ЦИФРОВАЯ КУЛЬТУРА
УНИВЕРСИТЕТ ИТМО

Логистическая регрессия

Высшая Школа Цифровой Культуры
Университет ИТМО
dc@itmo.ru

Содержание

1 Логистическая регрессия

1.1 Введение. Генеративные и дискриминативные алгоритмы.

Здравствуйте, уважаемые слушатели. В этой лекции мы продолжим изучение методов решения задачи классификации, а также познакомимся с еще одним способом оценки качества алгоритма – с так называемым ROC анализом.

Метод двухклассовой классификации, с описания которого мы начнем нашу лекцию, называется логистической регрессией. Внимательный слушатель, наверное, сразу встанет в ступор: при чем здесь регрессия? Ведь регрессия – это одна задача, а классификация – совсем другая. Регрессия – это задача предсказания числа (точнее – какой-то непрерывной переменной), а классификация – задача предсказания класса (какой-то переменной, принимающей, обычно, конечное число значений). Давайте проясним такую несогласованность. На самом деле алгоритм двухклассовой логистической регрессии выдает вероятность отнесения того или иного объекта к одному из двух рассматриваемых классов. Вероятность – это число из диапазона $[0, 1]$, то есть непрерывная переменная, поэтому в названии и присутствует слово «регрессия». В то же время, в зависимости от значения полученной вероятности, наблюдению назначается тот или иной класс – вот и классификация. Так что, как оказывается, несогласованности никакой и нет, хотя на первый взгляд терминология и правда сбивает с толку.

Второй естественный вопрос, наверное, такой: мы же в предыдущей лекции уже изучили наивный байесовский классификатор (кстати говоря, вероятностный), зачем нам еще один, в чем их принципиальное отличие? Оказывается, что отличие возникает уже в самом начале, в самом подходе к построению вероятностной модели.

Снова начнем с постановки задачи. Пусть X – это множество объектов, каждый из которых описывается p признаками – случайными величинами $X_1, X_2, ..., X_p$, и откликом Y, который принимает значения из множества $\{-1, 1\}$. Будем трактовать $X \times Y$ как вероятностное пространство с некоторым совместным распределением. Интересует нас, конечно, оценка вероятности $\mathsf{P}(Y = 1|X_1, X_2, ..., X_p)$, ну или оценка противоположной вероятности $\mathsf{P}(Y = -1|X_1, X_2, ..., X_p)$; понятно, что в случае двухклассовой классификации они связаны соотношением

$$\mathsf{P}(Y = 1|X_1, X_2, ..., X_p) + \mathsf{P}(Y = -1|X_1, X_2, ..., X_p) = 1.$$

При рассмотрении наивного байесовского классификатора мы, пользуясь формулой Байеса, переходили к рассмотрению несколько других вероятно-

стей:

$$\mathsf{P}(Y|X_1, X_2, ..., X_p) = \frac{\mathsf{P}(Y, X_1, X_2, ..., X_p)}{\mathsf{P}(X_1, X_2, ..., X_p)} = \frac{\mathsf{P}(X_1, X_2, ..., X_p|Y)\mathsf{P}(Y)}{\mathsf{P}(X_1, X_2, ..., X_p)}.$$

При классификации нового наблюдения мы, перебирая все возможные значения y отклика Y (говоря по-взрослому, производя **оценку апостериорного максимума MAP (Maximum a posteriori estimation)**), искали максимум выражения

$$F(y) = \mathsf{P}(X_1, X_2, ..., X_p|Y = y)\mathsf{P}(Y = y),$$

оценивая при этом вероятности $\mathsf{P}(Y)$ и $\mathsf{P}(X_1, X_2, ..., X_p|Y)$ на основе тренировочных данных и предположении о наивности. Иными словами, смотря на числители дробей парой строчек выше, можно сделать вывод, что мы моделировали не что иное, как совместное распределение $\mathsf{P}(Y, X_1, X_2, ..., X_p)$, на основе которого и проводили классификацию.

Определение 1.1.1 *Алгоритмы, моделирующие совместное распределение* $\mathsf{P}(Y, X_1, X_2, ..., X_p)$, *часто называют* ***генеративными***.

В противовес генеративным алгоритмам, часто рассматривают так называемые дискриминативные алгоритмы.

Определение 1.1.2 *Алгоритмы, непосредственно моделирующие* $\mathsf{P}(Y|X_1, X_2, ..., X_p)$, *называются* ***дискриминативными***.

Алгоритм логистической регрессии, о котором мы будем говорить далее, является дискриминативным. В его основе заложено следующее предположение о (параметрическом) виде распределения условных вероятностей:

$$\mathsf{P}(Y = 1|X_1, X_2, ..., X_p) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + ... \theta_p X_p)}},$$

где параметры алгоритма – коэффициенты $\theta_0, \theta_1, ..., \theta_p$ – оцениваются, исходя из тренировочного набора данных.

Отличие генеративных от дискриминативных моделей можно неформально пояснить на следующем примере. Предположим, что наша цель – научиться отличать собак от кошек. Дискриминативная модель пытается в p-мерном пространстве признаков $\mathbb{R}^p$ построить некоторую границу, разделяющую (или почти разделяющую) тренировочные данные разных классов: прямую, плоскость или многообразие более хитрой формы. Далее, для классификации нового наблюдения, оказывается достаточным просто определить: с какой стороны от построенной границы находится новое наблюдение.

Генеративные же модели, как было сказано ранее, моделируют распределения для каждого класса в отдельности. Например, в примере с кошками и собаками, сначала, на основе тренировочных данных, строится модель «как выглядит кошка», а потом модель «как выглядит собака». Далее, для классификации нового наблюдения, достаточно просто сравнить, чья модель ему подходит больше: кошки или собаки. Вот и все отличие.

Что же, разобравшись с отличием в типах алгоритмов, перейдем к описанию алгоритма логистической регрессии, но для начала поймем, из каких соображений логично использовать соотношение для условной вероятности, введенное выше:

$$\mathsf{P}(Y = 1|X_1, X_2, ..., X_p) = \frac{1}{1 + e^{-(\theta_0+\theta_1 X_1+...\theta_p X_p)}}.$$

1.2 Построение модели логистической регрессии

Итак, согласно нашей постановке задачи, каждый объект, обладающий набором предикторов $X_1, X_2, ..., X_p$, должен быть отнесен к одному из двух классов: к классу «+1» или, условно, к положительному классу или к классу «−1» – условно, отрицательному классу. Причем, как мы уже отмечали ранее, понятно, что вероятности отнесения объекта к положительному классу

$$\mathsf{P}_+ = \mathsf{P}(Y = 1|X_1, X_2, ..., X_p)$$

и к отрицательному классу

$$\mathsf{P}_- = \mathsf{P}(Y = -1|X_1, X_2, ..., X_p)$$

связаны условием нормировки, то есть соотношением

$$\mathsf{P}_+ + \mathsf{P}_- = 1.$$

В итоге, если мы научимся оценивать вероятность P_+ попадания интересующего нас объекта в положительный класс, то вероятность попадания противоположный класс будет оцениваться выражением $\mathsf{P}_- = 1 - \mathsf{P}_+$. Но как эти оценки получить?

Замечание 1.2.1 *Отметим отдельно, что значения* P_+ *и, соответсвенно,* P_-*, являются функциями от* $X_1, X_2, ..., X_p$*. Мы опускаем аргументы функций лишь для краткости изложения.*

Давайте теперь перейдем от вероятностей, которые измеряются в диапазоне $[0, 1]$, к так называемым шансам. Рассмотрим величину $\mathsf{P}_+ \in [0, 1]$ – вероятность отнесения объекта с предикторами $X_1, X_2, ..., X_p$ к положительному классу. Тогда P_- – вероятность противоположного события, то есть вероятность отнесения того же самого объекта к отрицательному классу.

Определение 1.2.1 *Шансом отнесения объекта с предикторами* $X_1, X_2, ..., X_p$ *к положительному классу называется величина*

$$\text{odds}_+ = \text{odds}_+(X_1, X_2, ..., X_p) = \begin{cases} \frac{\mathsf{P}_+}{\mathsf{P}_-}, & \mathsf{P}_- \neq 0 \\ +\infty, & \mathsf{P}_- = 0 \end{cases}.$$

Например, если вероятность P_+ отнесения объекта к положительному классу равна 0.8, то шанс составит $4:1$, так как

$$\text{odds}_+ = \frac{0.8}{1-0.8} = 4,$$

что вполне себе жизненно и интуитивно понятно, не так ли? Теперь давайте ответим на следующий вопрос: а чем шанс лучше, чем вероятность? Дело тут вот в чем: в отличие от вероятности, шанс принимает уже любые неотрицательные значения, то есть

$$\text{odds}_+ \in [0, +\infty].$$

Логарифмируя шанс, получим величину, которая теперь уже может принимать любые значения из (расширенного) множества вещественных чисел

$$\ln(\text{odds}_+) = \ln\left(\frac{\mathsf{P}_+}{1-\mathsf{P}_+}\right) \in [-\infty, +\infty].$$

Итак, благодаря нашим достаточно нехитрым выкладкам, мы получили непрерывную переменную, которая зависит от $X_1, X_2, ..., X_p$ и может принимать любые значения из диапазона $[-\infty, +\infty]$. По сути дела, наша задача плавно перетекла в задачу регрессии, которую мы умеем решать. Правда, тут кроется некоторая проблема: мы не знаем значения P_+, поэтому и не знаем $\ln(\text{odds}_+)$. С другой стороны, если бы знали, то нечего было бы и оценивать. Игнорируя на данный момент эту проблему, придем к уравнению регрессии вида

$$\ln\left(\frac{\mathsf{P}_+}{1-\mathsf{P}_+}\right) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_p X_p.$$

Естественно, на данный момент мы не знаем значений $\theta_0, \theta_1, ..., \theta_p$.

Для удобства дальнейших выкладок, обозначим правую часть выражения за Ψ, то есть

$$\Psi = \Psi(X_1, X_2, ..., X_p) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_p X_p$$

Из соотношения

$$\ln\left(\frac{\mathsf{P}_+}{1-\mathsf{P}_+}\right) = \Psi$$

выразим искомую вероятность P_+. Беря экспоненты от обеих частей (или потенцируя написанное равенство), придем к выражению

$$e^{\Psi} = \frac{\mathsf{P}_+}{1 - \mathsf{P}_+},$$

которое, в свою очередь, может быть переписано, как

$$(1 - \mathsf{P}_+) \cdot e^{\Psi} = \mathsf{P}_+,$$

откуда

$$\mathsf{P}_+ = \frac{e^{\Psi}}{1 + e^{\Psi}}.$$

Преобразовав последнюю дробь, придем к выражению

$$\mathsf{P}_+ = \frac{1}{e^{-\Psi} \cdot (1 + e^{\Psi})} = \frac{1}{1 + e^{-\Psi}} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + ... + \theta_p X_p)}}.$$

Обратите внимание, это ровно-таки заявленное во введении параметрическое семейство.

Замечание 1.2.2 *Еще раз отметим, что так как Ψ, будучи приравненной к $\ln(\text{odds}_+)$, принимает значения в диапазоне $[-\infty, +\infty]$, то $e^{-\Psi}$ принимает значения в диапазоне $[0, +\infty]$, а значит $\mathsf{P}_+ \in [0, 1]$.*

1.3 Логистическая функция и алгоритм предсказания

Прежде чем обсудить важный вопрос нахождения коэффициентов $\theta_0, \theta_1, ..., \theta_p$, чуть подробнее рассмотрим аналитическое выражение, полученное для вероятности P_+. Оказывается, оно тесно связано с так называемой логистической функцией или сигмоидой (logistic function or sigmoid function).

Определение 1.3.1 *Функция*

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

называется логистической функцией или сигмоидой.

Наверное, многие из вас видят явное сходство выражений для P_+ и логистической функции:

$$\mathsf{P}_+ = \frac{1}{1 + e^{-\Psi}} \longleftrightarrow \sigma(x) = \frac{1}{1 + e^{-x}}.$$

Что можно извлечь из этого сходства? Для того, чтобы это понять, давайте для начала сформулируем основные свойства сигмоиды.

Лемма 1.3.1 *Функция*

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

обладает следующими свойствами:

1. $\sigma(x)$ *– возрастающая функция;*
2. $\sigma(x)$ *– непрерывная на* $\mathbb{R}$ *функция;*
3. $$\lim_{x \to +\infty} \sigma(x) = 1, \quad \lim_{x \to -\infty} \sigma(x) = 0.$$
4. $\sigma(x)$ *ограничена двумя горизонтальными асимптотами* $y = 0$ *и* $y = 1$.

Доказательство. Несмотря на то, что все написанные пункты достаточно очевидны, мы все же дадим несколько пояснений. Начнем с первого пункта. Так как функция e^x возрастает, то e^{-x} – убывает, функция $1 + e^{-x}$ тоже убывает, функция $\frac{1}{x}$ – убывает, а следовательно интересующая нас композиция $\frac{1}{1+e^{-x}}$ – возрастает.

Непрерывность написанной функции следует из непрерывности композиции элементарных функций на своей области определения.

Уравнения асимптот получаются из заявленных в третьем пункте пределов

$$\lim_{x \to +\infty} \sigma(x) = 1, \quad \lim_{x \to -\infty} \sigma(x) = 0,$$

которые, при желании, можно проверить, хотя бы по определению. □

График логистической функции приведен на рисунке 1. Из только что сформулированных свойств следует, что $\sigma(x)$ является функцией распределения некоторой случайной величины ξ, а значит

$$\mathsf{P}_+ = \sigma(\Psi) = \mathsf{P}(\xi < \Psi).$$

Замечание 1.3.1 *Из написанных свойств легко вывести следующий интересный (и, кстати, идейный) момент: «пограничная вероятность» отнесения объекта к классу «+1» (или «−1»), то есть вероятность* $\mathsf{P}_+ = \mathsf{P}_- = 0.5$, *– это значение функции* $\sigma(x)$ *в точке* 0. *В наших обозначениях это условие равносильно тому, что*

$$0 = \Psi = \theta_0 + \theta_1 X_1 + ... + \theta_p X_p.$$

Множество точек, удовлетворяющих написанному равенству, – это гиперплоскость (точка на прямой $\mathbb{R}^1$, *прямая на плоскости* $\mathbb{R}^2$, *плоскость в*

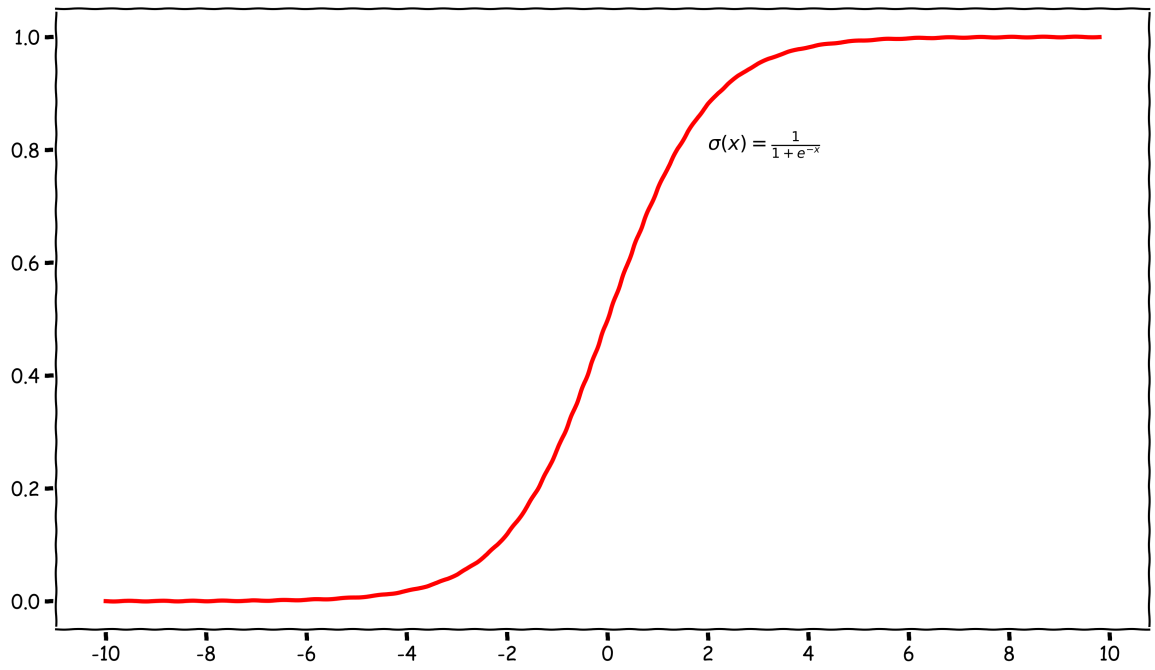

Рис. 1: Логистическая кривая (сигмоида).

пространстве $\mathbb{R}^3$ *и т.д.), лежащая в пространстве* $\mathbb{R}^p$ *и делящая его на две части. Эти части пространства имеет смысл трактовать следующим образом: в одной из частей лежат точки, для которых* $\mathsf{P}_+ > 0.5$*, то есть точки, которые скорее относятся к положительному, нежели к отрицательному классу, а в другой – точки, для которых* $\mathsf{P}_+ < 0.5$ *(или, что то же самое,* $\mathsf{P}_- > 0.5$*), то есть точки, которые скорее относятся к отрицательному, нежели положительному классу. Неуверенность, которую вы могли заметить в словах «скорее всего», и правда присутствует: мы вскоре проясним этот момент.*

В случае, когда граница двух классов является гиперплоскостью, классификатор называют линейным. Из всего сказанного можно сделать вывод, что алгоритм логистической регрессии, по своей сути, является линейным классификатором.

Сформулируем алгоритм предсказания класса нового объекта z с предикторами $(z_1, z_2, ..., z_p)$ в случае, когда коэффициенты $\theta_0, \theta_1, ..., \theta_p$ уже найдены.

1. Вычислить значение Ψ:

$$\Psi = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \ldots + \theta_p z_p.$$

2. Вычислить вероятность P_+:

$$\mathsf{P}_+ = \frac{1}{1 + e^{-\Psi}}.$$

3. Если $\mathsf{P}_+ \geq 0.5$, то объекту z назначить класс «+1», иначе – класс «−1».

На этом месте полезно остановиться и задаться вопросом: почему «пограничной» вероятностью является именно значение 0.5? На самом деле, выбор этой вероятности – вопрос очень неоднозначный. Например, при решении задачи классификации писем на две категории: «спам» и «не спам», вряд ли целесообразно отправлять письмо в категорию «спам», если классификатор уверен в этом лишь с вероятностью 0.51. В этом случае может быть правильнее использовать такое правило: если $\mathsf{P}_+ \geq 0.65$, то объекту z назначить класс «+1» (отправить в спам), иначе – класс «−1» (отправить в папку входящие). Или представьте, что мы исследуем самолет на надежность. Можно бы было его выпускать в рейс, если бы классификатор сказал, что самолет надежен с вероятностью 0.75? Видимо, в вопросах с самолетом «пограничная» вероятность должна быть серьезно больше, больше чем 0.9. В реальных задачах выбор этой вероятности часто отдается на откуп исследователю (или другим методам машинного обучения, как, например, k-fold cross-validation).

Замечание 1.3.2 *На самом деле, приведенный алгоритм логистической регрессии является алгоритмом, то есть отображением из пространства признаков в множество классов, лишь в случае, когда задано правило присвоения метки класса – когда выбрана вероятность, начиная с которой объект относят к классу «+1» (или, симметрично, к классу «−1»). Сама по себе логистическая регрессия (грубо говоря – функция P_+) выдает число в диапазоне* $[0, 1]$ *и часто называется базовым алгоритмом. Правило, согласно которому по выданной вероятности принимается решение об отнесении объекта к тому или иному классу, называется решающим правилом. Итого, можно сделать вывод, что алгоритм логистической регрессии – это композиция решающего правила и базового алгоритма логистической регрессии.*

Теперь применим алгоритм логистической регрессии к конкретным данным. В качестве последних будут выступать данные статистики футбольного матча; всего имеется три предиктора: количество ударов в створ ворот (X_1), процент владения мячом (X_2) и количество ударов в сторону ворот (X_3) в течение матча; отклик Y принимает всего два значения: значение 1 соответствует победе команды в матче (или отнесению ее к классу «+1»), а значение 0 соответствует проигрышу или ничьей (отнесение к классу «−1»). На основе тренировочных данных получены следующие значения параметров модели:

$$\theta_0 \approx -0.046, \quad \theta_1 \approx 0.541, \quad \theta_2 \approx -0.014, \quad \theta_3 \approx -0.132.$$

Классифицируем новый объект z:

$$z = (1, 40, 3)$$

– команду, которая в течение матча 1 раз ударила в створ ворот, владела мячом 40 процентов игрового времени и нанесла 3 удара в сторону ворот. Согласно описанному алгоритму, вероятность победы в матче оказывается равной

$$\mathsf{P}_+ = \frac{1}{1 + e^{-(\theta_0 + \theta_1 \cdot 1 + \theta_2 \cdot 40 + \theta_3 \cdot 3)}} \approx 0.38,$$

значит, команда скорее проиграет, нежели выиграет.

Итак, в случае, когда параметры модели найдены, предсказание осуществляется очень просто. Но как же найти эти параметры? Тут нам приходит на помощь метод максимального правдоподобия.

1.4 Метод максимального правдоподобия (ММП)

Метод максимального правдоподобия – один из мощнейших статистических методов, позволяющий получать по выборке оценки параметров семейств вероятностных распределений. Многие из вас, скорее всего, знакомы с этим методом. Однако в силу того, что он лежит в основе построения алгоритма логистической регрессии, мы считаем нужным все-таки подробно о нем рассказать (хотя и лишь в необходимой для нас строгости и общности – не в самой общей).

1.4.1 Наводящие соображения

Предположим, что проводится серия из n независимых одинаковых испытаний, вероятность успеха в каждом из которых равна $p \in (0, 1)$. Примером может служить стрельба по мишени или серия послематчевых пенальти. Напомним, что в озвученных предположениях вероятность получить ровно $k \in \{0, 1, .., n\}$ успехов в n испытаниях, то есть вероятность события $B(n, k)$, вычисляется по формуле Бернулли

$$\mathsf{P}(B(n, k)) = C_n^k p^k (1 - p)^{n-k},$$

где C_n^k – число сочетаний из n элементов по k элементов, равное

$$C_n^k = \frac{n!}{k!(n - k)!}$$

Пусть, к примеру, проводится следующий эксперимент: серия из пяти ударов по воротам с вероятностью успеха p (то есть с вероятностью забить мяч), равной 0.7, при каждом ударе. Эксперимент проводился дважды (независимо!), причем известно, что в первом эксперименте произошло два успеха (забили два гола), а во втором – четыре (забили 4 гола). Итого, перед нами следующая выборка: $X_1 = 2$, $X_2 = 4$. Какова же вероятность получить

такую выборку? В силу независимости, эта вероятность равна произведению вероятностей соответствующих событий, каждая из которых вычисляется по формуле Бернулли:

$$\mathsf{P}\left(X_1 = 2, X_2 = 4\right) = \mathsf{P}(X_1 = 2) \cdot \mathsf{P}(X_2 = 4) =$$

$$= \mathsf{P}(B(5,2)) \cdot \mathsf{P}(B(5,4)) = C_5^2 \cdot 0.7^2 \cdot (1-0.7)^3 \cdot C_5^4 \cdot 0.7^4 \cdot (1-0.7)^1.$$

Проведя вычисления, получаем, что вероятность рассматриваемого события невелика и примерно равна

$$\mathsf{P}\left(X_1 = 2, X_2 = 4\right) \approx 0.048.$$

Очевидно, что все вычисления мы смогли провести лишь потому, что изначально знали вероятность забития гола при каждом ударе по мячу. В реальности, конечно, все не так: обычно мы наблюдаем лишь какие-то проявления нашей случайной величины – выборку из нее, а также, если нам везет, знаем то параметрическое семейство вероятностных распределений, которому она подчиняется. В то же время сами параметры этих распределений мы не знаем! Поэтому ничего не остается, кроме как оценивать эти параметры по выборке. Как? Конечно, пытаясь максимизировать вероятность тех значений, которые мы наблюдаем.

Итак, продолжим рассмотрение нашего примера. Логично считать, что случайная величина – количество забитых голов в серии из 5 ударов, имеет так называемое биномиальное распределение с параметрами $n = 5$ (количество испытаний) и неизвестным параметром p (вероятность успеха в каждом испытании). В этом случае вероятность события $B(5,k)$ при $k \in \{0,1,2,...,5\}$, то есть события, что забито ровно k голов в серии из пяти ударов по воротам, может быть вычислена по формуле Бернулли

$$\mathsf{P}(B(5,k)) = C_5^k p^k (1-p)^{5-k}.$$

Значение p – вероятность забития гола при ударе по воротам, нам неизвестна, однако мы наблюдаем следующую выборку в результате двух независимых экспериментов: $X_1 = 2$, $X_2 = 4$. Вероятность получить эту выборку, в силу независимости экспериментов, может быть вычислена следующим образом:

$$f\left(X_1 = 2, X_2 = 4, p\right) = \mathsf{P}(X_1 = 2, X_2 = 4) = \mathsf{P}(X_1 = 2) \cdot \mathsf{P}(X_2 = 4) =$$

$$= C_5^2 \cdot p^2 \cdot (1-p)^3 \cdot C_5^4 \cdot p^4 \cdot (1-p)^1.$$

Перед нами – функция от p, а нам хочется найти такое значение p, при котором эта функция на отрезке $[0,1]$ примет свое наибольшее значение (именно при таком значении p наша выборка будет наиболее вероятной). Найденное

значение p и имеет смысл трактовать, как оценку истинного, нам неизвестного значения.

Легко видеть, что концы отрезка – не те значения, которые нас интересуют, так как в точках 0 и 1 рассматриваемая функция равна нулю. Поэтому впредь будем считать, что $p \in (0, 1)$. Итак, перед нами классическая задача математического анализа – задача нахождения точки, в которой заданная функция принимает наибольшее значение на заданном множестве. Для решения этой задачи можно воспользоваться следующим достаточно вольным алгоритмом.

1. Найти производную первого порядка.

2. Найти точки из области определения функции, в которых производная либо обращается в ноль, либо не существует. Все эти точки называются точками, подозрительными на экстремум.

3. Для проверки того, что точка, подозрительная на экстремум, является точкой локального максимума, воспользоваться каким-нибудь достаточным условием. Например, точка, подозрительная на экстремум будет точкой локального максимума, если при переходе через нее производная меняет знак с плюса на минус.

4. Сравнить значения функции в найденных точках локального максимума со значениями на границах множества (если таковые есть), выбрать наибольшее, а затем определить точку, в которой это наибольшее значение достигается.

Функция $f(X_1 = 2, X_2 = 4, p)$, которую мы хотим максимизировать, представляет собой произведение, что усложняет поиск производной, так как приходится многократно применять правило дифференцирования произведения функций, что, в свою очередь, довольно объемно. Для упрощения вычислений ее можно прологарифмировать и максимизировать так называемую логарифмическую функцию правдоподобия:

$$L(X_1 = 2, X_2 = 4, p) = \ln f(X_1 = 2, X_2 = 4, p).$$

Так как логарифм – монотонная функция, то точки экстремума функции $f(X_1 = 2, X_2 = 4, p)$ перейдут в точки экстремума функции $L(X_1 = 2, X_2 = 4, p)$, и наоборот. В итоге, после логарифмирования, произведение распадется в сумму логарифмов, и мы придем к окончательному выражению вида

$$L(X_1 = 2, X_2 = 4, p) = \ln\left(C_5^2 \cdot p^2 \cdot (1-p)^3 \cdot C_5^4 \cdot p^4 \cdot (1-p)^1\right) =$$

$$= \ln C_5^2 + \ln C_5^4 + 6 \ln p + 4 \ln(1-p).$$

Замечание 1.4.1 *Обратите внимание, что так как $p \in (0,1)$, то все преобразования оказываются законными.*

Итого, мы пришли к задаче поиска значения параметра $p \in (0,1)$, максимизирующего логарифмическую функцию правдоподобия

$$L\left(X_1 = 2, X_2 = 4, p\right) = \ln(C_5^2 \cdot C_5^4) + 6 \ln p + 4 \ln(1-p).$$

Для нахождения точек, подозрительных на экстремум, вычислим производную, она равна

$$\left(L\left(X_1 = 2, X_2 = 4, p\right)\right)'_p = \frac{6}{p} - \frac{4}{1-p}.$$

Приравняв производную к нулю, получим уравнение

$$\frac{6}{p} - \frac{4}{1-p} = 0,$$

откуда $p = 0.6$. Убедимся, что найденная точка – точка максимума. Для это-

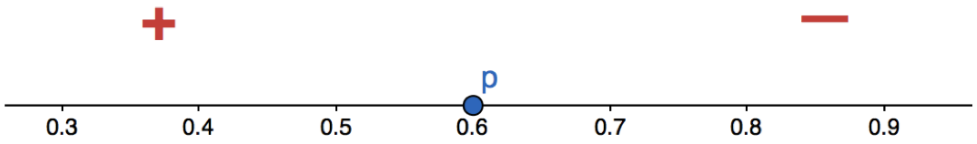

Рис. 2: Интервалы возрастания и убывания функции L.

го проверим знаки производной функции слева и справа от точки 0.6. Как видим, производная меняет свой знак с плюса на минус, а значит найденное значение $p = 0.6$ – точка максимума. Итак, вероятность события, что в первой серии из пяти ударов было забито два гола, а во второй – четыре, максимальна при $p = 0.6$.

Для того чтобы найти значение вероятности события $X_1 = 2, X_2 = 4$ при найденном $p = 0.6$, достаточно вычислить

$$f(X_1 = 2, X_2 = 4, 0.6) = C_5^2 \cdot 0.6^2 \cdot (1-0.6)^3 \cdot C_5^4 \cdot 0.6^4 \cdot (1-0.6)^1 \approx 0.06.$$

Итак, при найденном значении p вероятность события $X_1 = 2, X_2 = 4$ стала больше, чем при предыдущем, что и естественно, ведь мы нашли такое значение p, которое максимизирует наши наблюдения: два успеха в первом эксперименте и четыре во втором. Разобравшись в примере, перейдем к общему описанию метода максимального правдоподобия.

1.4.2 Сам метод максимального правдоподобия

Пусть имеется выборка X объема n, элементы которой $X_1, X_2, \ldots, X_n$ независимы, одинаково распределены и имеют некоторое распределение $\mathcal{P}_\theta$, известным образом зависящее от параметра θ. Этот параметр может принимать значения из какого-то множества Θ, то есть $\theta \in \Theta$. Например, в только что рассмотренном примере, семейство распределений – это семейство биномиальных распределений $\mathcal{P}_\theta = \mathsf{Bin}(5, p)$, зависящих от параметра $\theta = p$, причем множество значений Θ параметра θ – это отрезок $[0, 1]$.

Метод максимального правдоподобия – один из статистических методов, который состоит в построении оценки этого параметра. Грубо говоря, в качестве максимально правдоподобного значения θ берут такое, что при n испытаниях максимизируется вероятность получения выборки $x_1, x_2, \ldots, x_n$, полученной после эксперимента. Составим функцию

$$f(X, \theta) = \mathsf{P}_\theta \left(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\right).$$

Понятно, что функция $f(X, \theta)$ показывает вероятность события, что элементы выборки $X_1, X_2, ..., X_n$ равняются конкретным значениям $x_1, x_2, \ldots, x_n$, соответственно. Учитывая независимость элементов выборки $X_1, X_2, \ldots, X_n$, мы можем перейти к произведению вероятностей:

$$f(X, \theta) = \mathsf{P}_\theta \left(X_1 = x_1\right) \cdot \mathsf{P}_\theta \left(X_2 = x_2\right) \cdot \ldots \cdot \mathsf{P}_\theta \left(X_n = x_n\right).$$

Определение 1.4.1 *Функция*

$$f(X, \theta) = \mathsf{P}_\theta \left(X_1 = x_1\right) \cdot \mathsf{P}_\theta \left(X_2 = x_2\right) \cdot \ldots \cdot \mathsf{P}_\theta \left(X_n = x_n\right)$$

называется функцией правдоподобия.

Определение 1.4.2 *Оценкой максимального правдоподобия (Maximum likelihood estimate (MLE)) $\widehat{\theta}$ неизвестного параметра θ называется такое значение $\widehat{\theta} \in \Theta$, при котором функция правдоподобия $f(X, \theta)$ достигает максимума.*

Более коротко, интересующая нас задача переписывается следующим образом:

$$\widehat{\theta} = \operatorname{Arg}\max_{\theta} f(X, \theta),$$

где $f(X, \theta)$ – функция правдоподобия. Снова подчеркнем, что само название оператора говорит, что мы ищем аргумент (или аргументы, ведь их может быть несколько), максимизирующий функцию, а не значение максимума функции.

Замечание 1.4.2 *В принципе, аргументов θ, на которых достигается глобальный максимум функции $f(X,\theta)$, может быть несколько. В таком случае в качестве оценки $\widehat{\theta}$ выбирается любой элемент множества* $\underset{\theta}{\mathrm{Arg\,max}}\, f(X,\theta)$.

Как уже было отмечено в примере, рассмотренном ранее, для вычислительных удобств рассматривают не функцию правдоподобия $f(X,\theta)$, а ее логарифм – так называемую логарифмическую функцию правдоподобия.

Определение 1.4.3 *Пусть $f(X,\theta)$ – функция правдоподобия. Функция*

$$L(X,\theta) = \ln f(X,\theta) = \ln \mathsf{P}_\theta\left(X_1 = x_1\right) + \ldots + \ln \mathsf{P}_\theta\left(X_n = x_n\right).$$

называется логарифмической функцией правдоподобия.

Так как логарифм – монотонная функция, то точки экстремума функции $f(X,\theta)$ будут и точками экстремума функции $\ln f(X,\theta)$, и наоборот, а значит наша задача может быть переписана в виде

$$\widehat{\theta} = \underset{\theta}{\mathrm{Arg\,max}}\, L(X,\theta).$$

Теперь, изучив математическую суть вопроса, поговорим о применении рассмотренного аппарата к обучению алгоритма логистической регрессии.

1.5 Нахождение параметров модели

Итак, чтобы решить задачу нахождения неизвестных параметров модели, сначала вспомним наши исходные предположения. Мы предполагаем, что вероятность отнесения объекта с предикторами $X_1, X_2, ..., X_p$ к положительному классу «+1» задается выражением

$$\mathsf{P}_+ = \mathsf{P}_+(X_1, X_2, ..., X_p) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p)}}.$$

Обозначив

$$\Psi = \theta_0 + \theta_1 X_1 + ... + \theta_p X_p,$$

выражение для вероятности переписывается, как

$$\mathsf{P}_+ = \mathsf{P}_+(X_1, X_2, ..., X_p) = \frac{1}{1 + e^{-\Psi}} = \sigma[\Psi], \quad \sigma(x) = \frac{1}{1 + e^{-x}}.$$

Вероятность P_- отнесения объекта к отрицательному классу «-1» тогда равна

$$\mathsf{P}_- = \mathsf{P}_-(X_1, X_2, ..., X_p) = 1 - \mathsf{P}_+ = 1 - \sigma[\Psi].$$

Простейшие алгебраически преобразования приводят нас к тому, что

$$\mathsf{P}_- = 1 - \mathsf{P}_+ = 1 - \frac{1}{1+e^{-\Psi}} = \frac{e^{-\Psi}}{1+e^{-\Psi}} = \frac{1}{1+e^{\Psi}}.$$

Таким образом,

$$\mathsf{P}_- = \mathsf{P}_-(X_1, X_2, ..., X_p) = \frac{1}{1+e^{\Psi}} = \sigma[-\Psi].$$

Итого, у нас есть пара соотношений:

$$\mathsf{P}_+ = \mathsf{P}_+(X_1, X_2, ..., X_p) = \frac{1}{1+e^{-\Psi}} = \sigma[\Psi],$$

$$\mathsf{P}_- = \mathsf{P}_-(X_1, X_2, ..., X_p) = \frac{1}{1+e^{\Psi}} = \sigma[-\Psi].$$

Пусть теперь нам дан тренировочный набор данных $X = \{x_1, x_2, ..., x_n\}$ объема n,

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}), \quad i \in \{1, 2, ..., n\},$$

причем каждому объекту x_i соответствует отклик $y_i \in Y = \{-1, 1\}$. Для удобства введем следующее обозначение:

$$M(\theta, x_i) = y_i(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}).$$

Легко заметить, что если объект тренировочных данных относится к классу «+1», то, так как $y_i = 1$,

$$\sigma\left[M(\theta, x_i)\right] = \frac{1}{1+e^{-(\theta_0+\theta_1 x_{i1}+\theta_2 x_{i2}+...+\theta_p x_{ip})}} = \mathsf{P}_+,$$

а если к классу «−1», то, так как $y_i = -1$,

$$\sigma\left[M(\theta, x_i)\right] = \frac{1}{1+e^{(\theta_0+\theta_1 x_{i1}+\theta_2 x_{i2}+...+\theta_p x_{ip})}} = \mathsf{P}_-.$$

Естественно, имея тренировочный набор данных, в откликах которого мы уверены, все получившиеся вероятности должны быть настолько близкими к единице, насколько это возможно. «Настраивать» указанную близость мы можем, меняя значения $\theta_0, \theta_1, ..., \theta_p$. Ясно, что для этого логично использовать метод максимального правдоподобия, описанный ранее. Функция правдоподобия в нашем случае перепишется следующим образом:

$$f(X, \theta) = \sigma\left[M(\theta, x_1)\right] \cdot \sigma\left[M(\theta, x_2)\right] \cdot ... \cdot \sigma\left[M(\theta, x_n)\right] = \prod_{i=1}^{n} \sigma\left[M(\theta, x_i)\right].$$

Логарифмическая функция правдоподобия, которую, как и функцию правдоподобия, имеет смысл максимизировать, в этом случае примет следующий вид:

$$L(X, \theta) = \sum_{i=1}^{n} \ln \left(\sigma \left[M \left(\theta, x_i \right) \right] \right).$$

Нас же интересует такой набор значений θ, что

$$\theta = \operatorname{Arg}\max_{\theta} L(X, \theta).$$

С другой стороны, используя свойства логарифмов, задача может быть переписана и следующим образом:

$$\operatorname{Arg}\max_{\theta} L(X, \theta) = \operatorname{Arg}\max_{\theta} \sum_{i=1}^{n} \ln \left(\sigma \left[M(\theta, x_i \right] \right) =$$

$$= \operatorname{Arg}\max_{\theta} \sum_{i=1}^{n} \ln \left(1 + e^{-M(\theta, x_i)} \right)^{-1} = \operatorname{Arg}\min_{\theta} \sum_{i=1}^{n} \ln \left(1 + e^{-M(\theta, x_i)} \right),$$

так как максимизация функции

$$\sum_{i=1}^{n} \ln \left(1 + e^{-M(\theta, x_i)} \right)^{-1} = - \sum_{i=1}^{n} \ln \left(1 + e^{-M(\theta, x_i)} \right)$$

– то же самое, что минимизация функции

$$\text{logloss}(X, \theta) = \sum_{i=1}^{n} \ln \left(1 + e^{-M(\theta, x_i)} \right),$$

то есть той же самой функции, но без знака минус (формально – той же самой функции, но домноженной на -1).

Определение 1.5.1 *Введенная выше функция* logloss *называется логистической функцией ошибки (логистической функцией потерь).*

Замечание 1.5.1 *Полезно отметить, что логистическая функция потерь – частный случай эмпирического риска*

$$Q(a, L, x_1, x_2 ... x_n) = \frac{1}{n} \sum_{i=1}^{n} L(a, x_i),$$

где $L(a, x_i)$ – функция потерь, который обсуждался ранее. Чтобы в этом убедиться, достаточно в качестве функции потерь рассмотреть следующую функцию:

$$L(a, x) = n \ln \left(1 + e^{-M(\theta,x)}\right).$$

Итого, как и ранее, мы ищем такой алгоритм, то есть подбираем такие параметры $\theta_0, \theta_1, ..., \theta_p$, которые минимизируют эмпирический риск.

Написанная выше функция потерь вполне естественна, ведь чем хуже объект x согласуется с откликом, тем больше значение функции $1 + e^{-M(\theta,x)}$, и тем больший вклад логарифм дает в копилку эмпирического риска.

Минимизация полученной функции

$$\text{logloss}(X, \theta) = \sum_{i=1}^{n} \ln \left(1 + e^{-M(\theta,x_i)}\right),$$

конечно, вручную уже не проводится. Обычно это делается численно методами вроде градиентного спуска, но на этом подробно мы останавливаться не будем. Естественно, для решения задачи максимизации или минимизации, можно воспользоваться инструментами моделирования или математическими пакетами.

1.6 Пример составления оптимизационной задачи

Рассмотрим последовательность действий, приводящих к постановке задачи минимизации, описанной выше. Для начала рассмотрим урезанные данные по футбольной статистике, с которыми мы уже знакомы (с полной таблицей можно ознакомиться в дополнительных материалах к лекции).

Победа или проигрыш	Количество ударов в створ ворот	Процент владения мячом	Количество ударов в сторону ворот
1	7	40	13
0	0	60	6
0	3	43	8
1	4	57	14
...	...	...	...

Для наглядности будем считать, что наши исходные данные – это данные из таблицы, приведенной выше. Данные первого столбца отвечают откликам, а данные столбцов со второго по четвертый – предикторам. Каждая строка

отвечает своему тренировочному данному. Во введенных обозначениях, наши объекты таковы:

$$x_1 = (7, 40, 13), \quad y_1 = 1,$$
$$x_2 = (0, 60, 6), \quad y_2 = 0,$$
$$x_3 = (3, 43, 8), \quad y_3 = 0,$$
$$x_4 = (4, 57, 14), \quad y_4 = 1.$$

Поменяем значения класса 0 на значения -1, чтоы работать в уже введенных обозначениях. Тогда

$$x_1 = (7, 40, 13), \quad y_1 = 1,$$
$$x_2 = (0, 60, 6), \quad y_2 = -1,$$
$$x_3 = (3, 43, 8), \quad y_3 = -1,$$
$$x_4 = (4, 57, 14), \quad y_4 = 1.$$

Логистическая функция потерь переписывается в следующем виде:

$$\text{logloss}(X, \theta) = \sum_{i=1}^{4} \ln\left(1 + e^{-M(\theta, x_i)}\right) = \sum_{i=1}^{4} \ln\left(1 + e^{-y_i(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \theta_3 x_{i3})}\right)$$

Понятно, что при этом выражения на тренировочных данных для $M(\theta, x_i)$ таковы:

$$M(\theta, x_1) = +(\theta_0 + 7 \cdot \theta_1 + 40 \cdot \theta_2 + 13 \cdot \theta_3),$$
$$M(\theta, x_2) = -(\theta_0 + 0 \cdot \theta_1 + 60 \cdot \theta_2 + 6 \cdot \theta_3),$$
$$M(\theta, x_3) = -(\theta_0 + 3 \cdot \theta_1 + 43 \cdot \theta_2 + 8 \cdot \theta_3),$$
$$M(\theta, x_4) = +(\theta_0 + 4 \cdot \theta_1 + 57 \cdot \theta_2 + 14 \cdot \theta_3).$$

Полученная функция $\text{logloss}(X, \theta)$ является функцией четырех переменных, которую нужно минимизировать. Пример численной минимизации приведен в дополнительных материалах.

1.7 Отступ и «уверенность» классификации

Обратимся еще к одному примеру и обсудим некоторую геометрическую интерпретацию классификации при помощи логистической регрессии. Мы уже говорили, что в результате обучения модели мы получаем уравнение гиперплоскости

$$\theta_0 + \theta_1 X_1 + ... + \theta_p X_p = 0,$$

которая, в некотором смысле, отделяет представителей одного класса от представителей другого. В случае двух измерений гиперплоскость – это прямая на плоскости, в случае трех – плоскость в пространстве, и так далее. Рассмотрим простейший пример, в котором, что понятно из наглядных соображений,

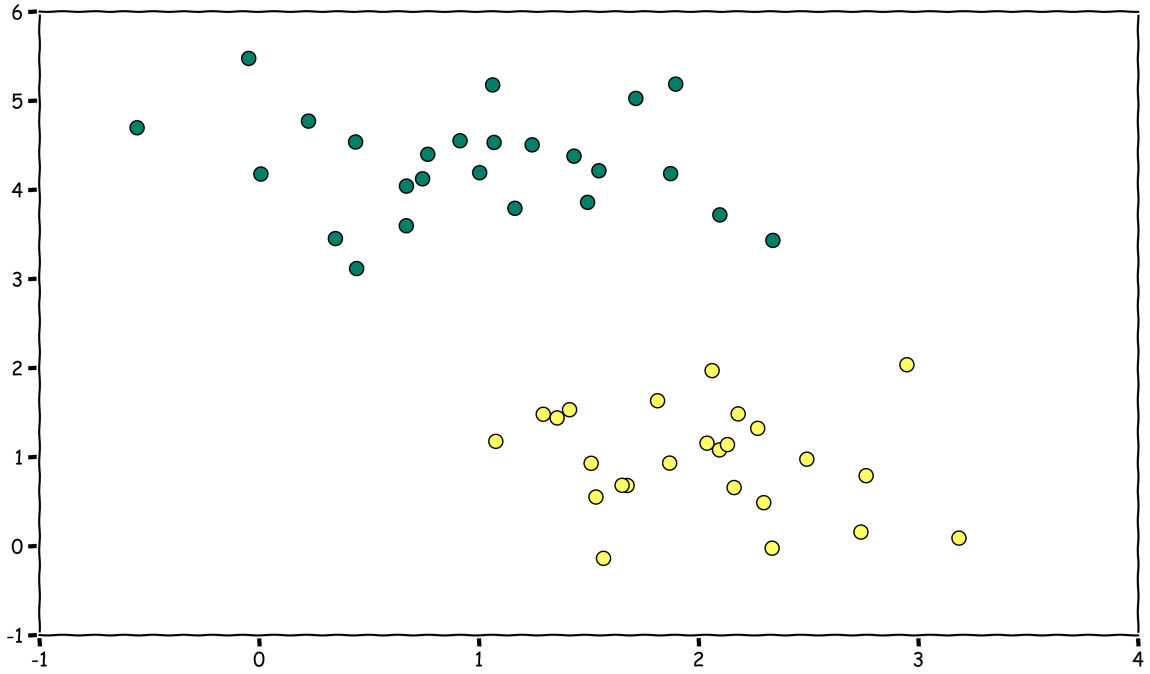

Рис. 3: Объекты интуитивно разделимы.

представители разных классов могут быть безошибочно разделены прямой на два класса, рисунок 3. Обучая модель логистической регрессии на представленных данных, приходим к следующему уравнению гиперплоскости:

$$1.07 + 1.65 \cdot X_1 - 1.55 \cdot X_2 = 0,$$

На рисунке 4 видно, что данные и правда безошибочно разделились построенной прямой. Важно отметить и следующее наблюдение: несмотря на то, что мы уверены в тренировочных данных (с вероятностью 1), некоторые данные находятся ближе к разделяющей прямой, а некоторые – дальше от нее, и ни для одного данного построенный классификатор не выдает вероятность, равную 1. Понятно, что те точки, которые находятся ближе к прямой, классифицируются менее «уверенно»: вероятности отнесения что к одному, что к другому классу близки к 0.5 (хотя одна из них и превалирует, раз точки не лежат на разделяющей прямой), а те, которые дальше от прямой – более «уверенно».

Определение 1.7.1 *Ранее введенная величина*

$$M(\theta, x_i) = y_i(\theta_0 + \theta_1 x_{i1} + ... + \theta_p x_{ip})$$

называется отступом (margin) объекта x_i.

В некотором смысле отступ показывает «степень погруженности» объекта в класс: чем отступ больше, тем дальше находится объект от разделяющей гиперплоскости и тем увереннее его классификация, и наоборот.

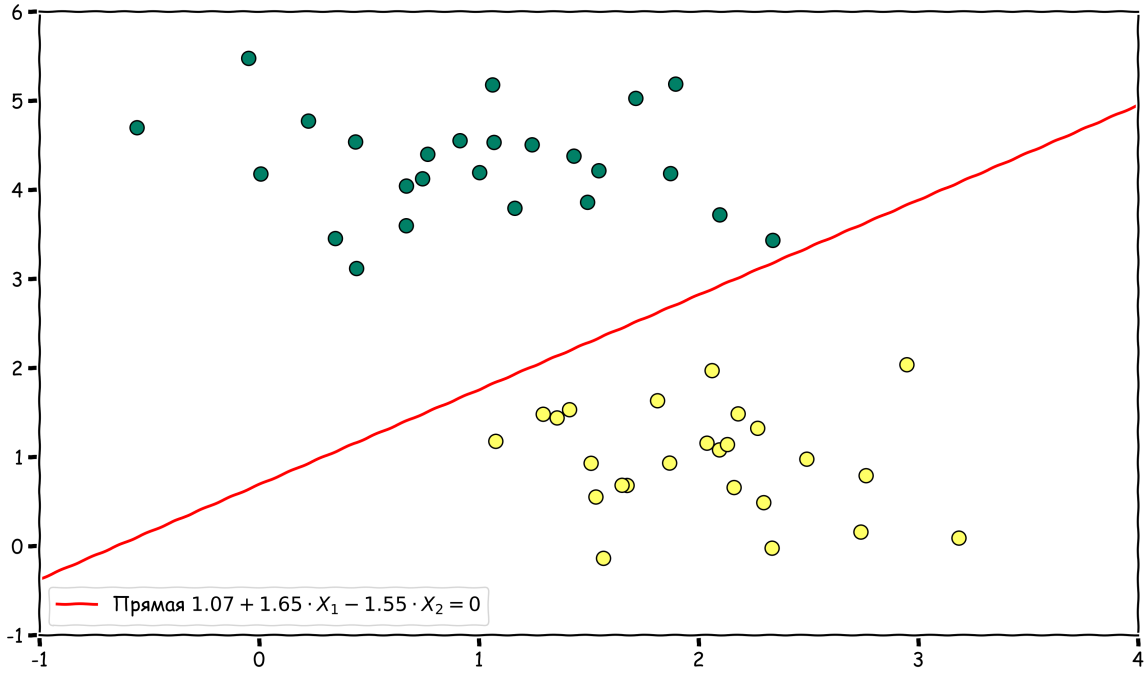

Рис. 4: Разделение объектов.

Замечание 1.7.1 *Отметим отдельно, что отступ $M(\theta, x_i)$ отрицателен тогда и только тогда, когда объект неправильно классифицирован нашим алгоритмом. Имея тренировочный набор данных $x_1, x_2, ..., x_n$, число ошибочно классифицированных объектов алгоритмом логистической регрессии можно записать через отступ следующим образом:*

$$\widetilde{Q}_{log}(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} \mathsf{I}(M(\theta, x_i) < 0).$$

Логично искать параметры алгоритма таким образом, чтобы минимизировать $\widetilde{Q}_{log}$, однако минимизировать написанное выражение неудобно. В то же время, так как

$$\mathsf{I}(M(\theta, x_i) < 0) \leq \log_2 \left(1 + e^{-M(\theta, x_i)}\right),$$

то мы приходим к тому, что

$$\widetilde{Q}_{log}(x_1, x_2, ..., x_n) \leq \sum_{i=1}^{n} \log_2 \left(1 + e^{-M(\theta, x_i)}\right) = \ln 2 \sum_{i=1}^{n} \ln \left(1 + e^{-M(\theta, x_i)}\right) =$$

$$= \ln 2 \cdot \text{logloss}(X, \theta).$$

Итого, минимизируя логарифмичекую функцию потерь, мы автоматически стараемся уменьшить и количество ошибок при классификации тренировочного набора данных. Все очень взаимосвязано!

Пусть теперь у нас есть набор тестовых объектов:

$$A = (0, 2), \quad y_A = 1,$$

$$B = (1, 2), \quad y_B = -1,$$

$$C = (3.5, 4), \quad y_C = 1,$$

$$D = (3, 0), \quad y_D = -1.$$

Обратимся к рисунку 5, на котором эти точки уже отмечены. Цвет точек соответствует исходному отклику (желтые точки относятся к классу «+1», а зеленые – к классу «−1»). Легко видеть, что объект A классифицирован неправильно: его цвет отличен от цвета представителей класса, располагающихся в принадлежащей ему части плоскости. Это можно понять и аналитически, используя отступ:

$$M(\theta, A) = 1 \cdot (1.07 + 1.65 \cdot 0 - 1.55 \cdot 2) = -2.03 < 0.$$

Итого, классификатор ошибается на объекте A и относит тестовое наблюдение к классу, отличному от истинного.

Объекты B и C находятся близко к разделяющей прямой, классификатор неуверен в своем ответе, однако ответ все равно оказывается верным. Это видно и аналитически:

$$M(\theta, B) = -1 \cdot (1.07 + 1.65 \cdot 1 - 1.55 \cdot 2) = 0.38 > 0,$$

$$M(\theta, C) = 1 \cdot (1.07 + 1.65 \cdot 3.5 - 1.55 \cdot 4) = 0.645 > 0$$

– значения хоть и маленькие, но положительные.

Объект D находится на значительном расстоянии от прямой и, похоже, является вбыросом. Это подтверждается и аналитически:

$$M(\theta, D) = -1 \cdot (1.07 + 1.65 \cdot 3 - 1.55 \cdot 0) = -6.02 < 0.$$

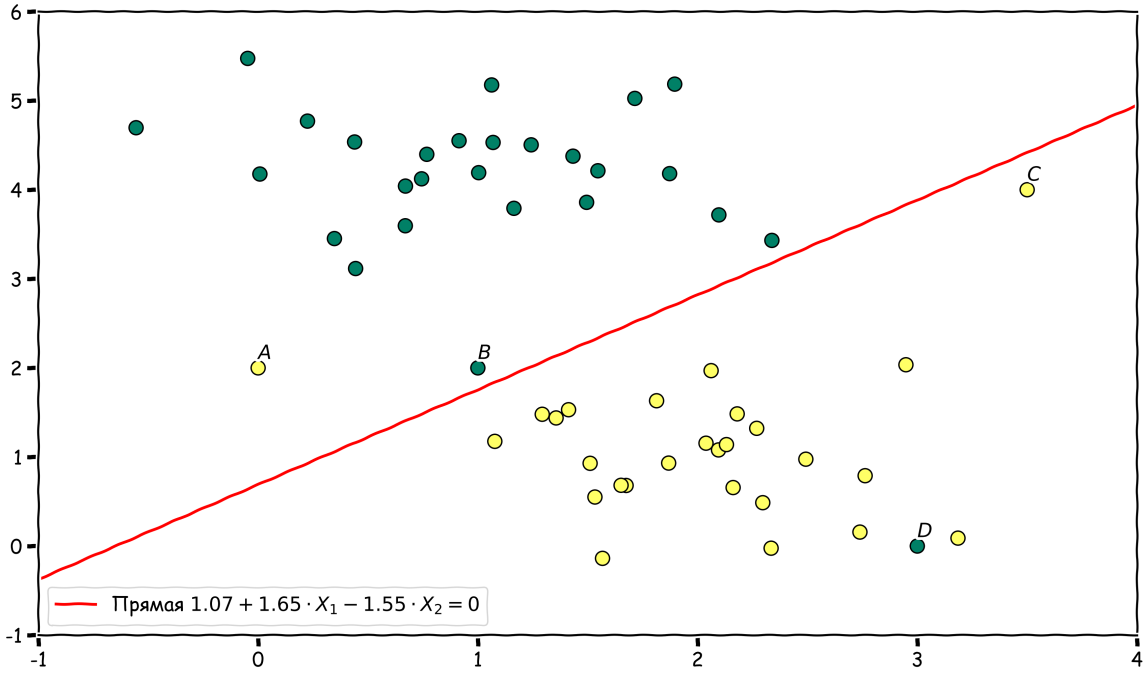

Рис. 5: Классификация новых объектов.

1.8 Сравнение линейной и логистической регрессий

Давайте посмотрим на существенную разницу в подходах линейной и логистической регрессии. Для этого возьмем в качестве предиктора, например, уровень дохода некоторого человека, а в качестве отклика факт одобрения банком кредита этому человеку. На рисунке 6 представлены наши обучающие данные. По горизонтальной оси отложен размер дохода (в тысячах рублей в месяц), а по вертикальной – факт получения кредита (единица означает, что кредит одобрен – желтые точки, ноль, что не одобрен – зеленые точки). Легко видеть, что при некотором «среднем» уровне дохода кредит иногда одобряется, а иногда – нет. Обучение модели логистической регрессии приводит к следующей сигмоиде, вы ее можете видеть на рисунке 7. Синим цветом на том же рисунке изображен график линии регрессии.

Возьмем набор новых клиентов с доходом 35, 40, 60, 70, 80 тысяч рублей и вычислим вероятности получения кредита. Они составят $0.11, 0.22, 0.89, 0.98, 0.99$ для логистической регрессии. Линейная регрессия дает вероятности $0.32, 0.39, 0.67, 0.82, 0.96$ (рисунки 7-8).

Какие выводы можно сделать? Да, в указанном примере классификация прошла одинаково с точки зрения конечного результата, то есть обе модели одинаково указали на клиентов, которым кредит, скорее всего, не дадут (предсказание меньше, чем 0.5), и которым кредит одобрят. Однако можно наблюдать следующее: при средних значениях предикторов (в примере – в

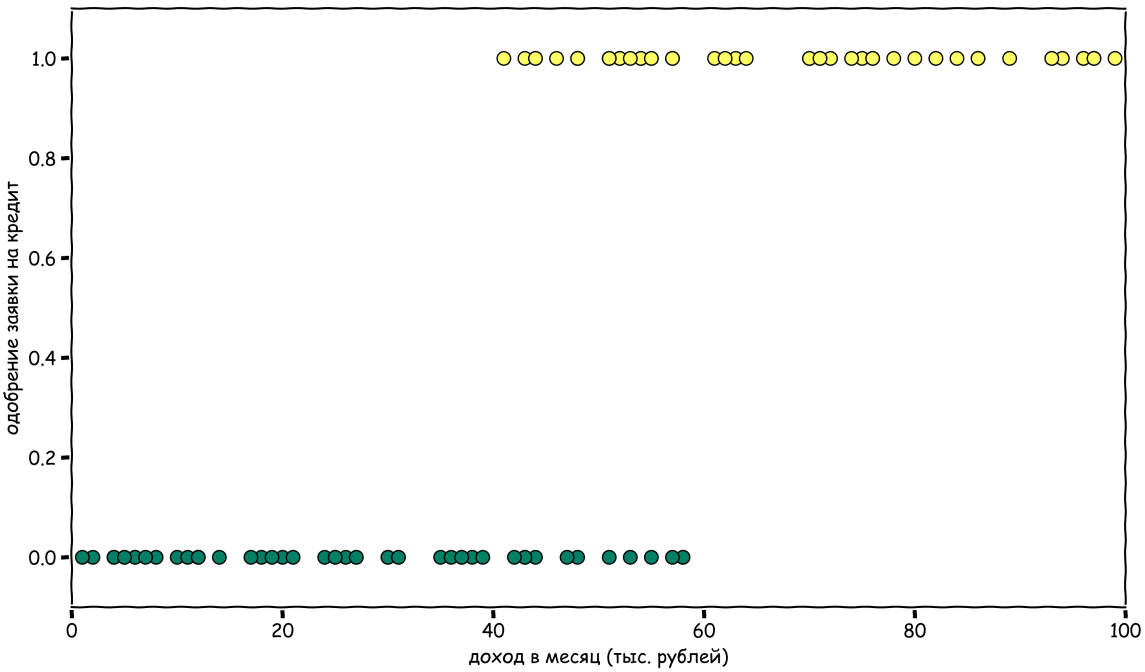

Рис. 6: Обучающие данные по выдаче кредитов.

районе 50 тыс.руб.) модели ведут себя по-разному. Так, в логистической регрессии видно сильное изменение значений вероятности за счет изменения выпуклости функции и ее стремительного роста при «средних» значениях дохода, тогда как в линейной модели значения вероятностей изменяются с одинаковой скоростью на всем диапазоне. С увеличением числа предикторов (например, с рассмотрением возраста, пола, наличия недвижимости), эти нюансы моделей будут давать значительный вклад при классификации.

Кроме того видно, что линейная регрессия даже на тренировочных данных дает странные результаты. Так, есть люди, для которых она выдает «результат», меньший нуля, а есть – для которых выдает «результат» больший, чем единица. И как это интерпретировать? Этот пример еще раз иллюстрирует одну из проблем использования линейной регрессии в задаче классификации: потеря нормировки.

2 Многоклассовая логистическая регрессия

2.1 Построение модели

Итак, мы разобрались с тем, как решать задачу двухклассовой классификации при помощи логистической регрессии. Но что, если классов больше, чем 2? Оказывается, что все написанное ранее можно обобщить. Пусть теперь каждый объект, имеющий набор предикторов $X_1, X_2, ..., X_p$, должен быть отнесен к одному из M классов: $Y = \{1, 2, ..., M\}$. Будем приближать так на-

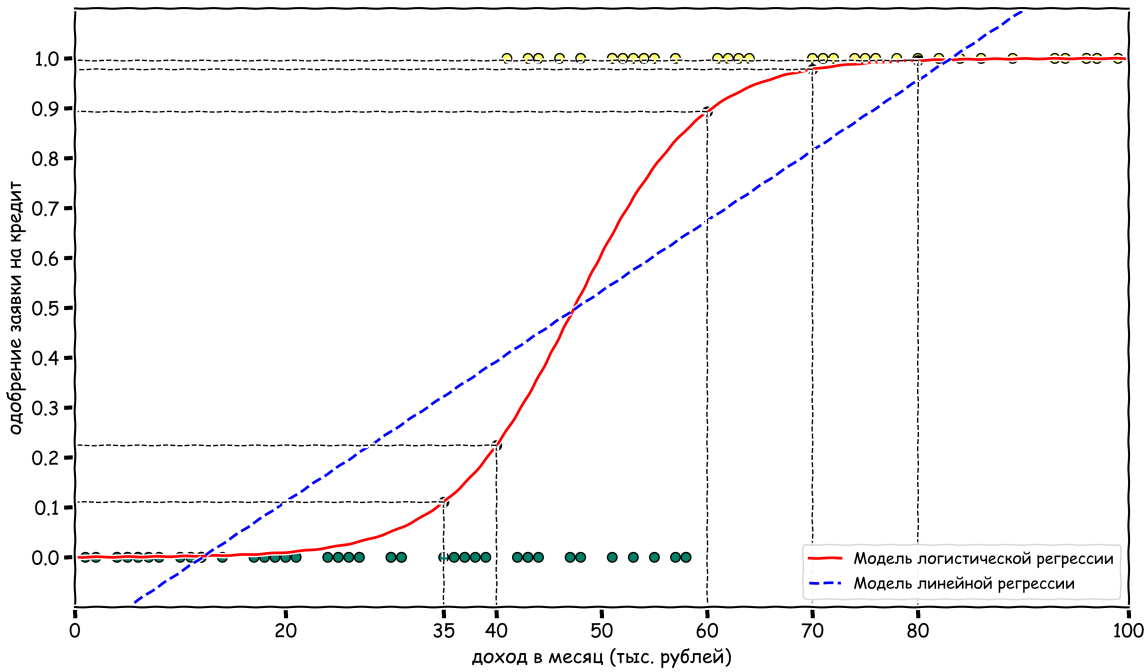

Рис. 7: Результаты логистической регрессии.

зываемые относительные шансы каждого класса своей линией регрессии. В итоге придем к следующим $(M-1)$ соотношениям:

$$\begin{aligned}
\ln \frac{\mathsf{P}(Y=1|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} &= \theta_0^1 + \theta_1^1 X_1 + ... + \theta_p^1 X_p, \\
\ln \frac{\mathsf{P}(Y=2|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} &= \theta_0^2 + \theta_1^2 X_1 + ... + \theta_p^2 X_p, \\
&\vdots \\
\ln \frac{\mathsf{P}(Y=M-1|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} &= \theta_0^{M-1} + \theta_1^{M-1} X_1 + ... + \theta_p^{M-1} X_p.
\end{aligned}$$

Мы видим, что в данной задаче нам нужно определить уже $(M-1)\cdot p$ неизвестных параметров. Прежде чем переходить к поиску этих параметров, сначала получим аналитические выражения для вероятности отнесения нового объекта к каждому из заявленных классов, диктуемые моделью. Обозначим

$$\Psi_i = \Psi_i(X_1, X_2, ..., X_p) = \theta_0^i + \theta_1^i X_1 + ... + \theta_p^i X_p, \quad i \in \{1, 2, ..., M-1\},$$

и добавим условие нормировки, чтобы получить вероятностное распределение. Тогда придем к следующей системе уравнений

$$\begin{cases}
\ln \frac{\mathsf{P}(Y=1|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} = \Psi_1, \\
\ln \frac{\mathsf{P}(Y=2|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} = \Psi_2, \\
\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
\ln \frac{\mathsf{P}(Y=M-1|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} = \Psi_{M-1}, \\
\sum\limits_{i=1}^{M} \mathsf{P}(Y=i|X_1,X_2,...,X_p) = 1.
\end{cases},$$

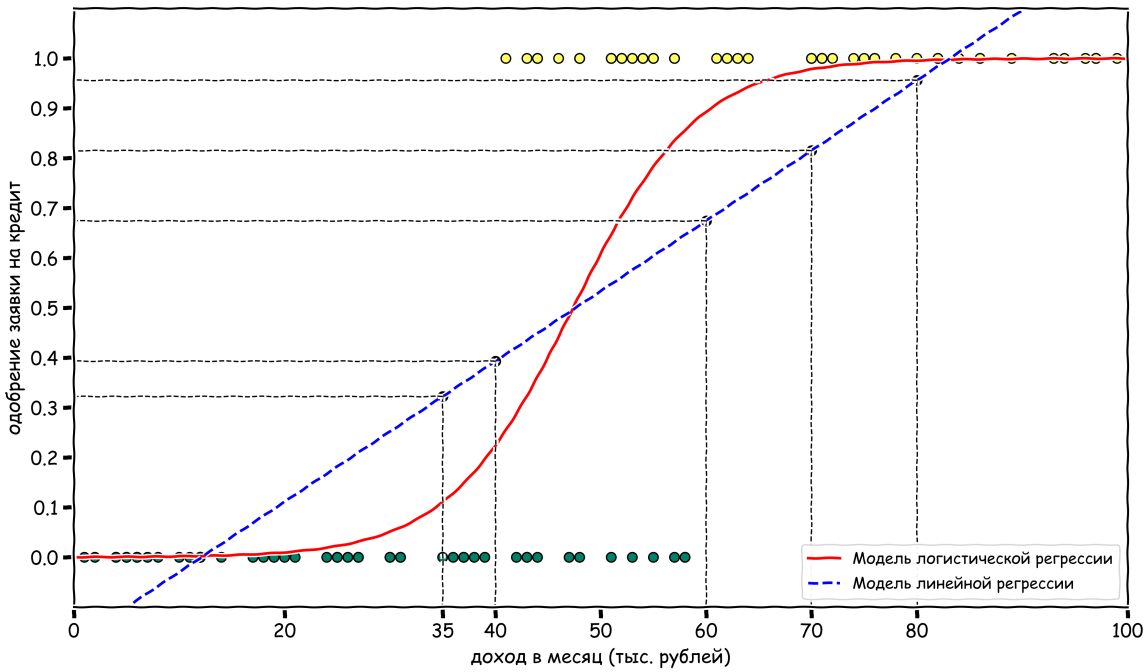

Рис. 8: Результаты линейной регрессии.

решив которую, получим следующие аналитические выражения для вероятностей:

$$\mathsf{P}_k = \mathsf{P}(Y = k|X_1, X_2, ..., X_p) = \frac{e^{\Psi_k}}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i}}, \quad k \in \{1, 2, ..., M-1\},$$

$$\mathsf{P}_M = \mathsf{P}(Y = M|X_1, X_2, ..., X_p) = \frac{1}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i}}.$$

Сформулируем алгоритм предсказания класса нового объекта z с предикторами $(z_1, z_2, ..., z_p)$ в случае, когда коэффициенты $\theta_0^i, \theta_1^i, ..., \theta_p^i$, $i \in \{1, 2, ..., M-1\}$, уже найдены.

1. Вычислить значения Ψ_k, $k \in \{1, 2, ..., M-1\}$:

$$\Psi_k = \theta_0^k + \theta_1^k z_1 + \theta_2^k z_2 + \ldots + \theta_p^k z_p.$$

2. Вычислить вероятности P_k, $k \in \{1, 2, ..., M-1\}$, исходя из соотношений:

$$\mathsf{P}_k = \frac{e^{\Psi_k}}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i}}, \quad k \in \{1, 2, ..., M-1\}$$

и вероятность P_m, исходя из соотношений

$$\mathsf{P}_M = \frac{1}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i}}.$$

3. Назначить тестовому объекту любой класс из множества

$$\underset{y\in\{1,2,...,M\}}{\operatorname{Arg\,max}}\ \mathsf{P}_y$$

Замечание 2.1.1 *Отметим отдельно, что последний пункт алгоритма может быть изменен исследователем в зависимости от задачи. Аргументация остается точно такой же, какой и была в случае двухклассовой классификации.*

2.2 Нахождение параметров модели

Перейдем к краткому описанию способа нахождения неизвестных параметров модели. Пусть нам дан тренировочный набор данных $X = \{x_1, x_2, ..., x_n\}$ объема n,

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}), \quad i \in \{1, 2, ..., n\},$$

причем каждому объекту x_i соответствует отклик $y_i \in Y = \{1, 2, ..., M\}$. Обозначим

$$\mathsf{P}_k(x_i) = \frac{e^{\Psi_k(x_{i1},x_{i2},...,x_{ip})}}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i(x_{i1},x_{i2},...,x_{ip})}}, \quad k \in \{1, 2, ..., M-1\}$$

и

$$\mathsf{P}_M(x_i) = \frac{1}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i(x_{i1},x_{i2},...,x_{ip})}}.$$

Используя метод максимального правдоподобия, функция правдоподобия примет вид:

$$f(X, \theta) = \mathsf{P}_{y_1}(x_1) \cdot \mathsf{P}_{y_2}(x_2) \cdot ... \cdot \mathsf{P}_{y_n}(x_n) = \prod_{i=1}^{n} \mathsf{P}_{y_i}(x_i),$$

а логарифмическая функция правдоподобия перепишется в виде

$$L(x, \theta) = \ln f(X, \theta) = \sum_{i=1}^{n} \ln \mathsf{P}_{y_i}(x_i).$$

Написанную функцию и нужно максимизировать, изменяя значения параметров $\theta_0^i, \theta_1^i, ..., \theta_p^i,\ i \in \{1, 2, ..., M-1\}$. Мы не будем останавливаться на пояснениях более детально, так как процесс максимизации сводится к применению тех или иных численных методов, и выходит за рамки нашего курса. Напоследок отметим, что аналогично случаю двух классов, максимизация логарифмической функции правдоподобия все так же ведет к минимизации количества ошибок рассматриваемого алгоритма на тренировочных данных.

2.3 Пример трехклассовой классификации

Применим описанный алгоритм к уже знакомым данным о сладости и хрусткости тех или иных классов продуктов. Каждый объект обладает двумя предикторами X_1 и X_2 и относится к одному из $M = 3$ классов: $Y = \{1, 2, 3\}$, где 1 отвечает фруктам, 2 – овощам, а 3 – протеинам. Как мы уже знаем, данные хорошо разделимы, так что модель на основе логистической регрессии должна показать себя хорошо.

Продукт	Сладость	Хруст	Класс
банан	10	1	фрукт
апельсин	7	4	фрукт
виноград	8	3	фрукт
креветка	2	2	протеин
бекон	1	5	протеин
орехи	3	3	протеин
сыр	2	1	протеин
рыба	3	2	протеин
огурец	2	8	овощ
яблоко	9	8	фрукт
морковь	4	10	овощ
сельдерей	2	9	овощ
салат айсберг	3	7	овощ
груша	8	7	фрукт

Обучив алгоритм многоклассовой логистической регрессии средствами моделирования на представленных данных, приходим к следующим выражениям для Ψ_1 и Ψ_2 (с округленными коэффициентами):

$$\Psi_1 = \Psi_1(X_1, X_2) = -5.561 + 10.786X_1 - 9.976X_2$$

$$\Psi_2 = \Psi_2(X_1, X_2) = -50.441 + 15.765X_1 - 3.961X_2.$$

Для нахождения вероятности отнесения тестового объекта к тому или иному классу воспользуемся выведенными ранее формулами:

$$\mathsf{P}_k\left(x_i\right) = \frac{e^{\Psi_k(x_{i1},x_{i2},\ldots,x_{ip})}}{1+\sum\limits_{i=1}^{2} e^{\Psi_i(x_{i1},x_{i2},\ldots,x_{ip})}}, \quad k \in \{1,2\},$$

$$\mathsf{P}_3\left(x_i\right) = \frac{1}{1+\sum\limits_{i=1}^{2} e^{\Psi_i(x_{i1},x_{i2},\ldots,x_{ip})}}.$$

Для начала вычислим значения Ψ_k, $k \in \{1,2\}$ для нового объекта «Перец» с предикторами $(6,9)$:

$$\Psi_1 = \Psi_1(6,9) = -5.561 + 10.786 \cdot 6 - 9.976 \cdot 9 \approx -29.012$$

$$\Psi_2 = \Psi_2(6,9) = -50.441 + 15.765 \cdot 6 - 3.961 \cdot 9 \approx 8.495.$$

Тогда:

$$\mathsf{P}_1 = \frac{e^{\Psi_1}}{1+e^{\Psi_1}+e^{\Psi_2}} \approx 0,$$

$$\mathsf{P}_2 = \frac{e^{\Psi_2}}{1+e^{\Psi_1}+e^{\Psi_2}} \approx 1,$$

$$\mathsf{P}_3 = \frac{1}{1+e^{\Psi_1}+e^{\Psi_2}} \approx 0.$$

Из проведенных вычислений становится совершенно понятно, что обученный алгоритм относит «Перец» к классу овощей чрезвычайно уверенно, с вероятностью очень близкой к 1. На рисунке можно увидеть три области, на которые построенный классификатор разделил все пространство признаков. Все точки, лежащие в верхней светло-зеленой части классифицируются как овощи, точки, находящиеся слева в желтой области – как протеины, а точки, находящиеся в темно-зеленой области справа – как фрукты. Понятно, что чем дальше объект расположен от образовавшихся границ, тем увереннее (то есть с большей вероятностью) он будет классифицирован к тому или иному классу.

3 F-мера и ROC-анализ

3.1 Матрица ошибок и F-мера

ROC-кривая или кривая ошибок – кривая, которая наиболее часто используется для оценки результатов бинарной классификации в машинном

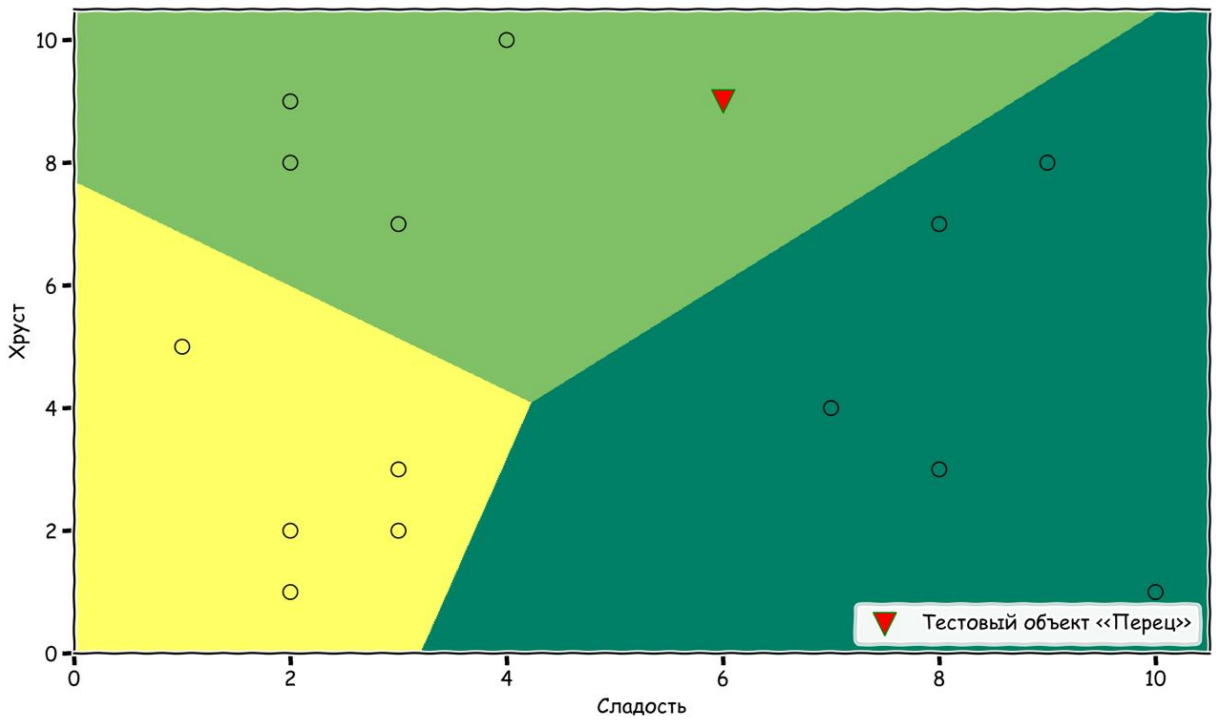

Рис. 9: Модель многомерное логистической регрессии.

обучении. Кривая ошибок показывает зависимость доли истинно положительных примеров от доли ложно положительных примеров. При этом предполагается, что у классификатора имеется некоторый параметр, изменение которого влияет на то или иное разбиение на эти два класса.

Для начала рассмотрим так называемую матрицу ошибок (confusion matrix) – способ разделить объекты на 4 группы в зависимости от комбинации истинного класса и ответа классификатора:

- TP (True Positives) – верно классифицированные объекты, исходно относящиеся к классу «+1»;
- TN (True Negatives) – верно классифицированные объекты, исходно относящиеся к классу «−1»;
- FN (False Negatives) – неверно классифицированные объекты, исходно относящиеся к классу «+1» (ошибка I рода);
- FP (False Positives) – неверно классифицированные объекты, исходно относящиеся к классу «−1» (ошибка II рода).

Обычно, конечно, оперируют не абсолютными показателями, а относительными – долями (rates), находящимися в диапазоне от 0 до 1:

Матрица ошибок		Исходный класс	
		+	–
Ответ классификатора	+	TP	FP
	–	FN	TN

- доля правильных ответов классификатора (иногда – точность):

$$\mathsf{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}.$$

Эта величина показывает отношение количества верно классифицированных объектов к общему количеству классифицируемых объектов и, грубо говоря, оценивает вероятность случайному объекту быть правильно классифицированным.

- доля истинно положительных примеров – True Positives Rate (TPR) или Sensitivity (чувствительность) или Recall:

$$\mathsf{TPR} = \frac{TP}{TP + FN}.$$

Эта величина показывает отношение количества верно классифицированных объектов, относящихся к классу «+1», к общему количеству объектов класса «+1». Иными словами – это оценка вероятности, что объект, относящийся к классу «+1» будет классифицирован корректно.

- доля ложно положительных примеров обозначается как – False Positives Rate (FPR):

$$\mathsf{FPR} = \frac{FP}{FP + TN}.$$

Величина показывает отношение количества неверно классифицированных объектов, относящихся к классу «−1», к общему количеству объектов класса «−1», или оценивает вероятность, что объект, относящийся к классу «−1», будет классифицирован неверно.

- Специфичность (Specificity) или True Negatives Rate (TNR):

$$\mathsf{TNR} = 1 - \mathsf{FPR} = \frac{TN}{TN + FP}.$$

Величина показывает отношение количества верно классифицированных объектов, относящихся к классу «−1», к общему количеству объектов класса «−1», или оценивает вероятность, что объект, относящийся к классу «−1», будет классифицирован верно.

- Precision (точность):

$$\mathsf{Precision} = \frac{TP}{TP + FP}.$$

Величина показывает, какая доля объектов, отнесенных классификатором классу «+1», действительно относится к этому классу.

Часть из введенных понятий очень хорошо иллюстрируется на таком примере из медицины. Пусть к положительному классу относятся пациенты, имеющие заболевание, а к отрицательному – не имеющие. Чувствительный диагностический тест (с высоким TPR) – это тот, который правильно идентифицирует пациентов с заболеванием. То есть если тест на 100% чувствителен ($\mathsf{TPR} = 1$), то он верно определит всех пациентов, у которых есть заболевание (то есть всем болеющим скажет, что они больны). В то же время, он может записать к заболевшим и тех, кто не болен. Высокочувствительный тест полезен для исключения заболевания.

Специфичный диагностический тест (с высоким TNR) диагностирует только доподлинно больных, то есть, если тест имеет 100% специфичность ($\mathsf{TNR} = 1$), он будет верно идентифицировать пациентов, которые не имеют заболевания, но может записать к здоровым и больных. Такой тест важен при лечении пациентов с определенным заболеванием.

Естественно возникает вопрос, нет ли какого-то обобщающего критерия, который может характеризовать качество построенной модели. Один из ни – так называемая F-мера (F_1-мера, F score, F_1 score) определяется следующим соотношением:

$$F = F_1 = 2 \cdot \frac{\mathsf{Precision} \cdot \mathsf{Recall}}{\mathsf{Precision} + \mathsf{Recall}}.$$

Замечание 3.1.1 *F-мера является средним гармоническим величин* $\mathsf{Precision}$ *и* Recall *и заключена в диапазоне* $[0, 1]$*. Среднее гармоническое обладает важным свойством: оно близко к нулю, если хотя бы один аргументов близок к нулю. Поэтому оно является куда более предпочтительным, чем, скажем, среднее арифметическое: если алгоритм относит все объекты к положительному классу, то* $\mathsf{Recall} = 1$*, а* $\mathsf{Precision}$*, скорее всего, будет небольшим. Но тогда среднее арифметическое будет больше, чем* 0.5*, что, конечно, никуда не годится.*

Вернемся к примеру по футбольной статистике, в котором мы уже обучили модель и нашли значения θ:

$$\theta_0 = -0.046, \theta_1 = 0.541, \theta_2 = -0.014, \theta_3 = -0.132.$$

Используем тестовый набор данных, представленный в таблице ниже, и составим матрицу ошибок.

Победа или проигрыш	Количество ударов в створ	Процент владения мячом	Удары по воротам
1	5	60	10
1	2	35	3
0	3	45	6
0	1	53	10
1	7	70	11
1	3	65	12
1	1	30	2
0	2	40	9
1	10	71	15
1	6	54	12
0	7	65	15
0	0	30	3

Для начала найдем вероятность победы для каждого набора данных. Так, для команды, попавшей в створ ворот в 5 из 10 случаев, и владевшей мячом 60 процентов игрового времени, вероятность победы составит:

$$\mathsf{P}_+ = \frac{1}{1 + e^{-(\theta_0 + \theta_1 \cdot 5 + \theta_2 \cdot 60 + \theta_3 \cdot 10)}} \approx 0.588.$$

Но к какому классу мы отнесем этот объект, глядя на результат? Если порог отсечения 0.5, то классу победивших, а если 0.6, то к классу проигравших. Найдем вероятности победы в матче для всех оставшихся тестовых данных и поместим округленные результаты в таблицу. На практике значения, конечно, лучше не округлять.

Вероятность победы
0.588
0.520
0.517
0.186
0.743
0.285
0.446
0.337
0.886
0.671
0.666
0.307

В качестве порога отсечения выберем значение 0.5 и назначим классы.

Вероятность победы	Победа или проигрыш (предсказание)
0.588	1
0.520	1
0.517	1
0.186	0
0.743	1
0.285	0
0.446	0
0.337	0
0.886	1
0.671	1
0.666	1
0.307	0

Легко сопоставить предсказанные классы с исходными. Как видим, они не всегда одинаковы, а это значит, что модель ошибается.

Победа или проигрыш	Вероятность победы	Победа или проигрыш (предсказание)
1	0.588	1
1	0.520	1
0	0.517	1
0	0.186	0
1	0.743	1
1	0.285	0
1	0.446	0
0	0.337	0
1	0.886	1
1	0.671	1
0	0.666	1
0	0.307	0

Теперь мы можем составить матрицу ошибок. Подсчитаем число команд, которые победили в матче как по исходным данным, так и по прогнозу модели, таких команд пять. Аналогично подсчитаем число команд, проигравших в матче как по исходным данным, так и по прогнозу, их три. Ошибок первого и второго рода по две. Тогда доля истинно положительных примеров составит

$$\mathsf{TPR} = \frac{TP}{TP + FN} = \frac{5}{5 + 2} \approx 0.71,$$

Матрица ошибок		Исходный класс	
		+	−
Прогноз	+	TP=5	FP=2
	−	FN=2	TN=3

а доля ложно положительных составит

$$\mathsf{FPR} = \frac{FP}{TN + FP} = \frac{2}{3+2} = 0.4.$$

Кроме того,

$$F_1 \approx 0.71,$$

что можно считать неплохим результатом.

3.2 ROC-кривая

Выше мы изучили точность, полноту и F-меру – оценки качества работы построенного алгоритма при фиксированном пороге отсечения. Однако часто выбор порога отсечения после построения модели – и есть еще одна немаловажная задача. Для ее решения часто бывает полезной такая интегральная метрика качества, как ROC-кривая. Она строится, как зависимость TPR от FPR. Для ее построения нужно вычислить соответствующие значения при разных порогах отсечения.

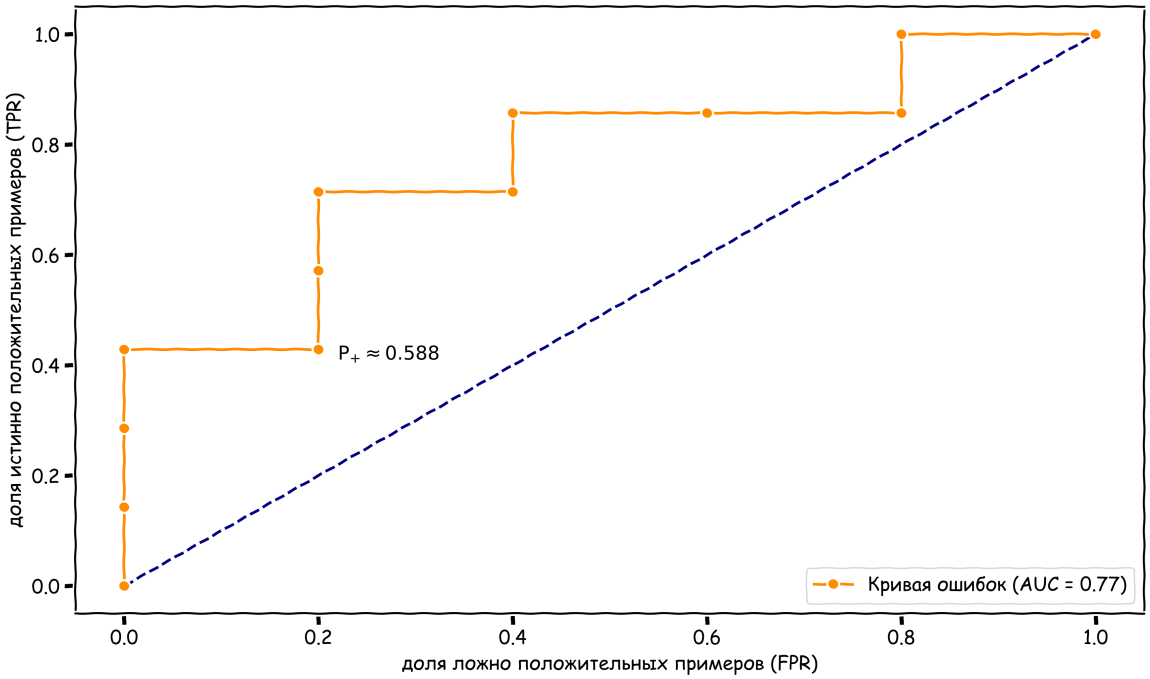

Рис. 10: Кривая ошибок.

Существуют различные подходы к изменению значений порога. Его можно менять от нуля до единицы с некоторым шагом, а можно в качестве его

значений брать полученные по модели вероятности, отсортированные по возрастанию.

В результате последнего, каждой паре значений $\mathsf{FPR}_i, \mathsf{TPR}_i$ соответствует точка на плоскости, соседние точки соединяются отрезками прямых. Так как при переходе через следующее значение вероятности меняется только один результат классификации, то скачок происходит либо вверх, либо вправо, а фигура получается ступенчатой (рисунок 10). Например, значению вероятности 0.588, соответствует доля истинно положительных примеров 0.2 и доля ложно положительных примеров 0.429. Точки с координатами $(0, 0)$ и $(1, 1)$ всегда отмечаются и являются началом и концом кривой.

В идеальной вселенной, кривая проходит через верхний левый угол, где процент истинно положительных случаев составляет 100%. Поэтому, чем больше выгнута ROC-кривая, тем более точным является прогнозирование результатов модели. Оценка модели может быть получена непосредственно вычислением площади под ROC-кривой. Показатель обозначается как AUC (Area Under Curve – площадь под кривой) или $AUC-ROC$. Так как получившаяся фигура является ступенчатой, ее площадь может быть легко вычислена как сумма площадей прямоугольников на основе значений $\mathsf{TPR}_i, \mathsf{FPR}_i$.

В зависимости от значений AUC, на практике часто оценивают качество модели следующим образом: отличное, если $AUC \in (0.9, 1]$; хорошее, если $AUC \in (0.7, 0.9]$; среднее, если $AUC \in (0.60.7]$; неудовлетворительное, если $AUC \in (0.5, 0.6]$.

Как же найти оптимальное значение порога отсечения? Оптимальным значением порога, будет точка пересечения графика чувствительности и специфичности. График строится аналогично случаю ROC-кривой, только теперь на одной плоскости строится зависимость чувствительности от порога, и специфичности от порога (рисунок 11). Для рассматриваемого примера это значение составит около 0.52. Еще раз отметим, что на практике значения вероятностей и прочих величин не стоит округлять.

Порог мало отличается от рассмотренного ранее, да и объем данных невелик, однако матрица ошибок уже будет другой:

Матрица ошибок		Исходный класс	
		+	–
Прогноз	+	TP=4	FP=1
	–	FN=3	TN=4

При таком пороге отсечения доля истинно положительных примеров составит 0.571 и доля ложно положительных примеров 0.2. То есть модель стала лучше отсекать отрицательные данные за счет увеличения ее специфичности.

Таким образом, цель ROC-анализа заключается в том, чтобы подобрать такое значение точки отсечения, которое позволит модели с наибольшей точ-

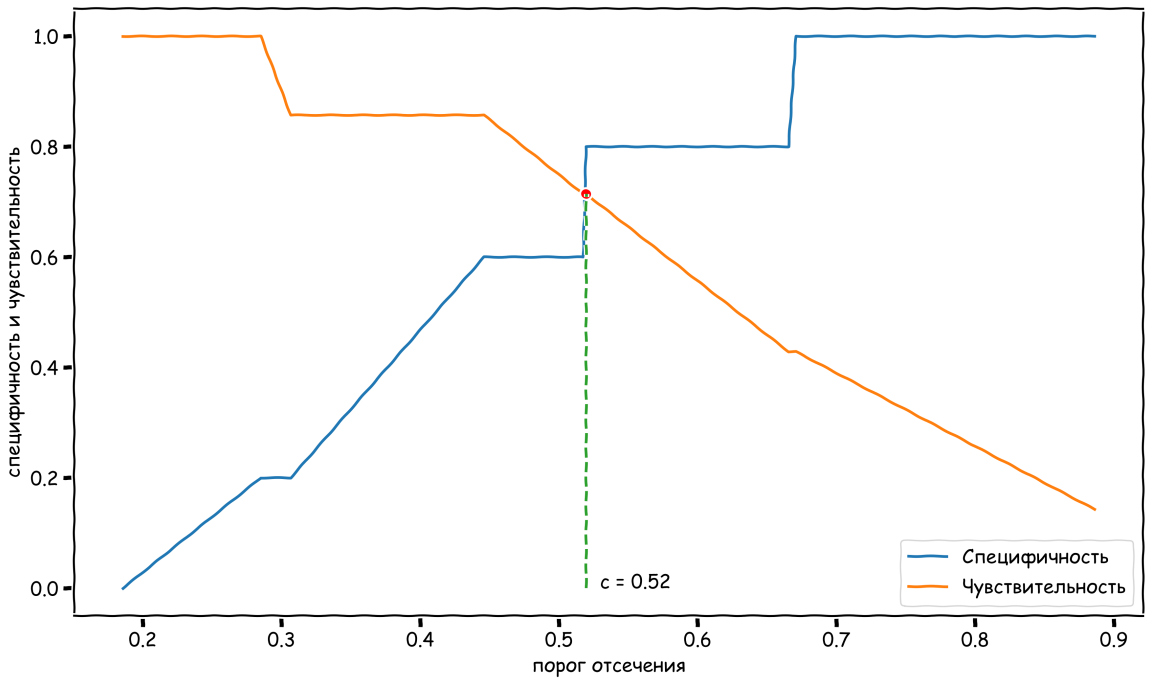

Рис. 11: Определение порога отсечения.

ностью распознавать положительные или отрицательные исходы и выдавать наименьшее количество ложноположительных или ложноотрицательных ошибок.

4 Заключение

Итак, в этой лекции мы познакомились с еще одним вероятностным алгоритмом многоклассовой классификации – с логистической регрессией. Кроме того, мы изучили различные методы оценки качества построенного классификатора. На сегодня все, до новых встреч!