

Лекции по алгоритмам восстановления регрессии

К. В. Воронцов

19 января 2009 г.

Материал находится в стадии разработки, может содержать ошибки и неточности. Автор будет благодарен за любые замечания и предложения, направленные по адресу vokov@forecsys.ru, либо высказанные в обсуждении страницы «Машинное обучение (курс лекций, К.В.Воронцов)» вики-ресурса www.MachineLearning.ru.

Перепечатка фрагментов данного материала без согласия автора является плагиатом.

Содержание

1	Восстановление регрессии	2
1.1	Метод наименьших квадратов	2
1.2	Непараметрическая регрессия: ядерное сглаживание	3
1.2.1	Метод наименьших квадратов и формула Надарая–Ватсона	3
1.2.2	Выбор ядра и ширины окна	4
1.2.3	Проблема выбросов: робастная непараметрическая регрессия	5
1.2.4	Проблема краевых эффектов	6
1.3	Многомерная линейная регрессия	7
1.3.1	Нормальная система уравнений	8
1.3.2	Сингулярное разложение	9
1.3.3	Стандартизация данных	10
1.3.4	Проблема мультиколлинеарности	10
1.3.5	Гребневая регрессия (регуляризация)	11
1.3.6	Лассо Тибширани	13
1.3.7	Линейная монотонная регрессия	14
1.3.8	Метод главных компонент	14
1.3.9	Метод ортогонализации Грама–Шмидта	18
1.3.10	Робастная регрессия	23
1.4	Нелинейные обобщения линейной регрессии	24
1.4.1	Нелинейная модель регрессии	24
1.4.2	Нелинейные одномерные преобразования признаков	25
1.4.3	Обобщённые линейные модели	26
1.4.4	Неквадратичные функции потерь	27
1.4.5	Логистическая регрессия и итерационный взвешенный МНК	29
1.5	Метод опорных векторов в задачах регрессии	30

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 $y^*: X \rightarrow Y$, 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 $a: X \rightarrow Y$, 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:

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

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 :

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

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:

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

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

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

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 $(\varepsilon_1, \dots, \varepsilon_\ell)$ с независимыми компонентами:

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

Запишем логарифм функции правдоподобия:

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

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*:

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

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:

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

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 $K: [0, \infty) \rightarrow [0, \infty)$, 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:

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

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 $\rho(x, x_i) = |x - x_i|$. 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 $\lim_{r \rightarrow \infty} rK(r) = 0$;
- 3) восстанавливаемая зависимость, определяемая плотностью $p(y|x)$, удовлетворяет при любом $x \in X$ ограничению $\mathbb{E}(y^2|x) = \int_Y y^2 p(y|x) dy < \infty$;
- 4) последовательность h_ℓ такова, что $\lim_{\ell \rightarrow \infty} h_\ell = 0$ и $\lim_{\ell \rightarrow \infty} \ell h_\ell = \infty$.

Тогда имеет место сходимость по вероятности: $a_{h_\ell}(x; X^\ell) \xrightarrow{P} \mathbb{E}(y|x)$ в любой точке $x \in X$, в которой $\mathbb{E}(y|x)$, $p(x)$ и $\mathbb{D}(y|x)$ 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 $K_G(r) = \exp(-\frac{1}{2}r^2)$ или кватрическое $K_Q(r) = (1 - r^2)^2 [r < 1]$.

Если ядро $K(r)$ финитно, то есть $K(r) = 0$ при $r \geq 1$, то ненулевые веса получают 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 ($h \rightarrow 0$) 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 $h \rightarrow \infty$ 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 $w_i(x) = K\left(\frac{\rho(x, x_i)}{h(x)}\right)$.

Самый простой способ — взять в качестве ширины окна $h(x)$ расстояние от объекта x до его $k + 1$ -го соседа: $h_k(x) = \rho(x, x_x^{(k+1)})$. Недостаток этого способа в том, что функция $h_k(x)$ является непрерывной, но не гладкой, поэтому функции $w_i(x)$ и $a_{h_k}(x)$ имеют разрывные первые производные, даже если ядро гладкое. Для устранения этого недостатка можно сгладить саму функцию $h_k(x)$ по точкам $(x_i, h_k(x_i))$, $i = 1, \dots, \ell$, взяв какое-либо достаточно гладкое ядро, скажем, K_Q или K_G .

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 $h \rightarrow 0$. Такой способ оценивания называется скользящим контролем *с исключением объектов по одному* (leave-one-out, LOO):

$$\text{LOO}(h, X^\ell) = \sum_{i=1}^{\ell} (a_h(x_i; X^\ell \setminus \{x_i\}) - y_i)^2 \rightarrow \min_h,$$

где минимизация осуществляется по ширине окна h или по числу соседей k .

1.2.3 Проблема выбросов: робастная непараметрическая регрессия

Оценка Надарай–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

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

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 γ_i , $i = 1, \dots, \ell$;

1: инициализация: $\gamma_i := 1$, $i = 1, \dots, \ell$;

2: **повторять**

3: вычислить оценки скользящего контроля на каждом объекте:

$$a_i := a_h(x_i; X^\ell \setminus \{x_i\}) = \frac{\sum_{j=1, j \neq i}^{\ell} y_j \gamma_j K\left(\frac{\rho(x_i, x_j)}{h(x_i)}\right)}{\sum_{j=1, j \neq i}^{\ell} \gamma_j K\left(\frac{\rho(x_i, x_j)}{h(x_i)}\right)}, \quad i = 1, \dots, \ell$$

4: вычислить коэффициенты γ_i :

$$\gamma_i := \tilde{K}(|a_i - y_i|); \quad i = 1, \dots, \ell;$$

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 $\varepsilon^{(1)} \leq \dots \leq \varepsilon^{(\ell)}$, и отбрасывается некоторое количество t объектов с наибольшей ошибкой. Это соответствует ядру $\tilde{K}(\varepsilon) = [\varepsilon \leq \varepsilon^{(\ell-t)}]$.

Мягкая фильтрация [3]: используется кватрическое ядро $\tilde{K}(\varepsilon) = K_Q\left(\frac{\varepsilon}{6 \operatorname{med}\{\varepsilon_i\}}\right)$, где $\operatorname{med}\{\varepsilon_i\}$ — 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:

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

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:

$$a_h(x; X^\ell) = \frac{\sum_{i=1}^{\ell} w_i d_i^2 \sum_{i=1}^{\ell} w_i y_i - \sum_{i=1}^{\ell} w_i d_i \sum_{i=1}^{\ell} w_i d_i y_i}{\sum_{i=1}^{\ell} w_i \sum_{i=1}^{\ell} w_i d_i^2 - \left(\sum_{i=1}^{\ell} w_i d_i \right)^2}.$$

В многомерном случае $X = \mathbb{R}^n$ для вычисления коэффициентов в линейной форме $a(u) = \alpha^\top(u - x) + \beta$ 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 $f_j(x)$, $j = 1, \dots, n$. 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 :

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

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 $F' = WF$, $y' = Wy$:

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

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*:

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

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, \dots, f_n подвергаются некоторым преобразованиям $\varphi_1, \dots, \varphi_k$, в общем случае нелинейным:

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

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

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

1.3.1 Нормальная система уравнений

Запишем необходимое условие минимума (1.3) в матричном виде:

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

откуда следует

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

Эта система линейных уравнений относительно α называется *нормальной системой* для задачи наименьших квадратов. Матрица $F^T F$ имеет размер $n \times n$ и совпадает с ковариационной матрицей набора признаков f_1, \dots, f_n . Если она невырождена, то решением системы является вектор

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

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

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

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

Решение имеет простую геометрическую интерпретацию. Произведение $P_F y$ есть проекция целевого вектора y на линейную оболочку столбцов матрицы F . Разность $(P_F y - y)$ есть проекция целевого вектора y на ортогональное дополнение этой линейной оболочки. Значение функционала $Q(\alpha^*) = \|P_F y - y\|^2$ есть 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

Произвольную $\ell \times n$ -матрицу можно представить в виде *сингулярного разложения* (singular value decomposition, SVD)

$$F = VDU^T,$$

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

1) $\ell \times n$ -матрица $V = (v_1, \dots, v_n)$ ортогональна, $V^T V = I_n$, столбцы v_j являются собственными векторами матрицы FF^T ;

2) $n \times n$ -матрица $U = (u_1, \dots, u_n)$ ортогональна, $U^T U = I_n$, столбцы u_j являются собственными векторами матрицы $F^T F$;

3) $n \times n$ -матрица D диагональна, $D = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$, где $\lambda_1, \dots, \lambda_n$ — общие собственные значения матриц $F^T F$ и FF^T ; число ненулевых собственных значений среди $\lambda_1, \dots, \lambda_n$ равно рангу матриц F , $F^T F$ и FF^T .

Имея сингулярное разложение, легко записать псевдообратную матрицу:

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

вектор МНК-решения:

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

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

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

и норму вектора коэффициентов:

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

Формула (1.5) означает, что МНК-аппроксимация $F\alpha^*$ есть проекция целевого вектора y на линейную оболочку собственных векторов матрицы FF^T .

Кажущаяся простота решения (1.5) компенсируется тем, что вычисление сингулярного разложения столь же трудоёмко, как и обращение матрицы $F^T F$. 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 $F = (f_{ij})$:

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

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 $F^T F$ является ковариационной матрицей, а после стандартизации — корреляционной.

1.3.4 Проблема мультиколлинеарности

Если ковариационная матрица $\Sigma = F^T F$ 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

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

где λ_{\max} и λ_{\min} — максимальное и минимальное собственные значения матрицы Σ , все нормы евклидовы. Матрицу принято считать плохо обусловленной, если $\mu(\Sigma)$ превышает $10^3 \dots 10^6$. 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

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

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 (регуляризация)

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. Приравнявая нулю производную $Q_\tau(\alpha)$ по параметру α , находим:

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

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 $F^T F$ 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:

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

Теперь найдём регуляризованную МНК-аппроксимацию целевого вектора y :

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

Как и прежде в (1.5), МНК-аппроксимация представляется в виде разложения целевого вектора y по базису собственных векторов матрицы FF^\top . 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:

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

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 — это второй смысл термина «сжатие».

Можно показать, что роль размерности играет след проекционной матрицы. Действительно, в нерегуляризованном случае имеем

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

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

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

При $\tau \rightarrow 0$ регуляризованное решение стремится к МНК-решению: $\alpha_\tau^* \rightarrow \alpha^*$. При $\tau \rightarrow \infty$ чрезмерная регуляризации приводит к вырожденному решению: $\alpha_\tau^* \rightarrow 0$. 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 :

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

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:

$$\begin{cases} Q(\alpha) = \|F\alpha - y\|^2 \rightarrow \min_{\alpha}; \\ \sum_{j=1}^n |\alpha_j| \leq \varkappa; \end{cases} \quad (1.9)$$

где \varkappa — параметр регуляризации. При больших значениях \varkappa ограничение (1.9) становится строгим неравенством, и решение совпадает с методом наименьших квадратов. Чем меньше \varkappa , тем больше коэффициентов α_j обнулятся, тем меньше признаков войдут в уравнение регрессии. Происходит отбор (селекция) признаков, поэтому параметр \varkappa называют ещё *селективностью*. Образно говоря, параметр \varkappa зажимает вектор коэффициентов, лишая его избыточных степеней свободы. Отсюда и название метода — *лассо* (LASSO, least absolute shrinkage and selection operator) [7].

Чтобы понять, почему лассо осуществляет отбор признаков, приведём задачу квадратичного программирования (1.9) к каноническому виду. Заменяем каждую переменную α_j разностью двух новых неотрицательных переменных: $\alpha_j = \alpha_j^+ - \alpha_j^-$. Функционал Q останется квадратичным по новым переменным, ограничение (1.9) примет линейный вид, и добавится $2n$ ограничений-неравенств:

$$\sum_{j=1}^n \alpha_j^+ + \alpha_j^- \leq \varkappa; \quad \alpha_j^+ \geq 0; \quad \alpha_j^- \geq 0.$$

Чем меньше \varkappa , тем больше ограничений обращаются в 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 должен быть отклик y ; иными словами, признак f_j и отклик y имеют положительную корреляцию.

В частности, при построении линейной композиции алгоритмов роль признаков играют базовые алгоритмы (см. главу ??). Естественно полагать, что если базовые алгоритмы и композиция настраиваются на один и тот же целевой вектор y , то они должны иметь положительную корреляцию. Если же базовый алгоритм настолько ненадёжен, что при настройке ему подбирается отрицательный вес, то лучше вовсе убрать этот алгоритм из композиции.

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

$$\begin{cases} Q(\alpha) = \|F\alpha - y\|^2 \rightarrow \min_{\alpha}; \\ \alpha_j \geq 0; \quad j = 1, \dots, n. \end{cases}$$

Метод решения данной задачи основан на применении теоремы Куна-Таккера. Когда ограничение $\alpha_j \geq 0$ становится активным, то есть обращается в равенство, признак f_j , фактически, исключается из уравнения регрессии. В линейной композиции это соответствует исключению j -го базового алгоритма из композиции.

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

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

Обозначим через $z_i = (g_1(x_i), \dots, g_m(x_i))$ признаковые описания тех же объектов в новом пространстве $Z = \mathbb{R}^m$ меньшей размерности, $m < n$:

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

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

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

или в векторной записи: $\hat{x} = zU^\top$. 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:

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

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

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

Исчерпывающее решение задачи (1.10) даёт следующая теорема.

Теорема 1.3. *Если $m \leq \text{rk} F$, то минимум $\Delta^2(G, U)$ достигается, когда столбцы матрицы U есть собственные векторы $F^\top F$, соответствующие m максимальным собственным значениям. При этом $G = FU$, матрицы U и G ортогональны.*

Доказательство.

Запишем необходимые условия минимума:

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

Поскольку искомые матрицы G и U невырождены, отсюда следует

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

Функционал $\Delta^2(G, U)$ зависит только от произведения матриц GU^\top , поэтому решение задачи (1.10) определено с точностью до произвольного невырожденного преобразования R : $GU^\top = (GR)(R^{-1}U^\top)$. Распорядимся свободой выбора R так, чтобы матрицы $U^\top U$ и $G^\top G$ оказались диагональными. Покажем, что это всегда возможно.

Пусть $\tilde{G}\tilde{U}^\top$ — произвольное решение задачи (1.10).

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

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

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

$$\begin{aligned} G^\top G &= T^\top(S^\top\tilde{G}^\top\tilde{G}S)T = \Lambda; \\ U^\top U &= T^{-1}(S^{-1}\tilde{U}^\top\tilde{U}S^{-1\top})T^{-1\top} = (T^\top T)^{-1} = I_m. \end{aligned}$$

В силу $GU^\top = \tilde{G}\tilde{U}^\top$ матрицы G и U являются решением задачи (1.10) и удовлетворяют необходимому условию минимума. Подставим матрицы G и U в (1.11). Благодаря диагональности $G^\top G$ и $U^\top U$ соотношения существенно упростятся:

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

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

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

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

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

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

Собственные векторы u_1, \dots, 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 $F = GU^\top$ является точным и совпадает с сингулярным разложением: $F = GU^\top = VDU^\top$, если положить $G = VD$ и $\Lambda = D^2$. При этом матрица V ортогональна: $V^\top V = I_m$. Остальные свойства сингулярного разложения, перечисленные на стр. 9, непосредственно вытекают из Теоремы 1.3.

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

Задача наименьших квадратов в новом признаковом пространстве имеет вид

$$\|G\beta - y\|^2 \rightarrow \min_{\beta}.$$

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

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

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

$$\begin{aligned}\beta^* &= \Lambda^{-1} G^T y = D^{-1} V^T y; \\ G\beta^* &= V D \beta^* = V V^T y.\end{aligned}$$

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

Эффективная размерность выборки. Главные компоненты содержат основную информацию о матрице F . Если точность приближения $F \approx GU^T$ 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 $F^T F$ упорядочиваются по убыванию: $\lambda_1 \geq \dots \geq \lambda_n \geq 0$. 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 :

$$E(m) = \frac{\|GU^T - F\|^2}{\|F\|^2} = \frac{\lambda_{m+1} + \dots + \lambda_n}{\lambda_1 + \dots + \lambda_n} \leq \varepsilon.$$

Величина $E(m)$ показывает, какая доля информации теряется при замене исходных признаков 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 мало.

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 $(g_1(x_i), g_2(x_i))$, $i = 1, \dots, \ell$, 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, $G^T G = I_n$. Для любой матрицы F существует бесконечно много разложений указанного вида. Имея одно из них, легко выразить псевдообратную матрицу F^+ через G и R :

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

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

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

$$\begin{aligned} F &= (f_1, \dots, f_n); \\ G &= (\tilde{g}_1, \dots, \tilde{g}_n). \end{aligned}$$

Волной здесь и далее обозначается операция нормирования вектора:

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

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

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

Легко проверить, что для всех k, j из $\{1, \dots, n\}$, $k \neq j$, 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, $j = 1, \dots, n$, матрица $F_j = (f_1, \dots, f_j)$ представима в виде ортогонального разложения $F_j = G_j R_j$, где

$$\begin{aligned} G_j &= (\tilde{g}_1, \dots, \tilde{g}_j) \text{ — ортонормированная матрица;} \\ R_j &= \begin{pmatrix} r_{11} & \dots & r_{1j} \\ & \ddots & \vdots \\ 0 & & r_{jj} \end{pmatrix} \text{ — верхняя треугольная матрица, } r_{ij} = \begin{cases} \tilde{g}_i^T f_j, & i < j; \\ \|g_j\|, & i = j. \end{cases} \end{aligned}$$

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

$$R_1 = (r_{11});$$

$$R_j = \begin{pmatrix} R_{j-1} & r_j \\ 0 & r_{jj} \end{pmatrix}, \quad j = 2, \dots, n,$$

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

$$T_1 = (t_{11});$$

$$T_j = \begin{pmatrix} T_{j-1} & t_j \\ 0 & t_{jj} \end{pmatrix}, \quad j = 2, \dots, n,$$

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 $\{f_1, \dots, f_{j-1}\}$. 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: $r_{jj} < \delta \max_{i < j} r_{ii}$, где δ имеет порядок $10^{-2} \dots 10^{-5}$.

Назовём вектор $\alpha_j = F_j^+$ у 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

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

Доказательство.

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

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

откуда следует требуемая рекуррентная формула. ■

Назовём величину $Q_j = \min_{\alpha} \|y - F_j \alpha\|^2 = \|y - F_j \alpha_j\|^2$ 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

$$\begin{aligned}Q_0 &= \|y\|^2; \\ Q_j &= Q_{j-1} - (y^\top \tilde{g}_j)^2, \quad j = 1, \dots, n.\end{aligned}$$

Доказательство.

Воспользуемся разложением псевдообратной матрицы $F_j^+ = T_j G_j^\top$:

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

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

$$\begin{aligned}\varepsilon_0 &= y; \\ \varepsilon_j &= \varepsilon_{j-1} - \tilde{g}_j (y^\top \tilde{g}_j), \quad j = 1, \dots, n.\end{aligned}$$

Алгоритм 1.2. Ортогонализация Грама-Шмидта.

- 1: инициализация: $g_j := f_j, j := 1, \dots, n$;
 - 2: **для** $j := 1, \dots, n$
 - 3: **для** $i := 1, \dots, j - 1$
 - 4: $r_{ij} := \tilde{g}_i^\top g_j$; (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: $r_{jj} := \|g_j\|$;
-

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

$$\tilde{g}_i^\top g_j = \tilde{g}_i^\top \left(f_j - \sum_{s=1}^{i-1} \tilde{g}_s r_{sj} \right) = \tilde{g}_i^\top f_j - \sum_{s=1}^{i-1} \underbrace{\tilde{g}_i^\top \tilde{g}_s}_0 r_{sj} = \tilde{g}_i^\top f_j.$$

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 , см. Алгоритм 1.2.

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

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

Тогда в начале j -го шага все столбцы g_j, \dots, g_n по построению будут ортогональны всем столбцам g_1, \dots, 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 $g_m \in \{g_j, \dots, g_n\}$, 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 $\|g_m\| \rightarrow \max_m$, что соответствует выбору столбца f_m , максимально некоррелированного с g_1, \dots, 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: $\frac{y^\top g_m}{\|g_m\|} \rightarrow \max_m$; 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 $X^k = \{x'_1, \dots, x'_k\}$ окажется минимальным, что соответствует применению скользящего контроля (hold-out CV).

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

Вход:

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

Выход:

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

1: инициализация:

$$Q_0 := \|y\|^2; \quad g_j := f_j; \quad Z_j := \|g_j\|^2; \quad D_j := y^\top g_j; \quad j := 1, \dots, n;$$

2: **для** $j := 1, \dots, n$

3: выбор $m \in \{j, \dots, n\}$ по критериям $Z_m \rightarrow \max_m$ и $(D_m^2/Z_m) \rightarrow \max_m$;

4: перестановка местами столбцов:

$$g_j \Leftrightarrow g_m, \quad f_j \Leftrightarrow f_m, \quad r_j \Leftrightarrow r_m;$$

5: $r_{jj} := \sqrt{Z_m}$; нормировка: $\tilde{g}_j := g_j/r_{jj}$;

6: вычисление текущего значения функционала:

$$d_j := D_j/r_{jj}; \quad (\text{эффективное вычисление } d_j := y^\top \tilde{g}_j);$$

$$Q_j := Q_{j-1} - d_j^2;$$

7: обращение верхней треугольной матрицы $T_j = R_j^{-1}$:

$$t_{jj} := 1/r_{jj}; \quad t_j := -t_{jj}T_{j-1}r_j; \quad (\text{вектор-столбец } t_j \text{ длины } j-1);$$

$$T_j := \begin{pmatrix} T_{j-1} & t_j \\ 0 & t_{jj} \end{pmatrix};$$

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** $i := j+1, \dots, n$

12: $r_{ji} := \tilde{g}_j^\top g_i$; (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 $D_i = y^\top g_i$);

16: **конец** цикла по j .

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 поочередно 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 осуществляет *отбор информативных признаков* (features selection). Шаги 11–15 ре-

ализуют 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 $D_j = y^T g_j$, 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 большими 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: $\mathcal{L}(a, y) = 1 - \exp\left(-\frac{1}{2\sigma}(a - y)^2\right)$, 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 $\alpha \in \mathbb{R}^p$:

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

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

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

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 градиента:

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

Запишем компоненты гессиана:

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

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 :

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

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

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

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

В правой части записано решение стандартной задачи многомерной линейной регрессии $\|F_t \delta - (f_t - y)\|^2 \rightarrow \min_{\delta}$. 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

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

где $\varphi_j: \mathbb{R} \rightarrow \mathbb{R}$ — некоторые преобразования исходных признаков, в общем случае нелинейные. Задача состоит в том, чтобы подобрать неизвестные одномерные преобразования φ_j , при которых достигается минимум квадратичного функционала (1.1).

Метод настройки с возвращениями предложен Хасти и Тибширани в 1986 году [4]. Схема реализации показана в Алгоритме 1.4.

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

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

Алгоритм 1.4. Метод настройки с возвращениями (backfitting)

Вход:

F, y — матрица «объекты–признаки» и вектор ответов;

Выход:

α — вектор коэффициентов линейной комбинации.

- 1: нулевое приближение: $\alpha :=$ решение задачи МЛР с признаками $f_j(x)$;
 - 2: **повторять**
 - 3: **для** $j = 1, \dots, n$
 - 4: $z_i := y_i - \sum_{k=1, k \neq j}^n \varphi_k(f_k(x_i)), i = 1, \dots, \ell;$
 - 5: $\varphi_j := \arg \min_{\varphi} \sum_{i=1}^{\ell} (\varphi_j(f_j(x)) - z_i)^2;$
 - 6: **пока** α не сойдётся
-

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

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 $\Phi(z) = P_x \{ \varphi_j(f_j(x)) < z \}$, 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 $f_j(x_i)$, $i = 1, \dots, \ell$ упорядочиваются по возрастанию, образуя вариационный ряд $f_j^{(1)} < \dots < f_j^{(\ell)}$. 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 $Z_j^\ell = (f_j^{(i)}, \Phi_0^{-1}(\frac{i-0.5}{\ell}))_{i=1}^{\ell}$, 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:

$$f_j^{(i)} < f_j^{(i')} \Leftrightarrow i < i' \Leftrightarrow \Phi_0^{-1}(\frac{i-0.5}{\ell}) < \Phi_0^{-1}(\frac{i'-0.5}{\ell}).$$

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 призна-

ком y . Задача аппроксимации ставится, исходя из принципа наименьших квадратов:

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

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 :

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

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 $\mathcal{L}(a, y)$ 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:

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

Если функция потерь квадратична, $\mathcal{L}(a, y) = (a - y)^2$, 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: $\mathcal{L}(a, y) = c_1|a - y|$, где c_1 — коэффициент торговой наценки. Потери от завышенного прогноза $a > y$ связаны с замораживанием средств, затовариванием склада, а в худшем случае — с истечением срока годности и списанием товара. В первом приближении эти потери также прямо пропорциональны отклонению, но с другим коэффициентом: $\mathcal{L}(a, y) = c_2|a - y|$. Коэффициенты c_1 и c_2 известны для каждого магазина и каждого товара. Таким образом, в данной задаче более обоснованной оказывается не квадратичная, а кусочно-линейная несимметричная функция потерь.

Пример 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»), и тут же принимаем следующее решение согласно алгоритму $a(x)$, и так далее. Суммарная прибыль, заработанная в течение ℓ последовательных моментов времени x_1, \dots, x_ℓ , равна

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

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

Робастная регрессия 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.

1.4.5 Логистическая регрессия и итерационный взвешенный МНК

Неквадратичные функции потерь используются в методах классификации, в частности, в *логистической регрессии*, см. ???. Там минимизируемый функционал имеет вид

$$\tilde{Q}(w, X^\ell) = \sum_{i=1}^{\ell} \ln(1 + \exp(-w^\top x_i y_i)) = - \sum_{i=1}^{\ell} \ln \sigma(w^\top x_i y_i) \rightarrow \min_w,$$

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

Стандартная техника настройки параметров w заключается в применении метода Ньютона-Рафсона для минимизации нелинейного функционала $Q(w)$. В качестве нулевого приближения можно взять «наивное» решение задачи классификации как задачи многомерной линейной регрессии, в которой ответы принимают только два значения, $y_i \in \{-1, +1\}$. Затем начинается итерационный процесс, на t -м шаге которого уточняется вектор коэффициентов $w^{(t+1)}$:

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

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

Найдём выражения для градиента и гессиана.

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

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

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

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

$$\begin{aligned} \frac{\partial^2 Q(w)}{\partial w_j \partial w_k} &= - \frac{\partial}{\partial w_k} \sum_{i=1}^{\ell} (1 - \sigma_i) y_i f_j(x_i) = \\ &= \sum_{i=1}^{\ell} (1 - \sigma_i) \sigma_i f_j(x_i) f_k(x_i), \quad j = 1, \dots, n, \quad k = 1, \dots, n. \end{aligned}$$

Введём матричные обозначения:

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

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

$\tilde{F} = \Gamma F$ — взвешенная матрица признаков 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:

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

Алгоритм 1.5. IRLS — итерационный взвешенный метод наименьших квадратов

Вход:

F, y — матрица «объекты–признаки» и вектор ответов;

Выход:

w — вектор коэффициентов линейной комбинации.

1: нулевое приближение — обычный МНК:

$$w := (F^T F)^{-1} F^T y;$$

2: **для** $t := 1, 2, 3, \dots$

3: $z := Fw$;

4: $\gamma_i := \sqrt{(1 - \sigma(z_i))\sigma(z_i)}$ для всех $i = 1, \dots, \ell$;

5: $\tilde{F} := \text{diag}(\gamma_1, \dots, \gamma_\ell)F$;

6: $\tilde{y}_i := y_i \sqrt{(1 - \sigma(z_i))/\sigma(z_i)}$ для всех $i = 1, \dots, \ell$;

7: выбрать градиентный шаг h_t ;

8: $w := w + h_t (\tilde{F}^T \tilde{F})^{-1} \tilde{F}^T \tilde{y}$;

9: **если** $\sigma(z_i)$ мало изменились относительно предыдущей итерации **то**

10: прервать итерации, выйти из цикла;

11: **конец** цикла по t .

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

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

Таким образом, решение задачи классификации сводится к последовательности регрессионных задач, для каждой из которых веса объектов и ответы пересчитываются заново. Отсюда и название — метод *наименьших квадратов с итерационным пересчетом* (iteratively reweighted least squares, IRLS)

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

§1.5 Метод опорных векторов в задачах регрессии

В главе 1 мы уже рассматривали задачи многомерной линейной регрессии, предполагая, что $X = \mathbb{R}^n$, $Y = \mathbb{R}$, алгоритм имеет вид $a(x) = \langle w, x \rangle - w_0$, и для настройки параметров $w \in \mathbb{R}^n$ и $w_0 \in \mathbb{R}$ минимизируется квадратичный функционал. В случае гребневой регрессии (см. раздел 1.3.5) вводится ещё и штрафное слагаемое,

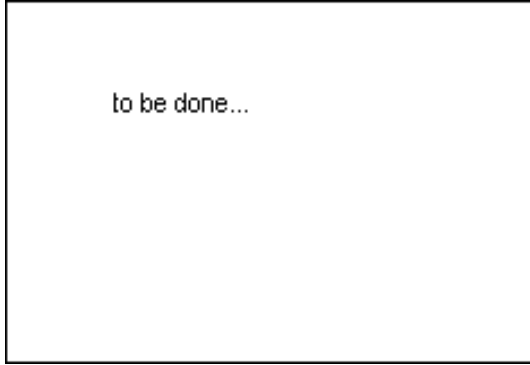


Рис. 1. Сигмоидная функция переводит значение дискриминантной функции в вероятность.

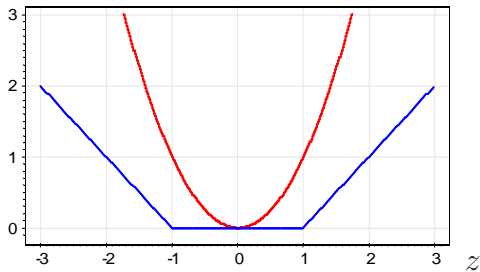


Рис. 2. Функции потерь в задачах регрессии: кусочно-линейная $|z|_\varepsilon$ при $\varepsilon = 1$ и квадратичная z^2 .

предотвращающее бесконтрольное увеличение коэффициентов w :

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

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

Однако в некоторых случаях более естественно использовать кусочно-линейную функцию ε -чувствительности, показанную на Рис 2: $|z|_\varepsilon = \max\{0, |z| - \varepsilon\}$, которая не считает за ошибки отклонения $a(x_i)$ от y_i , меньшие ε . Предполагается, что значение параметра ε задаёт эксперт, исходя из априорных соображений.

С этой функцией потерь функционал принимает вид

$$Q_\varepsilon(a, X^\ell) = \sum_{i=1}^{\ell} |\langle w, x_i \rangle - w_0 - y_i|_\varepsilon + \tau \langle w, w \rangle \rightarrow \min_{w, w_0}. \quad (1.13)$$

Легко обнаруживается сходство данной задачи с задачей классификации (??). Покажем, что минимизация (1.13) эквивалентна некоторой задаче квадратичного программирования с линейными ограничениями типа неравенств. При этом также возникает двойственная задача, зависящая только от двойственных переменных; также достаточно оставить в выборке только опорные объекты; также решение выражается через скалярные произведения объектов, а не сами объекты; и также можно использовать ядра. Иными словами, SVM-регрессия отличается от SVM-классификации только в технических деталях, основные идеи остаются теми же.

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

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

Тогда задача минимизации (1.13) может быть переписана в эквивалентной форме как задача квадратичного программирования с линейными ограничениями-неравенствами относительно переменных w_i , w_0 , ξ_i^+ и ξ_i^- :

$$\begin{cases} \frac{1}{2} \langle w, w \rangle + C \sum_{i=1}^{\ell} (\xi_i^+ + \xi_i^-) \rightarrow \min_{w, w_0, \xi_i^+, \xi_i^-}; \\ y_i - \varepsilon - \xi_i^- \leq \langle w, x_i \rangle - w_0 \leq y_i + \varepsilon + \xi_i^+, \quad i = 1, \dots, \ell; \\ \xi_i^- \geq 0, \quad \xi_i^+ \geq 0, \quad i = 1, \dots, \ell. \end{cases} \quad (1.14)$$

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

$$\begin{cases} \mathcal{L}(\lambda^+, \lambda^-) = -\varepsilon \sum_{i=1}^{\ell} (\lambda_i^- + \lambda_i^+) + \sum_{i=1}^{\ell} (\lambda_i^- - \lambda_i^+) y_i - \\ \quad - \frac{1}{2} \sum_{i,j=1}^{\ell} (\lambda_i^- - \lambda_i^+) (\lambda_j^- - \lambda_j^+) K(x_i, x_j) \rightarrow \max_{\lambda^+, \lambda^-}; \\ 0 \leq \lambda_i^+ \leq C, \quad 0 \leq \lambda_i^- \leq C, \quad i = 1, \dots, \ell; \\ \sum_{i=1}^{\ell} (\lambda_i^- + \lambda_i^+) = 0. \end{cases}$$

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

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

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

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

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

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

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

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

Уравнение регрессии также выражается через двойственные переменные:

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

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

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

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

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

Более обстоятельное изложение многочисленных особенностей SVM-регрессии можно найти в руководстве [6].

Резюме

1. *Задача восстановления регрессии* отличается от задачи классификации тем, что ответы y — действительные числа. Основным инструментом настройки регрессионных зависимостей является *метод наименьших квадратов*, МНК. Однако надо помнить, что МНК неявно соответствует предположению о нормальном распределении ошибок; если это предположение недопустимо, то квадратичная функция потерь должна быть заменена на другую.
2. *Непараметрическая регрессия* основана на локальном МНК, который приводит к ядерному сглаживанию Надарая-Ватсона. Выбор ширины окна сглаживания решающим образом влияет на качество восстановления регрессионной зависимости. Выбор ядра сглаживания слабо влияет на качество, но определяет степень гладкости аппроксимирующей функции.
3. *Робастная непараметрическая регрессия* устойчива к редким большим выбросам. Итерационный алгоритм локально взвешенного сглаживания LOWESS основан на том, что объекты с большими ошибками получают меньший вес.
4. В *многомерной линейной регрессии* МНК приводит к минимизации квадратичного функционала $\|F\alpha - y\|^2$ по вектору коэффициентов α . Чтобы получить решение, требуется вычислить обратную матрицу $(F^T F)^{-1}$. Для этого можно строить *сингулярное разложение* матрицы F , модифицированную ортогонализацию Грама-Шмидта или другие ортогональные разложения.
5. *Проблема мультиколлинеарности* возникает из-за недостаточного объёма выборки или наличия скрытых зависимостей между признаками. Мультиколлинеарность приводит к неустойчивости МНК-решения: увеличивается разброс коэффициентов α , понижается обобщающая способность. Для устранения мультиколлинеарности применяются различные методы сокращения размерности: регуляризация, преобразование признаков, отбор признаков.
6. *Гребневая регрессия* является частным случаем регуляризации. К квадратичному функционалу добавляется штрафное слагаемое $\tau\|\alpha\|^2$, и задача сводится к вычислению другой обратной матрицы $(F^T F + \tau I)^{-1}$. Эту модификацию особенно просто выполнить с помощью сингулярного разложения. Неотрицательный параметр регуляризации τ определяющим образом влияет на обобщающую способность. Чем больше τ , тем ниже эффективная размерность решения. На практике τ подбирают по скользящему контролю, причём сингулярное разложение требуется вычислить только один раз.
7. *Лассо Тибширани* — это ещё один способ регуляризации, но теперь ограничение накладывается на сумму модулей коэффициентов: $\sum_{j=1}^n |\alpha_j| \leq \kappa$. Это приводит к обнулению части коэффициентов α_j и отбору признаков. Чем меньше κ , тем меньше признаков остаётся в уравнении регрессии. На практике κ также подбирают по скользящему контролю, причём существует эффективный метод, вычисляющий α одновременно для последовательности значений κ .
8. *Метод главных компонент* тесно связан с сингулярным разложением. Он находит такое линейное преобразование исходного n -мерного пространства призна-

ков в m -мерное, $m < n$, чтобы матрица F представлялась в нём с наименьшими потерями. Для этого достаточно взять m наибольших собственных значений (и соответствующих им собственных векторов) матрицы $F^T F$.

9. *Метод модифицированной ортогонализации Грама-Шмидта* позволяет реализовать отбор признаков на основе «жадного» добавления. При этом в метод легко встроить несколько различных критериев, оценивающих полезность добавляемых признаков, в том числе по скользящему контролю.
10. *Нелинейная регрессия* строится с помощью итерационного процесса, на каждом шаге которого задача линеаризуется методом Ньютона-Гаусса и сводится к линейной регрессионной задаче.
11. *Метод настройки с возвращениями* позволяет подобрать одномерные нелинейные преобразования признаков в линейной регрессионной модели. Он заключается в итерационном повторении двух шагов: одномерного сглаживания одного из признаков и многомерной линейной регрессии по всем признакам.
12. *SVM-регрессия* эквивалентна минимизации регуляризованного эмпирического риска, если вместо квадратичной функции потерь взять кусочно-линейную. Как и в случае SVM-классификации, возникает задача квадратичного программирования с ограничениями-неравенствами. Она имеет единственное решение, и оно также выражается через опорные векторы.

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] Smola A., Schoelkopf B. A tutorial on support vector regression: Tech. Rep. NeuroCOLT2 NC2-TR-1998-030: 1998.
<http://citeseer.ist.psu.edu/smola98tutorial.html>.
- [7] 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>.