при которых достигается минимум квадратичного функционала (1.1).

Метод настройки с возвращениями основан на итерационном повторении двух шагов. На первом шаге фиксируются функции φ_j, и методами многомерной линейной регрессии вычисляются коэффициенты α_j. На втором шаге фиксируются коэффициенты α_j и все функции $\{\varphi_k\}_{k\neq j}$ кроме одной φ_j, которая настраивается методами одномерной непараметрической регрессии.

На втором шаге решается задача минимизации функционала

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

Здесь коэффициенты α_j и функции $\{\varphi_k\}_{k\neq j}$ фиксированы и не зависят от φ_j. Поэтому настройка φ_j сводится к стандартной задаче наименьших квадратов с обучающей выборкой $Z_j^\ell = \left(f_j(x_i), \frac{1}{\alpha_j}z_i\right)_{i=1}^{\ell}$. Для её решения годятся любые одномерные методы: ядерное сглаживание, сплайны, полиномиальная или Фурье-аппроксимация. Заметим, что для ядерного сглаживания с фиксированной шириной окна этап

Алгоритм 1.4. Метод настройки с возвращениями

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

1: нулевое приближение: $\varphi_j(u) \equiv u, \;\; j = 1, \ldots, n$;
2: **повторять**
3: $\quad \alpha :=$ решение задачи МЛР с признаками $\varphi_j(f_j(x))$;
4: $\quad$ **для** $j = 1, \ldots, n$
5: $\quad\quad z_i := \sum\limits_{k=1, k\neq j}^{n} \alpha_k \varphi_k(f_k(x_i)) - y_i, \;\; i = 1, \ldots, \ell$;
6: $\quad\quad \varphi_j := \arg\min\limits_{\varphi} \sum\limits_{i=1}^{\ell} \big(\varphi_j(f_j(x)) - \frac{1}{\alpha_j} z_i\big)^2$;
7: **пока** α не сойдётся

настройки функций φ_j фактически отсутствует; чтобы вычислять значения $\varphi_j(f)$ по формуле Надарая–Ватсона, достаточно просто запомнить выборку Z_j^ℓ, см. §1.2.

После настройки всех функций φ_j происходит возврат к первому шагу, и снова решается задача многомерной линейной регрессии для определения α_j. Отсюда и название метода — *настройка с возвращениями* (backfitting). Он был предложен Хасти и Тибширани в 1986 году [4]. Схема реализации показана в Алгоритме 1.4.

Гистограммный метод. Для каждого признака f_j, независимо от остальных, подбирается такое преобразование φ_j, чтобы значения нового признака $\varphi_j(f_j(x_i))$ имели распределение $\Phi(z) = \mathsf{P}_x\big\{\varphi_j(f_j(x)) < z\big\}$, близкое к заданному $\Phi_0(z)$. Чаще всего в качестве $\Phi_0(z)$ берут нормальное распределение; тогда распределение объектов в пространстве новых признаков становится похожим на n-мерное нормальное.

Значения $f_j(x_i), \;\; i = 1, \ldots, \ell$ упорядочиваются по возрастанию, образуя вариационный ряд $f_j^{(1)} < \cdots < f_j^{(\ell)}$. Если в этом ряду есть совпадающие значения (связки), то они удаляются и ряд укорачивается, так чтобы получилась строго возрастающая последовательность.

Строится обучающая выборка из пар чисел $Z_j^\ell = \big(f_j^{(i)}, \Phi_0^{-1}(\frac{i-0.5}{\ell})\big)_{i=1}^{\ell}$, где Φ_0^{-1} — обратная функция для заданной функции распределения $\Phi_0(z)$.

Методами одномерной непараметрической регрессии строится функция $\varphi_j(f)$, аппроксимирующая выборку Z_j^ℓ.

Нетрудно понять, что выборка Z_j^ℓ является монотонной:

$$f_j^{(i)} < f_j^{(i')} \;\; \Leftrightarrow \;\; i < i' \;\; \Leftrightarrow \;\; \Phi_0^{-1}\left(\tfrac{i-0.5}{\ell}\right) < \Phi_0^{-1}\left(\tfrac{i'-0.5}{\ell}\right).$$

Поэтому на функцию $\varphi_j(f)$ обычно накладывается дополнительное ограничение — она должна быть строго возрастающей. При использовании ядерного сглаживания с непрерывным ядром и достаточно широким окном выполнение этого требования гарантируется.

1.4.3 Обобщённые линейные модели

Рассмотрим другую ситуацию, когда модель регрессии $f(x,\alpha)$ линейна, но известна нелинейная функция связи $g(f)$ между выходом модели f и 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 Q аппроксимируется функционалом $\tilde{Q}$, квадратичным по вектору коэффициентов α:

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

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

1.4.4 Неквадратичные функции потерь

Функция потерь $\mathscr{L}(a, y)$ 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:

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

Если функция потерь квадратична, $\mathscr{L}(a, y) = (a - y)^2$, то минимизация Q соответствует 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 уже говорилось в начале главы, 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 1.1. При планировании закупок в сетевых супермаркетах 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 (для конкретности, в течение следующей недели). Квадрат отклонения $(a-y)^2$ прогноза a от реального спроса y экономического смысла не имеет. Гораздо удобнее измерять потери в рублях. Потери от заниженного прогноза $a < y$ связаны с недополученной прибылью и прямо пропорциональны величине отклонения: $\mathscr{L}(a, y) = c_1|a - y|$, где c_1 — коэффициент торговой наценки. Потери от завышенного прогноза $a > y$ связаны с замораживанием средств, затариванием 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: $\mathscr{L}(a, y) = c_2|a - y|$. Коэффициенты c_1 и c_2 для данного магазина известны, причём они зависят от товара и могут отличаться в десятки раз. Таким образом, в данной задаче более обоснованной оказывается не квадратичная, а кусочно-линейная несимметричная функция потерь.

Пример 1.2. При создании автоматических систем биржевой торговли строится алгоритм $a(x)$, прогнозирующий в момент времени x_i цену акции на следующий момент x_{i+1}. В данном случае ни величина отклонения $a - y$, ни, тем более, её квадрат, особого интереса не представляет. Экономический смысл имеет прибыль, 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»), затем принимаем следующее решение согласно алгоритму $a(x)$, и так далее. Суммарная прибыль, заработанная в течение ℓ последовательных моментов времени $x_1, \ldots, x_\ell$ имеет вид

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

Обучение алгоритма a игре на бирже сводится к максимизации функционала $Q(a)$ по обучающей последовательности цен $y(x_1), \ldots, y(x_{\ell+1})$. В данной задаче содержательно обоснованной оказалась кусочно-постоянная функция потерь.

Робастная регрессия уже рассматривалась выше в 1.3.10. Чтобы функционал $Q(a)$ был нечувствителен к сверхбольшим выбросам, вводится ограниченная сверху функция потерь, например, функция Мешалкина. Для минимизации функционала $Q(a)$ применяются численные методы, в частности, метод сопряжённых градиентов.

Логистическая регрессия. Неквадратичные функции потерь возникают также при попытке приспособить хорошо развитые методы многомерной линейной регрессии для решения задач классификации. Об этом — в следующем параграфе.

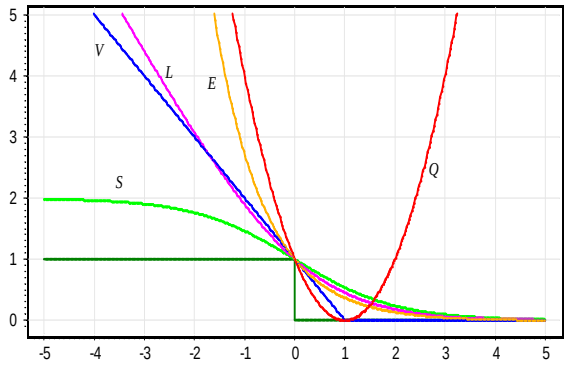

Рис. 1. Гладкие аппроксимации пороговой функции потерь $[M < 0]$.

$(1-M)^2$	— квадратичная (Q);
$(1-M)_+$	— кусочно-линейная (V);
e^{-M}	— экспоненциальная (E);
$2(1+e^M)^{-1}$	— сигмоидная (S);
$\log_2(1+e^{-M})$	— логарифмическая (L).

§1.5 Логистическая регрессия

Техника наименьших квадратов оказалась очень плодотворной в задаче восстановления регрессии. Нельзя ли применить её и для классификации?

Рассмотрим задачу классификации с двумя классами, $Y = \{-1, +1\}$.

Пусть модель алгоритмов задана в виде параметрического семейства отображений $\big\{\operatorname{sign} f(x,\alpha) \;\big|\; \alpha \in \mathbb{R}^n\big\}$. Функция $f(x,\alpha)$ называется *дискриминантной функцией*. Если $f(x,\alpha) > 0$, то алгоритм a относит объект x к классу $+1$, иначе к классу -1. Уравнение $f(x,\alpha) = 0$ описывает поверхность, разделяющую классы.

Величина $M_i(\alpha) = y_i f(x_i,\alpha)$ называется *отступом* (margin) объекта x_i от разделяющей поверхности. Алгоритм $a(x_i)$ допускает ошибку на объекте x_i тогда и только тогда, когда его отступ отрицателен.

Для настройки вектора коэффициентов α обычно минимизируют число ошибок на обучающей выборке:

$$Q(\alpha) = \sum_{i=1}^{\ell} [M_i(\alpha) < 0] = \sum_{i=1}^{\ell} [y_i f(x_i,\alpha) < 0] \to \min_{\alpha}.$$

Гладкие аппроксимации пороговой функции потерь. К сожалению, функционал $Q(\alpha)$ не является ни квадратичным, ни даже непрерывным, что исключает возможность аналитического решения оптимизационной задачи, как это было сделано в случае линейной регрессии. Однако можно заменить пороговую функцию потерь какой-нибудь её гладкой аппроксимацией, см. Рис. 1.

Квадратичная аппроксимация соответствует самому простому «наивному» сведению классификации к регрессии, когда регрессионными методами минимизируется функционал квадратичной невязки

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

На практике квадратичная аппроксимация может давать как хорошие, так и плохие результаты. Недостаток квадратичной функции потерь в том, что положительные и отрицательные отклонения отступа M_i от единицы штрафуются одинаково. В то же время, штрафовать следовало бы только отрицательные отступы. Большие положительные значения M_i свидетельствуют как раз о надёжной классификации объекта x_i.

Аппроксимации E, S и L лишены этого недостатка. Возникает вопрос: какая из аппроксимаций лучше? Далее мы покажем, что при некоторых ограничениях логарифмическая аппроксимация эквивалентна применению байесовского решающего правила и принципа максимума правдоподобия. Именно она и приводит к методу классификации, называемому *логистической регрессией.*

1.5.1 Обоснование логистической регрессии

Пусть объекты описываются n числовыми признаками $f_j\colon X \to \mathbb{R}$, $j = 1, \ldots, n$. Как обычно, будем полагать $X = \mathbb{R}^n$, отождествляя объекты с их признаковыми описаниями: $x \equiv (f_1(x), \ldots, f_n(x))^\intercal$.

Гипотеза 1.1. *Множество прецедентов $X \times Y$ является вероятностным пространством. Выборка прецедентов $X^\ell = (x_i, y_i)_{i=1}^\ell$ получена случайно и независимо согласно вероятностному распределению с плотностью $p(x, y) = P_y p_y(x) = \mathsf{P}\{y|x\}p(x)$, где P_y — априорные вероятности, $p_y(x)$ — функции правдоподобия, $\mathsf{P}\{y|x\}$ — апостериорные вероятности классов $y = -1, +1$.*

Опр. 1.1. *Плотность распределения $p_y(x)$, $x \in \mathbb{R}^n$ называется экспонентной, если*

$$p_y(x) = \exp\bigl(\theta^\intercal x \cdot a(\delta) + b(\delta, \theta) + d(x, \delta)\bigr),$$

где $\theta \in \mathbb{R}^n$ — векторный параметр, называемый сдвигом; δ — параметр, называемый разбросом; a, b, d — произвольные числовые функции.

Класс экспонентных распределений очень широк. К нему относятся многие непрерывные и дискретные распределения: равномерное, нормальное, гипергеометрическое, пуассоновское, биномиальное, Г-распределение, и другие.

Пример 1.3. Многомерное нормальное распределение с вектором матожидания $\mu \in \mathbb{R}^n$ и ковариационной матрицей $\Sigma \in \mathbb{R}^{n\times n}$ является экспонентным с параметром сдвига $\theta = \Sigma^{-1}\mu$ и параметром разброса $\delta = \Sigma$:

$$\begin{aligned}\mathcal{N}(x; \mu, \Sigma) &= (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\bigl(-\tfrac{1}{2}(x-\mu)^\intercal \Sigma^{-1}(x-\mu)\bigr) = \\ &= \exp\bigl(\underbrace{\mu^\intercal\Sigma^{-1}x}_{\theta^\intercal x} - \underbrace{\tfrac{1}{2}\mu^\intercal\Sigma^{-1}\Sigma\Sigma^{-1}\mu}_{b(\delta,\theta)} - \underbrace{\tfrac{1}{2}x^\intercal\Sigma^{-1}x - \tfrac{n}{2}\ln(2\pi) - \tfrac{1}{2}\ln|\Sigma|}_{d(x,\delta)}\bigr).\end{aligned}$$

Гипотеза 1.2. *Функции правдоподобия классов $p_y(x)$ принадлежат экспонентному семейству плотностей и имеют равные значения параметра разброса.*

Напомним, что оптимальное байесовское решающее правило имеет вид (**??**):

$$a(x) = \arg\max_{y\in Y} \lambda_y \mathsf{P}\{y|x\} = \operatorname{sign}\bigl(\lambda_+ \mathsf{P}\{+1|x\} - \lambda_- \mathsf{P}\{-1|x\}\bigr),$$

где λ_y — величина штрафа за ошибку на объектах класса y.

Теорема 1.9. *Если справедливы гипотезы 1.1, 1.2, и среди признаков $f_1(x), \ldots, f_n(x)$ есть константа, то байесовское решающее правило линейно:*

$$a(x) = \operatorname{sign}(\alpha^\intercal x - \alpha_0), \qquad \alpha_0 = \ln(\lambda_-/\lambda_+),$$

причём апостериорная вероятность принадлежности произвольного объекта $x \in X$ классу $y \in \{-1, +1\}$ связана со значением дискриминантной функции:

$$\mathsf{P}\{y|x\} = \sigma(y\,\alpha^\intercal x),$$

где $\sigma(z) = \frac{1}{1+e^{-z}}$ — функция, называемая логистической или сигмоидной.

Доказательство.

Рассмотрим отношение апостериорных вероятностей классов и воспользуемся тем, что $p_y(x)$ — экспонентные плотности с параметрами θ_y и δ:

$$\frac{\mathsf{P}\{+1|x\}}{\mathsf{P}\{-1|x\}} = \frac{P_+ p_+(x)}{P_- p_-(x)} = \exp\Big(\underbrace{a(\delta)(\theta_+ - \theta_-)}_{\alpha=\text{const}(x)}{}^\intercal x + \underbrace{b(\delta,\theta_+) - b(\delta,\theta_-) + \ln\tfrac{P_+}{P_-}}_{\text{const}(x)}\Big).$$

Здесь вектор α не зависит от x и является вектором свободных коэффициентов при признаках. Все слагаемые под экспонентой, не зависящие от x, можно считать аддитивной добавкой к коэффициенту при константном признаке. Поскольку свободные коэффициенты настраиваются по обучающей выборке, вычислять эту аддитивную добавку нет никакого смысла, и её можно включить в $\alpha^\intercal x$. Следовательно

$$\frac{\mathsf{P}\{+1|x\}}{\mathsf{P}\{-1|x\}} = \exp\big(\alpha^\intercal x\big).$$

Используя формулу полной вероятности $\mathsf{P}\{-1|x\} + \mathsf{P}\{+1|x\} = 1$, нетрудно выразить апостериорные вероятности $\mathsf{P}\{-1|x\}$ и $\mathsf{P}\{+1|x\}$ через $\alpha^\intercal x$:

$$\mathsf{P}\{+1|x\} = \sigma(\alpha^\intercal x);$$
$$\mathsf{P}\{-1|x\} = \sigma(-\alpha^\intercal x).$$

Объединяя эти два равенства в одно, получаем требуемое: $\mathsf{P}\{y|x\} = \sigma(y\,\alpha^\intercal x)$.

Разделяющая поверхность в байесовском решающем правиле определяется уравнением $\lambda_- \mathsf{P}\{-1|x\} = \lambda_+ \mathsf{P}\{+1|x\}$, которое равносильно $\alpha^\intercal x - \ln\frac{\lambda_-}{\lambda_+} = 0$, следовательно, разделяющая поверхность линейна. ■

Итак, мы выяснили, что при некоторых теоретико-вероятностных предположениях байесовское решающее правило является линейным и имеет n свободных параметров $\alpha = (\alpha_1, \ldots, \alpha_n)$. Чтобы настроить их по обучающей выборке X^ℓ, воспользуемся принципом максимума правдоподобия:

$$L(\alpha) = \ln \prod_{i=1}^{\ell} p_{y_i}(x_i) \to \max_\alpha,$$

где $p_y(x)$ — функция правдоподобия класса y. Согласно определению условной вероятности $p_y(x)P_y = \mathsf{P}\{y|x\}p(x)$, где априорные вероятности классов P_y и вероятности появления объектов $p(x)$ не зависят от вектора параметров α. Апостериорные вероятности выражаются согласно Теореме 1.9 через линейную дискриминантную функцию: $\mathsf{P}\{y|x\} = \sigma(y\,\alpha^\intercal x)$. Таким образом,

$$p_y(x) = \sigma(y\alpha^\intercal x) \cdot \text{const}(\alpha).$$

Подставим это выражение в функционал логарифма правдоподобия $L(\alpha)$:

$$L(\alpha) = \sum_{i=1}^{\ell} \ln \sigma(y_i \alpha^{\intercal} x_i) + \text{const}(\alpha) \to \max_{\alpha}.$$

Функция $\ln \sigma(z) = -\ln(1+e^{-z})$ с точностью до постоянного множителя $\ln 2$ совпадает с логарифмической аппроксимацией пороговой функции потерь, введённой выше. Поэтому максимизация логарифма правдоподобия $L(\alpha)$ эквивалентна минимизации функционала $Q(\alpha)$:

$$Q(\alpha) = \sum_{i=1}^{\ell} \ln(1 + e^{-y_i \alpha^{\intercal} x_i}) \to \min_{\alpha}.$$

Этим и завершается обоснование логистической регрессии.

Замечание 1.2. Линейный дискриминант Фишера (ЛДФ) и логистическая регрессия исходят из байесовского решающего правила и принципа максимума правдоподобия, однако результат получается разный. В ЛДФ приходится оценивать $n(n+1)/2$ параметров, в логистической регрессии — только n. Почему? Дело в том, что ЛДФ решает вспомогательную задачу восстановления плотностей распределения классов, предполагая к тому же, что плотности нормальны. Логистическая регрессия не пытается восстанавливать плотности классов и опирается на более слабые предположения о виде плотностей. С точки зрения *принципа Оккама* «не размножать сущности без необходимости» логистическая регрессия явно предпочтительнее, поскольку ЛДФ вводит избыточную сущность — плотности распределения классов, и сводит задачу классификации к более сложной задаче восстановления плотностей.

1.5.2 Итерационный взвешенный МНК

Стандартная техника настройки параметров α заключается в применении метода Ньютона-Рафсона для минимизации нелинейного функционала $Q(\alpha)$. В качестве нулевого приближения можно взять «наивное» решение задачи классификации каким-либо методом многомерной линейной регрессии. Затем начинается итерационный процесс, на t-м шаге которого уточняется вектор коэффициентов $\alpha^{(t+1)}$:

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

где $Q'(\alpha^t)$ — вектор первых производных (градиент) функционала $Q(\alpha)$ в точке α^t, $Q''(\alpha^t)$ — матрица вторых производных (гессиан) функционала $Q(\alpha)$ в точке α^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 $\sigma_i \equiv \sigma_i(\alpha) = \sigma(y_i \alpha^{\intercal} x_i)$ и заметим, что производная логистической функции есть $\sigma'(z) = \sigma(z)(1-\sigma(z))$.

Элементы градиента (вектора первых производных) функционала $Q(\alpha)$:

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

Элементы гессиана (матрицы вторых производных) функционала $Q(\alpha)$:

$$\frac{\partial^2 Q(\alpha)}{\partial\alpha_j\partial\alpha_k} = -\frac{\partial}{\partial\alpha_k}\sum_{i=1}^{\ell}(1-\sigma_i)y_i f_j(x_i) =$$

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

Введём матричные обозначения:
$F_{\ell\times n} = \big(f_j(x_i)\big)$ — матрица признаковых описаний объектов;
$W_{\ell\times\ell} = \operatorname{diag}\big(\sqrt{(1-\sigma_i)\sigma_i}\,\big)$ — диагональная матрица весов объектов;
$\tilde{F} = WF$ — 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:

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

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

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

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 итерационным перевзвешиванием* (iteratively reweighted least squares, IRLS)

Понять смысл этого пересчёта совсем нетрудно. Во-первых, заметим, что величина σ_i равна вероятности правильного ответа алгоритма α^t на объекте x_i. Поэтому вес w_i максимален 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 $\alpha^{(t)}$ на объекте x_i модифицированный ответ $\tilde{y}_i$ возрастает по модулю. Это приводит к повышению точности настройки алгоритма α^{t+1} на тех объектах, которые оказались «наиболее трудными» для алгоритма α^t на предыдущей итерации.

1.5.3 Оценивание рисков

Логистическая функция σ переводит значение линейной дискриминантной функции $\alpha^\intercal x$ в оценку вероятности того, что объект x принадлежит классу $+1$, см. Рис 2. Это свойство логистической регрессии активно используется в тех приложениях, где наряду с классификацией объекта x требуется оценить связанный с ним *риск* как математическое ожидание потерь:

$$R(x) = \sum_{y\in Y}\lambda_y\,\mathsf{P}\{y|x\}.$$

Алгоритм 1.5. IRLS — итерационный взвешенный метод наименьших квадратов

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

нулевое приближение — обычный МНК:
$\alpha := (F^\intercal F)^{-1} F^\intercal y$;
для $t := 1, 2, 3, \ldots$
  $z := F\alpha$;
  $w_i := \sqrt{(1-\sigma(z_i))\sigma(z_i)}$ для всех $i = 1, \ldots, \ell$;
  $\tilde{F} := \operatorname{diag}(w_1, \ldots, w_\ell) F$;
  $\tilde{y}_i := \sqrt{(1-\sigma(z_i))/\sigma(z_i)}$ для всех $i = 1, \ldots, \ell$;
  выбрать градиентный шаг h_t;
  $\alpha := \alpha + h_t (\tilde{F}^\intercal \tilde{F})^{-1} \tilde{F}^\intercal \tilde{y}$;
  если $\sigma(z_i)$ мало изменились относительно предыдущей итерации **то**
    прервать итерации, выйти из цикла;
конец цикла по t.

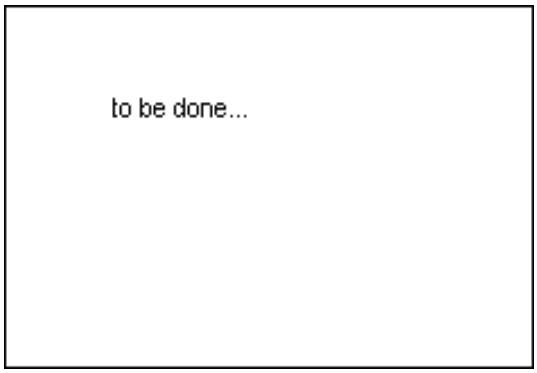

Рис. 2. Сигмоидная функция переводит значение дискриминантной функции в вероятность.

В практических ситуациях к оценкам риска следует относиться с осторожностью, поскольку значение $\sigma(\alpha^\intercal x)$ является, по сути дела, лишь эвристической оценкой вероятности $\mathsf{P}\{+1|x\}$. Как следует из условий Теоремы 1.9, эта оценка будет точной только для экспонентных классов с равными параметрами разброса. На практике экспонентность никогда не проверяется, а гарантировать одинаковость разбросов вообще не представляется возможным.

1.5.4 Кривая ошибок и выбор порогового параметра α_0

Согласно Теореме 1.9 пороговый параметр $\alpha_0 = \ln \frac{\lambda_-}{\lambda_+}$ зависит только от отношения величины потерь λ_+ (цена ошибки I рода, когда на объекте класса «+1» алгоритм выдаёт «−1») и λ_- (цена ошибки II рода, когда на объекте класса «−1» алгоритм выдаёт «+1»). Таким образом, значение α_0 целиком определяется спецификой задачи. На практике отношение потерь может многократно пересматриваться.

ROC-кривая показывает, что происходит с количеством ошибок I и II рода, если изменяется отношение потерь.

Термин *операционная характеристика приёмника* (receiver operating characteristic, ROC curve) пришёл из теории обработки сигналов. Эту характеристику впервые ввели во время II мировой войны, после поражения американского военного флота в Пёрл Харборе в 1941 году, когда остро встала проблема повышения точности распознавания самолётов противника по радиолокационному сигналу. Позже нашлись и дугие применения: медицинская диагностика, приёмочный контроль качества, кредитный скоринг, предсказание лояльности клиентов, и т. д.

Каждая точка на ROC-кривой соответствует некоторому алгоритму. В общем случае это даже не обязательно кривая — дискретное множество алгоритмов может быть отображено в тех же координатах в виде точечного графика.

По оси X откладывается доля *ошибочных положительных классификаций* (false positive rate, FPR):

$$\mathrm{FPR}(a, X^\ell) = \frac{\sum_{i=1}^{\ell} [y_i = -1][a(x_i) = +1]}{\sum_{i=1}^{\ell} [y_i = -1]}.$$

Величина $1 - \mathrm{FPR}(a)$ называется *специфичностью* алгоритма a. Поэтому на горизонтальной оси иногда пишут «1 − специфичность».

По оси Y откладывается доля *правильных положительных классификаций* (true positive rate, TPR), называемая также *чувствительностью* алгоритма a:

$$\mathrm{TPR}(a, X^\ell) = \frac{\sum_{i=1}^{\ell} [y_i = +1][a(x_i) = +1]}{\sum_{i=1}^{\ell} [y_i = +1]}.$$

В случае логистической регрессии каждая точка ROC-кривой соответствует определённому значению параметра α_0. При этом ROC-кривая монотонно не убывает и проходит из точки $(0, 0)$ в точку $(1, 1)$. Для построения ROC-кривой нет необходимости вычислять FPR и TPR суммированием по всей выборке при каждом α_0. Более эффективный Алгоритм 1.6 основан на простой идее, что в качестве значений порога α_0 достаточно перебрать только ℓ значений дискриминантной функции $f(x_i) = \alpha^\intercal x_i$, которые она принимает на объектах выборки.

Чем выше проходит ROC-кривая, тем выше качество классификации. Идеальная ROC-кривая представляет собой «угол», проходящий через точки $(0, 0)$, $(0, 1)$, $(1, 1)$. Наихудший алгоритм соответствует диагональной прямой, соединяющей точки $(0, 0)$ и $(1, 1)$; её также изображают на графике как ориентир.

В роли общей характеристики качества классификации, не зависящей от конъюнктурного параметра α_0, выступает *площадь под ROC-кривой* (area under curve, AUC). Её вычисление также показано в Алгоритме 1.6.

1.5.5 Скоринг

В случае бинарных признаков, $X = \{0, 1\}^n$, вычисление линейной дискриминантной функции удобно рассматривать как подсчёт *баллов* (score): если $f_j(x) = 1$, то есть признак f_j наблюдается у объекта x, то к сумме баллов добавляется вес α_j. Классификация производится путём сравнения набранной суммы баллов с пороговым значением α_0.

Алгоритм 1.6. Эффективный алгоритм построения ROC-кривой

Вход:
обучающая выборка X^ℓ;
$f(x) = \alpha^\intercal x$ — дискриминантная функция;
Выход:
$\big\{(\mathrm{FPR}_i, \mathrm{TPR}_i)\big\}_{i=0}^{\ell}$ — последовательность точек ROC-кривой;
AUC — площадь под ROC-кривой.

$\ell_- := \sum_{i=1}^{\ell}[y_i = -1]$ — число объектов класса -1;
$\ell_+ := \sum_{i=1}^{\ell}[y_i = +1]$ — число объектов класса $+1$;
упорядочить выборку X^ℓ по убыванию значений $f(x_i)$;
поставить первую точку в начало координат:
$(\mathrm{FPR}_0, \mathrm{TPR}_0) := (0, 0)$; $\mathrm{AUC} := 0$;
для $i := 1, \ldots, \ell$
  если $y_i = -1$ **то**
    сместиться на один шаг вправо:
    $\mathrm{FPR}_i := \mathrm{FPR}_{i-1} + \frac{1}{\ell_-}$; $\mathrm{TPR}_i := \mathrm{TPR}_{i-1}$;
    $\mathrm{AUC} := \mathrm{AUC} + \frac{1}{\ell_-}\mathrm{TPR}_i$;
  иначе
    сместиться на один шаг вверх:
    $\mathrm{FPR}_i := \mathrm{FPR}_{i-1}$; $\mathrm{TPR}_i := \mathrm{TPR}_{i-1} + \frac{1}{\ell_+}$;

Благодаря свой простоте подсчёт баллов или *скоринг* (scoring), пользуется большой популярностью в таких областях, как медицина, геология, банковское дело, социология, маркетинг, и др. Абсолютное значение веса α_j можно интерпретировать как степень важности признака f_j, а знак $\mathrm{sign}(\alpha_j)$ показывает, в пользу какого класса свидетельствует наличие данного признака. Это важная дополнительная информация о признаках, помогающая экспертам лучше понимать задачу.

Во многих прикладных задачах исходные данные содержат разнотипные признаки. Если признак не бинарный, то его *бинаризуют*, разбивая множество его значений на подмножества, например, с помощью методов, описанных в разделе **??**. В результате один небинарный признак заменяется несколькими бинарными.

После бинаризации классификатор представляется в виде так называемой *скоринговой карты* (scorecard), в которой перечисляются все исходные признаки, для каждого исходного — все построенные по нему бинарные признаки, для каждого бинарного — его вес. Имея такую карту, классификацию можно проводить с помощью стандартной электронной таблицы или даже вручную.

Пример 1.4. В задаче *кредитного скоринга* (credit scoring) признаковое описание заёмщика (физического лица) состоит из его ответов на вопросы анкеты. Среди признаков встречаются бинарные (пол, согласие дать телефон, наличие задолженностей), номинальные (место проживания, профессия, работодатель), порядковые (образование, должность) и количественные (сумма кредита, возраст, стаж, доход). Все они в конечном итоге приводятся к бинарному виду. Количество признаков при этом может существенно возрасти.

Достоинства логистической регрессии.

- Возможность оценить вероятность принадлежности классифицируемого объекта каждому из классов.
- Возможность представить классификатор в виде понятной экспертам скоринговой карты.
- Настройка вектора весов производится путём многократного применения метода наименьших квадратов, который хорошо изучен, имеет массу стандартных реализаций и допускает всевозможные обобщения, в том числе отбор признаков, регуляризацию, ограничение неотрицательности весов.

Недостатки логистической регрессии.

- Метод логистической регрессии наследует все недостатки многомерной линейной регрессии, перечисленные в §1.3. Практичная реализация должна предусматривать стандартизацию данных, отсев выбросов, регуляризацию весов, отбор признаков.
- Относительно низкая эффективность. Трудоёмкость метода в T раз превышает трудоёмкость обычного МНК, где T — число итераций.
- Оценки вероятностей могут оказаться неадекватными, если не выполняются предположения Теоремы 1.9.

Список литературы

[1] *Лоусон Ч., Хенсон Р.* Численное решение задач метода наименьших квадратов. — М.: Наука, 1986.

[2] *Хардле В.* Прикладная непараметрическая регрессия. — М.: Мир, 1993.

[3] *Cleveland W. S.* Robust locally weighted regression and smoothing scatter plots // *Journal of the American Statistical Association.* — 1979. — Vol. 74, no. 368. — Pp. 829–836.

[4] *Hastie T., Tibshirani R.* Generalized additive models // *Statistical Science.* — 1986. — Vol. 1. — Pp. 297–318.
http://citeseer.ist.psu.edu/hastie95generalized.html.

[5] *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning. — Springer, 2001.

[6] *Tibshirani R. J.* Regression shrinkage and selection via the lasso // *Journal of the Royal Statistical Society. Series B (Methodological).* — 1996. — Vol. 58, no. 1. — Pp. 267–288.
http://citeseer.ist.psu.edu/tibshirani94regression.html.