Математика для Machine Learning

Линдеманн Никита

22 октября 2019 г.

Содержание

1 Элементы математического анализа

1.1 Производная функции одной переменной

Определение 1.1.1. Производная функции одной переменной $y = f(x)$ в точке x_0 – это предел:

$$\frac{d}{dx}f(x_0) = \frac{d}{dx}y(x_0) = f'(x_0) = y'(x_0) = \lim_{x\to x_0}\frac{f(x)-f(x_0)}{x-x_0}.$$

Если обозначить $\Delta x = x - x_0$, то производную функции в точке можно записать в виде:

$$f'(x_0) = \lim_{x\to x_0}\frac{f(x)-f(x_0)}{x-x_0} = \lim_{\Delta x\to 0}\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}.$$

Пример 1.1.1. Найдем по определению производную функции $y = x^2$ в точке $x_0 = 3$:

$$y'(3) = \lim_{x\to 3}\frac{x^2-3^2}{x-3} = \lim_{x\to 3}\frac{(x-3)(x+3)}{x-3} = \lim_{x\to 3}(x+3) = 3+3 = 6.$$

Можно было действовать по-другому:

$$y'(x) = \lim_{\Delta x\to 0}\frac{(x+\Delta x)^2-x^2}{\Delta x} = \lim_{\Delta x\to 0}\frac{2x\Delta x+\Delta x^2}{\Delta x} = \lim_{\Delta x\to 0}(2x+\Delta x) = 2x.$$

Далее подставим значение $x_0 = 3$ в найденную функцию $y'(x) = 2x$:

$$y'(x_0) = y'(3) = 2\cdot 3 = 6.$$

Конечно, на практике никто не считает производные по определению, для этого есть готовая таблица производных элементарных функций (которую полезно помнить наизусть) и правила дифференцирования.

Правила дифференцирования:

1. Линейность производной: $(\alpha f(x) + \beta g(x))' = \alpha f'(x) + \beta g'(x)$.
2. Производная произведения: $(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$.
3. Производная частного:
$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x)-f(x)g'(x)}{g^2(x)}.$$
4. Производная сложной функции:
$$(f(g(x)))' = f'_g(g)\cdot g'(x).$$

Пример 1.1.2. Найдем производную функции $y(x) = \dfrac{\cos(x^2+1)}{\ln(1-2x)+3}$:

$$y'(x) = \frac{(\cos(x^2+1))'\cdot(\ln(1-2x)+3) - (\cos(x^2+1))\cdot(\ln(1-2x)+3)'}{(\ln(1-2x)+3)^2}.$$

Отдельно найдем производные:

$$(\cos(x^2+1))' = 2x\cdot(-\sin(x^2+1)).$$

$$(\ln(1-2x)+3)' = -2\cdot\frac{1}{1-2x}.$$

Окончательно имеем:

$$y'(x) = \frac{-2x\sin(x^2+1)\cdot(\ln(1-2x)+3) + (\cos(x^2+1))\cdot 2/(1-2x)}{(\ln(1-2x)+3)^2}.$$

Здесь стоит упомянуть про физический и геометрический смысл производной: ее можно интерпретировать как скорость изменения какой-либо величины (например, скорость материальной точки в механике) или как тангенс угла наклона касательной к графику функции. Именно из интерпретации производной как тангенса угла наклона касательной следуют важные прикладные теоремы, ради которых мы и ввели это понятие.

Теорема 1.1.1. *Функция примнимает свое локально максимальное или минимальное (экстремальное) значение только в тех точках, где ее производная равна нулю:*

$$y(x_0) \to \min \;\Rightarrow\; y'(x_0) = 0,$$

$$y(x_0) \to \max \;\Rightarrow\; y'(x_0) = 0.$$

Теорема 1.1.2. *Функция $y = f(x)$ строго возрастает на промежутке (a, b), тогда и только тогда, когда в каждой точке этого промежутка производная $f'(x)$ строго положительна:*

$$y = f(x) \nearrow,\ x \in (a, b) \Leftrightarrow\ \forall\ x \in (a, b) \hookrightarrow f'(x) > 0.$$

Теорема 1.1.3. *Функция $y = f(x)$ строго убывает на промежутке (a, b), тогда и только тогда, когда в каждой точке этого промежутка производная $f'(x)$ строго отрицательна:*

$$y = f(x) \searrow,\ x \in (a, b) \Leftrightarrow\ \forall\ x \in (a, b) \hookrightarrow f'(x) < 0.$$

1.2 Частные производные и градиент

Определение 1.2.1. Рассмотрим функцию f от n переменных $f : \mathbb{R}^n \longrightarrow \mathbb{R}$. Частная производная $f(x_1, x_2, \ldots, x_n)$ по переменной x_k – это предел:

$$f'_{x_k} = \frac{\partial}{\partial x_k} f(x_1, \ldots, x_k, \ldots, x_n) = \lim_{\Delta x \to 0} \frac{f(x_1, \ldots, x_k + \Delta x, \ldots, x_n) - f(x_1, \ldots, x_k, \ldots, x_n)}{\Delta x}.$$

Нахождение частных производных почти ничем не отличается от нахождения производной функции одной переменной (а часто даже и проще) – просто надо найти производную функции $f(x_k)$, считая все остальные переменные $x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n$ константами.

Пример 1.2.1. Найдем дифференциал функции $f(x, y, z) = x\cos(2y - z) + xy$ по формуле

$$df(x, y, z) = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial z}dz.$$

Для этого найдем частные производные:

$$\frac{\partial f}{\partial x} = f'_x = \cos(2y - z) + y,$$

$$\frac{\partial f}{\partial y} = f'_y = 2x \cdot (-\sin(2y - z)) + x,$$

$$\frac{\partial f}{\partial z} = f'_z = -x \cdot (-\sin(2y - z)).$$

Окончательно имеем:

$$df(x, y, z) = (\cos(2y - z) + y)dx + (-2x \cdot \sin(2y - z) + x)dy + (x \cdot \sin(2y - z))dz.$$

Определение 1.2.2. Градиентом функции $f(x_1, x_2, \ldots, x_n)$ от n переменных называется n-мерный вектор, составленный из частных прозводных функции f:

$$\operatorname{grad} f(x_1, x_2, \ldots, x_n) = \begin{pmatrix} f'_{x_1} \\ f'_{x_2} \\ \vdots \\ f'_{x_n} \end{pmatrix}.$$

Смысл градиента таков: это вектор, указывающий в направлении наибольшего возрастания функции $f(x_1, x_2, \ldots, x_n)$, значение которой меняется от одной точки пространства к другой, а по величине (модулю) равный скорости роста этой функции в этом направлении.

Пример 1.2.2. Найдем градиент функции $f(x_1, x_2, x_3, x_4) = x_2 \sin(x_1 x_3) + 2x_4$ в точке $(\frac{3\pi}{2}, 0, 1, -2)$. Для этого вычислим частные производные:

$$f'_{x_1} = x_2 x_3 \cos(x_1 x_3),$$

$$f'_{x_2} = \sin(x_1 x_3),$$

$$f'_{x_3} = x_2 x_1 \cos(x_1 x_3),$$

$$f'_{x_4} = 2.$$

Подставляя в частные производные соответствующие значения переменных, находим:

$$\operatorname{grad} f = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 2 \end{pmatrix}.$$

Теорема 1.2.1. *Функция $y = f(x_1, x_2, \ldots x_n)$ принимает свое локально экстремальное значение только в тех точках, где ее градиент обращается в ноль:*

$$f(x_1, x_2, \ldots x_n) \to \min \;\Rightarrow\; \operatorname{grad} f(x_1, x_2, \ldots x_n) = \vec{0},$$

$$f(x_1, x_2, \ldots x_n) \to \max \;\Rightarrow\; \operatorname{grad} f(x_1, x_2, \ldots x_n) = \vec{0}.$$

2 Основы линейной алгебры

2.1 Векторы и матрицы

Определение 2.1.1. Вектор – это упорядоченный одномерный массив чисел.

Мы будем обозначать векторы строчными латинскими буквами полужирным шрифтом и записывать их в виде столбцов:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}.$$

Определение 2.1.2. Матрица – это упорядоченный двумерный массив чисел.

Множество матриц с m строками и n столбцами, то есть матрицы размеров $m \times n$, будем обозначать $\underset{m\times n}{\mathbf{M}}$. Произвольную матрицу $A \in \underset{m\times n}{\mathbf{M}}$ будем обозначать заглавными латинскими буквами и записывать в следующем виде:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

где на пересечении i-той строки с j-тым столбцом будет находится элемент матрицы a_{ij}. То есть первый индекс элемента матрицы – номер строки, второй – номер столбца, в котором находится элемент.

Заметим, что вектор – частный случай матрицы: действительно, матрица размерами $m \times 1$ – в точности и есть вектор-столбец. Значит, если мы научимся работать с матрицами, то сразу же получим и правила работы с векторами. Поэтому далее пока что не будем отдельно рассматривать векторы, а будем работать с матрицами произвольных размеров $m \times n$, считая, что возможны случаи $n = 1$.

2.2 Сложение и вычитание матриц, умножение матриц на число

Определение 2.2.1. Пусть даны две матрицы $A, B \in \underset{m\times n}{\mathbf{M}} : A = (a_{ij})$, $B = (b_{ij})$. Матрица $C = (c_{ij})$ называется суммой матриц A и B, если $c_{ij} = a_{ij} + b_{ij}$ для любых $i = \overline{1, m}$ и $j = \overline{1, n}$.

Определение 2.2.2. Пусть даны две матрицы $A, B \in \underset{m\times n}{\mathbf{M}} : A = (a_{ij})$, $B = (b_{ij})$. Матрица $C = (c_{ij})$ называется разностью матриц A и B, если $c_{ij} = a_{ij} - b_{ij}$ для любых $i = \overline{1, m}$ и $j = \overline{1, n}$.

Заметим, что при таком определении мы можем складывать только матрицы одного размера.

Пример 2.2.1.

$$\begin{pmatrix} 1 & -4 & 0 \\ 5 & 3 & -1 \end{pmatrix} + \begin{pmatrix} 4 & 5 & -7 \\ 0 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 5 & 1 & -7 \\ 5 & 5 & 0 \end{pmatrix}.$$

Определение 2.2.3. Пусть дана матрица $A \in \underset{m\times n}{\mathbf{M}} : A = (a_{ij})$ и число λ. Матрица $C = (c_{ij})$ называется произведением матрицы A на число λ, если $c_{ij} = \lambda a_{ij}$ для любых $i = \overline{1,m}$ и $j = \overline{1,n}$.

Пример 2.2.2.

$$8 \cdot \begin{pmatrix} 1 & 7 \\ 2 & -3 \end{pmatrix} = \begin{pmatrix} 8 & 56 \\ 16 & -24 \end{pmatrix}.$$

Так как мы ввели сложение и вычитание матриц и умножение матриц на число покомпонентно, то все свойства чисел, связанные с этими операциями, остаются верными и для матриц.

2.3 Умножение матриц

Определение 2.3.1. Пусть даны две матрицы $A \in \underset{m\times n}{\mathbf{M}}$ и $B \in \underset{n\times p}{\mathbf{M}}$: $A = (a_{ij})$, $B = (b_{ij})$. Матрица $C = (c_{ij}) \in \underset{m\times p}{\mathbf{M}}$ называется произведением матриц A и B, если

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

для любых $i = \overline{1,m}$ и $j = \overline{1,n}$.

Из определения следует, что мы можем перемножать матрицы только тогда, когда количество столбцов у первой матрицы равно количеству строк у второй.

Пример 2.3.1. Найдем $C = AB$, где

$$A = \begin{pmatrix} 0 & -1 \\ 4 & 5 \\ -8 & 6 \end{pmatrix}, \; B = \begin{pmatrix} 3 & 1 \\ -2 & 0 \end{pmatrix}.$$

1. Найдем размер результирующей матрицы. Для этого запишем размеры перемножаемых матриц, в данном случае это 3×2 и 2×2. Убеждаемся, что количество столбцов у первой матрицы равно количеству строк у второй, значит, мы можем перемножить матрицы. Размер результирующей матрицы будет равен «количество строк первой матрицы $\times$ количество столбцов второй матрицы», то есть 3×2.
2. Последовательно найдем элементы результирующей матрицы $C = (c_{ij})$. Для этого обозначим матрицы $A = (a_{ij})$, $B = (b_{ij})$:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}, \; B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}.$$

Тогда по определению:

$$c_{11} = \sum_{k=1}^{2} a_{1k} b_{k1} = a_{11} b_{11} + a_{12} b_{21} = 0 \cdot 3 + (-1) \cdot (-2) = 2,$$

$$c_{12} = \sum_{k=1}^{2} a_{1k} b_{k2} = a_{11} b_{12} + a_{12} b_{22} = 0 \cdot 1 + (-1) \cdot 0 = 0,$$

$$c_{21} = \sum_{k=1}^{2} a_{2k}b_{k1} = a_{21}b_{11} + a_{22}b_{21} = 4 \cdot 3 + 5 \cdot (-2) = 2,$$

$$c_{22} = \sum_{k=1}^{2} a_{2k}b_{k2} = a_{21}b_{12} + a_{22}b_{22} = 4 \cdot 1 + 5 \cdot 0 = 4,$$

$$c_{31} = \sum_{k=1}^{2} a_{3k}b_{k1} = a_{31}b_{11} + a_{32}b_{21} = (-8) \cdot 3 + 6 \cdot (-2) = -36,$$

$$c_{32} = \sum_{k=1}^{2} a_{3k}b_{k2} = a_{31}b_{12} + a_{32}b_{22} = (-8) \cdot 1 + 6 \cdot 0 = -8.$$

3. Запишем ответ:

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 2 & 4 \\ -36 & -8 \end{pmatrix} = 2 \cdot \begin{pmatrix} 1 & 0 \\ 1 & 2 \\ -18 & -4 \end{pmatrix}.$$

Таким образом, можно вывести мнемоническое правило для произведения матриц: чтобы получить элемент c_{ij} результирующей матрицы, умножаем i-ую строку первой матрицы на j-ый столбец второй матрицы «почленно» и складываем.

Свойства умножения матриц:

1. Ассоциотивность: $A(BC) = (AB)C$.
2. Некоммутативность в общем случае: $AB \neq BA$.
3. Дистрибутивность: $A(B + C) = AB + AC$, $(A + B)C = AC + BC$.
4. Ассоциативность и коммутативность относительно умножения на число: $(\lambda A)B = \lambda(AB) = A(\lambda B)$.
5. $AE = EA = A$.
6. $AA = AA = A^2$.
7. $A^0 = E$.
8. Наличие делителей нуля: $AB = O \nRightarrow \begin{bmatrix} A = O \\ B = O \end{bmatrix}$

2.4 Транспонирование, скалярное произведение, длина вектора

Определение 2.4.1. Пусть дана матрица $A \in \underset{m \times n}{\mathbf{M}}: A = (a_{ij})$. Матрица $C = (c_{ij}) \in \underset{n \times m}{\mathbf{M}}$ называется транспонированной к A, если $c_{ij} = a_{ji}$ для любых $i = \overline{1, m}$ и $j = \overline{1, n}$. Будем обозначать транспонированную к A матрицу как A^T.

Пример 2.4.1.

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}.$$

То есть при транспонировании матрицы i-ая строка становится i-ым столбцом, а j-ый столбец стоновится j-ой строкой.

Свойства транспонирования:

1. $(A+B)^T = A^T + B^T$.
2. $(\lambda A)^T = \lambda A^T$.
3. $(AB)^T = B^T A^T$.

Определение 2.4.2. Скалярным произведением векторов

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

называется

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T \cdot \mathbf{b} = \begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \sum_{k=1}^{m} a_k b_k.$$

Пример 2.4.2. Найдем скалярное произведение векторов $\mathbf{a}^T = (-2 \ \ 6 \ \ 0 \ \ 5)$ и $\mathbf{b}^T = (4 \ \ 3 \ \ 8 \ \ -2)$. По определению имеем:

$$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{k=1}^{4} a_k b_k = (-2) \cdot 4 + 6 \cdot 3 + 0 \cdot 8 + 5 \cdot (-2) = 0.$$

Мы получили, что векторы $\mathbf{a}$ и $\mathbf{b}$ ортогональны.

Определение 2.4.3. Два ненулевых вектора называются ортогональными, если их скалярное произведение равно нулю.

Определение 2.4.4. Длина (модуль) вектора $\mathbf{a}^T = (a_1 \ \ a_2 \ \ \dots \ \ a_m)$ – квадратный корень из скалярного произведения этого вектора на себя, то есть:

$$a = |\mathbf{a}| = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle} = \sqrt{\sum_{k=1}^{m} (a_k)^2}.$$

Пример 2.4.3. Найдем длину вектора $\mathbf{a}^T = (-2 \ \ 6 \ \ 0 \ \ 5)$. По определению:

$$a = \sqrt{\sum_{k=1}^{4} (a_k)^2} = \sqrt{(-2)^2 + 6^2 + 0^2 + 5^2} = \sqrt{65}.$$

Определение 2.4.5. Скалярное произведение векторов $\mathbf{a}$ и $\mathbf{b}$ – это произведение модулей этих векторов на косинус угла α между ними:

$$\langle \mathbf{a}, \mathbf{b} \rangle = ab \cdot \cos(\alpha).$$

Мы привели два определения скалярного произведения векторов, что может вызывать некоторую путаницу – ведь определения очень разные. На самом деле эти определения эквивалентны (результат не будет зависить от способа вычисления), просто определение (2.4.5) более общее, а (2.4.2) удобно при практическом вычислении, если известны координаты перемножаемых векторов.

Определение 2.4.6. Следом квадратной матрицы $A \in \underset{n\times n}{\mathbf{M}}$ называется сумма элементов матрицы A, стоящих на главной диагонали:

$$\operatorname{tr} A = \sum_{i=1}^{n} a_{ii}.$$

2.5 Ранг и определитель матрицы

Определение 2.5.1. Система столбцов $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ из $\underset{m\times n}{\mathbf{M}}$ называется линейно зависимой, если некоторая их нетривиальная линейная комбинация равна нулевому столбцу, то есть если существуют числа $\lambda_1, \lambda_2, \ldots, \lambda_n$ такие, что хотя бы одно из этих чисел не нулевое (нетривиальность) и выполнено:

$$\sum_{k=1}^{n} \lambda_k \mathbf{a}_k = \vec{0} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Пример 2.5.1. Докажем, что следующие столбцы линейно зависимы:

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \begin{pmatrix} -4 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 7 \\ 9 \end{pmatrix}.$$

Действительно, существует нетривиальная линейная комбинация этих столбцов, равная нулевому вектору (напомним, что мы отождествляем понятие матрицы размера $m \times 1$ с вектором):

$$3 \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} -4 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} -1 \\ 7 \\ 9 \end{pmatrix} = \vec{0}.$$

Определение 2.5.2. Целое неотрицательное число r называется рангом матрицы A и обозначается $r = \operatorname{rg} A$, если из столбцов этой матрицы можно выбрать r линейно независимых столбцов, но нельзя выбрать $r + 1$ линейно независимых столбцов.

Свойства ранга:

1. $\operatorname{rg} A^T = \operatorname{rg} A$.
2. $\operatorname{rg} AB = \operatorname{rg} BA$, когда оба произведения существуют.
3. Если A – квадратная матрица и $|A| = 0$, то строки и столбцы матрицы линейно зависимы.
4. Линейная (не)зависимость столбцов матрицы эквивалентна линейной (не)зависимости строк.

5. Если все элементы строки или столбца матрицы умножить на отличное от нуля число, то ранг матрицы не изменится.

6. Ранг не изменится, если к элементам любой строки или столбца матрицы прибавить соответствующие элементы другой строки или столбца, умноженные на одно и то же число.

7. Ранг не изменится, если переставить два любых столбца или две строки матрицы.

Определение 2.5.3. Определитель (детерминант) квадратной матрицы $\underset{n\times n}{A} = (a_{ij})$ порядка n – это сумма по всем перестановкам из n элементов:

$$|\underset{n\times n}{A}| = \det \underset{n\times n}{A} = \sum_{\alpha_1 \dots \alpha_n} (-1)^{N(\alpha_1 \dots \alpha_n)} a_{1\alpha_1} \dots a_{n\alpha_n},$$

где $N(\alpha_1 \dots \alpha_n)$ – число инверсий в перестановке $(\alpha_1 \dots \alpha_n)$.

Так как количество перестановок (биекций множества в себя) на множестве из n элементов равно $n!$, то количество слагаемых в сумме при подсчете определителя через формулу полного разложения будет тоже равно факториалу от n.

Определение 2.5.4. Определитель – это полилинейная кососимметричная функция столбцов $\det : \underset{n\times n}{\mathbf{M}} \to \mathbb{R}$, такая, что детерминант единичной матрицы равен еденице: $\det(E) = 1$.

Как и в случае со скалярным произведением, мы ввели два эквивалентных определения детерминанта. Заметим, что детерминант определен только для квадратных матриц, то есть только для матриц размеров $n \times n$.

Пример 2.5.2. Найдем определитель матрицы: $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$.

1. По первому определению:
   $|A| = \sum_{\alpha_1,\alpha_2} (-1)^{N(\alpha_1,\alpha_2)} a_{1\alpha_1} \cdot a_{n\alpha_2} = (-1)^{N(1,2)} a_{11} \cdot a_{22} + (-1)^{N(2,1)} a_{12} \cdot a_{21} = 1 \cdot 4 - 2 \cdot 3 = -2.$

2. По второму определению:
   $\det(A) = \det\left[\begin{pmatrix}1\\3\end{pmatrix}, \begin{pmatrix}2\\4\end{pmatrix}\right] = \det\left[\begin{pmatrix}1\\0\end{pmatrix} + 3\begin{pmatrix}0\\1\end{pmatrix},\ 2\begin{pmatrix}1\\0\end{pmatrix} + 4\begin{pmatrix}0\\1\end{pmatrix}\right] = \det\left[\begin{pmatrix}1\\0\end{pmatrix},\ 2\begin{pmatrix}1\\0\end{pmatrix}\right] +$
   $\det\left[\begin{pmatrix}1\\0\end{pmatrix},\ 4\begin{pmatrix}0\\1\end{pmatrix}\right] + \det\left[3\begin{pmatrix}0\\1\end{pmatrix},\ 2\begin{pmatrix}1\\0\end{pmatrix}\right] + \det\left[3\begin{pmatrix}0\\1\end{pmatrix},\ 4\begin{pmatrix}0\\1\end{pmatrix}\right] = 4\det\left[\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right] -$
   $6\det\left[\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right] = 4\det(E) - 6\det(E) = -2.$

На практике для вычисления определителей обычно используют следствие из определений, а именно формулу разложения по i-ой строке (чаще всего по первой):

$$|\underset{n\times n}{A}| = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} M_{ij},$$

где M_{ij} – дополнительный минор к элементу матрицы a_{ij}, то есть определитель матрицы, которая получается из исходной при вычеркивании i-ой строки и j-того столбца.

Для матриц второго и третьего порядка определитель выражается следующим образом (разложение по первой строке):

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}.$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}.$$

Пример 2.5.3. Найдем определитель:

$$|A| = \det A = \begin{vmatrix} 0 & 1 & 3 \\ 2 & 3 & 5 \\ 3 & 5 & 7 \end{vmatrix} = 0 \cdot \begin{vmatrix} 3 & 5 \\ 5 & 7 \end{vmatrix} - 1 \cdot \begin{vmatrix} 2 & 5 \\ 3 & 7 \end{vmatrix} + 3 \cdot \begin{vmatrix} 2 & 3 \\ 3 & 5 \end{vmatrix} = 0 - (14 - 15) + 3 \cdot (10 - 9) = 4.$$

Мы получили, что наша матрица не вырождена.

Определение 2.5.5. Матрица называется вырожденой, если ее определитель равен нулю.

Свойства детерминанта:

1. $|A^T| = |A|$.
2. $|AB| = |A| \cdot |B|$.
3. Умножение всех элементов одной строки или столбца определителя на некоторое число равносильно умножению определителя на это число.
4. Если матрица содержит нулевую строку или столбец, то определитель этой матрицы равен нулю.
5. Если хотя бы две строки или два столбца матрицы линейно зависимы, то определитель этой матрицы равен нулю.
6. При перестановке двух любых строк или столбцов определитель матрицы меняет знак.
7. Определитель не изменится, если к элементам любой строки или столбца матрицы прибавить соответствующие элементы другой строки или столбца, умноженные на одно и то же число.
8. Определитель матрицы треугольного вида равен произведению элементов, стоящих на главной диагонали:

$$|A| = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{vmatrix} = \prod_{k=1}^{n} a_{kk} = a_{11} \cdot a_{22} \cdot \ldots \cdot a_{nn}.$$

Заметим, что последнее свойство позволяет легко находить определитель матриц большого порядка, если перед этим привести матрицу к диоганальному виду (что всегда можно сделать, опираясь на предпоследнее свойство и используя алгоритм Гаусса).

2.6 Обратная матрица

Определение 2.6.1. Матрица A^{-1} называется обратной к квадратной невырожденной матрице A, если $AA^{-1} = A^{-1}A = E$.

Пример 2.6.1. Найдем обратную матрицу к

$$A = \begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix}.$$

Решим задачу несколькими способами.

1. Воспользуемся методом Гаусса. Он заключается в следующем: к матрице, к которой ищется обратная, справа приписывается через черту единичная матрица, после этого, считая исходную матрицу и приписанную единичную одной матрицей, элементарными преобразованиями (прямым и обратным ходом метода Гаусса) исходную матрицу приводим к диоганальному виду, а потом к единичному. Получившаяся справа матрица и будет обратной к исходной. В нашем случае:

$$\left(\begin{array}{cc|cc} 2 & 5 & 1 & 0 \\ 1 & 3 & 0 & 1 \end{array}\right) \xrightarrow{II=I-2II} \left(\begin{array}{cc|cc} 2 & 5 & 1 & 0 \\ 0 & -1 & 1 & -2 \end{array}\right) \xrightarrow{I=I+5II} \left(\begin{array}{cc|cc} 2 & 0 & 6 & -10 \\ 0 & -1 & 1 & -2 \end{array}\right) \xrightarrow[II=-II]{I=0{,}5\cdot I} \left(\begin{array}{cc|cc} 1 & 0 & 3 & -5 \\ 0 & 1 & -1 & 2 \end{array}\right).$$

   Запись над стрелкой $II = I - 2II$ означает, что на следующем шаге вместо второй строки будет записана разность первой строки и удвоенной второй.

2. Воспользуемся методом неопределенных коэффициентов. Пусть искомая матрица имеет вид:

$$A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

   тогда по определению должно выполнятся

$$AA^{-1} = \begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = E.$$

   Перемножая матрицы, имеем систему:

$$\begin{cases} 2a + 5c = 1, \\ 2b + 5d = 0, \\ a + 3c = 0, \\ b + 3d = 1. \end{cases}$$

   Решая, находим, что $a = 3,\ b = -5,\ c = -1,\ d = 2.$

3. Воспользуемся явной формулой для обратной матрицы:

$$A^{-1} = \frac{1}{|A|} A_+^T,$$

   где A_+ – матрица алгебраических дополнений соответствующих элементов матрицы A, это значит, что элементы матрицы A_+ равны $a_{+ij} = (-1)^{i+j} M_{ij}$, где M_{ij} – дополнительный минор к элементу a_{ij}, то есть определитель матрицы, получающийся из исходной вычеркиванием i-ой строки и j-того столбца. Тогда согласно этой формуле:

$$A^{-1} = \frac{1}{1} \begin{pmatrix} 3 & -1 \\ -5 & 2 \end{pmatrix}^T = \begin{pmatrix} 3 & -5 \\ -1 & 2 \end{pmatrix}.$$

Итак, мы нашли, что

$$\begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix}^{-1} = \begin{pmatrix} 3 & -5 \\ -1 & 2 \end{pmatrix}.$$

Заметим, что из формулы для обратной матрицы можно получить простое мнемоническое правило отыскания обратных матриц для матриц размером 2×2: надо просто поменять местами элементы на главной диоганали и домножить на -1 элементы на побочной диоганали, разделив полученную матрицу на детерминант исходной, мы и получим обратную матрицу:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

Свойства обратной матрицы:

1. $(AB)^{-1} = B^{-1}A^{-1}$.
2. $|A^{-1}| = |A|^{-1}$.
3. $(A^T)^{-1} = (A^{-1})^T$.
4. $(kA)^{-1} = k^{-1}A^{-1}$.
5. $E^{-1} = E$.

2.7 Матрично-векторное дифференцирование

Определение 2.7.1. Пусть задано отображение $f : D \to E$, определим производную $\frac{\partial f}{\partial x} \in G$:

	D	E	G	Название
1	$\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$	Производная
2	$\mathbb{R}^n$	$\mathbb{R}$	$\mathbb{R}^n$	Градиент
3	$\mathbb{R}^n$	$\mathbb{R}^m$	$\mathbb{R}^{n\times m}$	Матрица Якоби
4	$\mathbb{R}^{n\times m}$	$\mathbb{R}$	$\mathbb{R}^{n\times m}$	Производная функции по матрице

Матрица Якоби отображения $f : \mathbb{R}^n \to \mathbb{R}^m$ – это:

$$\frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}.$$

Производная функции $f : \mathbb{R}^{n\times m} \to \mathbb{R}$ по матрице $A = (a_{ij}) \in \underset{m\times n}{\mathbf{M}}$ – это матрица, состоящая из частных производных функции f по элементам матрицы A:

$$\frac{\partial f}{\partial A} = \begin{pmatrix} \frac{\partial f}{\partial a_{11}} & \frac{\partial f}{\partial a_{12}} & \cdots & \frac{\partial f}{\partial a_{1n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial a_{m1}} & \frac{\partial f}{\partial a_{m2}} & \cdots & \frac{\partial f}{\partial a_{mn}} \end{pmatrix}.$$

Определение 2.7.2. Для матриц $A, B \in \underset{m\times n}{\mathbf{M}}$ стандартное (фробениусово) скалярное произведение определяется следующим образом:

$$\langle A, B \rangle = \operatorname{tr}(A^T B) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij}.$$

3 Некоторые понятия теории вероятностей и математической статистики

3.1 Определение вероятности

Определение 3.1.1. Вероятностное пространство – это тройка $(\Omega, \mathcal{F}, P)$, где:

1. $\Omega = \{\omega_1, \omega_2, \ldots\}$ – произвольное непустое множество элементарных исходов.
2. $\mathcal{F} \subseteq 2^\Omega$ – множество случайных событий, состоящее из подмножеств Ω, такое, что:
   2.1. $\Omega \in \mathcal{F}$.
   2.2. $A \in \mathcal{F} \Rightarrow \Omega \setminus A = \overline{A} \in \mathcal{F}$.
   2.3. $\{A_i\}_{i=1}^{\infty} \in \mathcal{F} \Rightarrow \bigcup\limits_{i=1}^{\infty} A_i \in \mathcal{F}$.
3. $P : \mathcal{F} \to [0, 1]$ – функция, такая, что:
   3.1. $P(\Omega) = 1$.
   3.2. $\forall\, \{A_i\}_{i=1}^{\infty} \in \mathcal{F} : A_i \cap A_j = \varnothing\ \forall\, i \neq j \ \Rightarrow P\left(\bigsqcup\limits_{i=1}^{\infty} A_i\right) = \sum\limits_{i=1}^{\infty} P(A_i)$.

При этом $\mathcal{F}$ называется σ-алгеброй событий (так как это множество замкнуто относительно счетного объединения), а P – σ-аддитивной вероятностной мерой.

Такой подход к построению теории вероятностей называется аксиоматическим и, вообще говоря, избыточен для большинства прикладных задач. Мы чаще будем пользоваться следующим классическим определением вероятности.

Определение 3.1.2. Пусть задано конечное множество взаимоисключающих элементарных исходов $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$. Событием A называется некоторое подмножество множества элементарных исходов: $A \subset \Omega$. Вероятностью события A называется доля подмножества A в множестве Ω:

$$P(A) = \frac{|A|}{|\Omega|}.$$

Пример 3.1.1. Рассмотрим опыт с бросанием игральной кости.
1) Вероятностное пространство – это множество всевозможных элементарных исходов:

$$\Omega = \{\omega_i \mid \omega_i = \{\text{выпала } i\text{-ая грань}\},\ i = \overline{1, 6}\}.$$

2) Так как в данном эксперименте естественно предположить, что выпадение любой грани равновероятно (то есть наше вероятностное пространство однородное), то с учетом условия нормировки получается, что любая i-ая грань выпадет с вероятностью:

$$P(\omega_i) = \frac{1}{6}.$$

3) Пример события – выпала четная грань. Действительно, выпадение четной грани есть подмножество $A = \{\omega_2, \omega_4, \omega_6\}$ всех элементарных исходов Ω. Тогда вероятность такого события:

$$P(A) = \frac{|A|}{|\Omega|} = \frac{3}{6} = \frac{1}{2}.$$

3.2 Совместная вероятность и теорема Байеса

Определение 3.2.1. Совместная вероятность событий A и B – это вероятность одновременного наступления этих событий $P(AB)$.

Определение 3.2.2. Два события A и B называются независимыми, если $P(AB) = P(A)P(B)$.

Обратите внимание, что независимость событий в теории вероятностей определяется сугубо формально. Ее не надо путать ни с отношением «причина-следствие», которого между зависимыми случайными величинами может и не быть, ни с корреляцией, которая отражает только линейную часть зависимости между случайными величинами.

Определение 3.2.3. Условная вероятность – вероятность наступления события A, если известно, что произошло событие B, которое имеет положительную вероятность:

$$P(A|B) = \frac{P(AB)}{P(B)}.$$

Определение 3.2.4. События A и B условно независимы при условии C, если выполнено $P(AB \mid C) = P(A|C)P(B|C)$.

Из этих определений следует важнейшая для машинного обучения и анализа данных теорема – теорема Байеса.

Теорема 3.2.1 (Байеса).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

Доказательство.

$$P(A|B) = \frac{P(AB)}{P(B)}, \quad P(B|A) = \frac{P(AB)}{P(A)}.$$

Выразим и приравняем из первого и второго равенства $P(A, B)$:

$$P(A|B)P(B) = P(B|A)P(A).$$

Из последнего равенства следует утверждение теоремы. □

Чтобы из совместной вероятности получить вероятность того или иного исхода одной из случайных величин, можно воспользоваться формулой полной вероятности, то есть просуммировать одну случайную величину по другой:

$$P(A) = \sum_B P(A|B)P(B).$$

Пример 3.2.1. Рассмотрим классическую задачу на применение теоремы Байеса. Предположим, что некий тест на какую-нибудь страшную болезнь с 95% точностью определяет, болен ли человек. Предположим так же, что болезнь достаточно распространена и имеется у 1% респондентов. Пусть некоторый человек получил позитивный результат теста, то есть, тест говорит, что страшная болезнь у человека присутствует. С какой вероятностью выбранный человек действительно болен?

Пусть событие A – результат теста положителен, событие B – человек действительно болен. Нам надо найти вероятность, что человек болен, при условии, что результат теста положителен, то есть $P(B|A)$. По теореме Байеса:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}.$$

$P(A|B) = 95\%$ – вероятность того, что результат теста положителен, при условии, что болезнь есть. $P(B) = 1\%$ – вероятность того, что человек болен. Чтобы найти $P(A)$ воспользуемся формулой полной вероятности, учтя, что для B возможны всего два исхода: B и $\overline{B}$ – человек либо болен, либо здоров, тогда:

$$P(A) = \sum_B P(A|B)P(B) = P(A|B)P(B) + P(A|\overline{B})P(\overline{B}) = 95 \cdot 1 + 5 \cdot 99 = 590.$$

Окончательно имеем:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{95 \cdot 1}{590} \approx 0{,}161 \approx 16\%.$$

3.3 Случайные величины

Определение 3.3.1. Случайная величина ξ – это любая функция, сопоставляющая элементарному исходу из множества Ω некоторое вещественное число, то есть:

$$\xi : \Omega \longrightarrow \mathbb{R}.$$

Очевидно, что по определению функции, случайная величина в отдельной своей реализации может принимать только одно из значений $\xi \in \{x_1, x_2, ...\}$ (условимся для удобства записывать значения случайной величины в порядке возрастания: $x_i < x_{i+1}$).

Определение 3.3.2. Вероятность того, что ξ принимает некоторое значение x_k равна сумме вероятностей событий ω_i для которых $\xi(\omega_i) = x_k$:

$$P_\xi(k) = \sum_{\xi(\omega_i)=x_k} P(\omega_i),$$

где за $P_\xi(k)$ обозначена вероятность того, что ξ принимает значение x_k, то есть величина $P(\xi = x_k)$.

Таким образом, случайную величину удобно интерпритировать как результат какого-либо случайного события, то есть $\omega_i \longrightarrow \xi(\omega_i)$.

Обратим внимание на то, что под ξ в зависимости от контекста мы можем понимать разные вещи. Запись $\xi(\omega_i) = x_j$ означает, что случайная величина ξ ставит в соответствие элементарному исходу ω_i вещественное число x_j, и здесь ξ – функция. Так же под ξ мы можем понимать само значение, которое ставится в соответствие какому-то элементарному исходу.

Пример 3.3.1. Рассмотрим опыт с бросанием уже двух разных игральных костей (что на самом деле эквивалентно двум последовательным бросанием одной игральной кости). Пусть случайная величина – это сумма очков, выпавшая на костях. Тогда очевидно, что $\xi \in \{2, 3, \ldots, 12\}$.

Найдем вероятность того, что $\xi = 7$: это реализуется при элементарных исходах (1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1), где на первом месте записано количество очков, выпавших на первой кости, а на втором – количество очков, выпавших на второй кости. Вероятность любого исхода (i, j) равна $1/36$ при $i, j = \overline{1, 6}$. Тогда по определению:

$$P(\xi = 7) = 6 \cdot \frac{1}{36} = \frac{1}{6}.$$

3.4 Математическое ожидание и дисперсия

Определение 3.4.1. Математическое ожидание случайной величины – это сумма произведений вероятности того, что случайная величиная принимает некоторое значение, на это значение:

$$E(\xi) = \sum_{i=1}^{|\Omega|} P(\xi_i)\xi(\omega_i)$$

По сути матожидание – это некое среднее значение случайной величины при достаточно большом количестве испытаний.

Свойства математического ожидания:

1. $E(c\xi) = cE(\xi)$.
2. $E(\xi + \eta) = E(\xi) + E(\eta)$.
3. $\xi \geqslant 0 \Rightarrow E(\xi) \geqslant 0$.
4. $\forall\, i, j \hookrightarrow \xi_i \geqslant \eta_j \Rightarrow E(\xi) \geqslant E(\eta)$.

Пример 3.4.1. Рассмотрим следующую игру. В коробке лежит 10 фишек: 5 штук со зачением -5, 4 штуки со значением 2,5 и 1 фишка со значением 10. Предлагается вытащить одну из фишек и, если на фишке написано положительное число, то ваш выйгрыш – очки на фишке, умноженные на 10, но если вы вытащили фишку с отрицательным значением, то вы должны отдать 50 рублей.

Посчитаем матожидание выйгрыша. Для этого составим закон распределения случайной величины ξ – выйгрыша:

ξ_i	-5	2,5	10
$P(\xi_i)$	0,5	0,4	0,1

По определению математического ожидания:

$$E(\xi) = \sum_{i=1}^{|\Omega|} P(\xi_i)\xi_i = \sum_{i=1}^{3} P(\xi_i)\xi_i = 0{,}5 \cdot 5 + 0{,}4 \cdot 2{,}5 + 0{,}1 \cdot 10 = -0{,}5.$$

Таким образом, математическое ожидание данной игры проигрышно, то есть, сыграв много раз в эту игру, вы уйдете, оказавшись в минусе в среднем на 5 рублей.

Определение 3.4.2. Дисперсией случайной величины называется величина

$$D(\xi) = E[(\xi - E(\xi))^2].$$

Дисперсия позволяет количественно оценить, насколько сильно конкретное значение случайной величины отличается от среднего значения, то есть насколько значения рассеяны, а рассеяние с латыни переводится не иначе, как дисперсия (поэтому этот же термин используется в оптике, но означает, конечно, совсем другое).

Утверждение 3.4.1. *Для дисперсии справедливо следующее соотношение:*

$$D(\xi) = E(\xi^2) - E(\xi)^2.$$

Доказательство. Используя линейность матожидания, имеем:

$$D(\xi) = E[(\xi - E(\xi))^2] = E[\xi^2 - 2E(\xi)\xi + E(\xi)^2] = E(\xi^2) - 2E(\xi)^2 + E(\xi)^2 = E(\xi^2) - E(\xi)^2.$$

□

Свойства дисперсии:

1. $D(\xi) = E(\xi^2) - E(\xi)^2$.
2. $D(c\xi) = c^2 D(\xi)$.
3. $D(\xi) \geqslant 0$.
4. $D(\xi) = 0 \Leftrightarrow \forall\, i \hookrightarrow \xi(\omega_i) = E(\xi)$.
5. $D(\xi + \eta) \geqslant D(\xi) + D(\eta)$.

Первое свойство имеет важное прикладное значение. Именно по этой формуле чаще всего и считают дисперсию, так как это проще, чем по определению (в этом мы сможем убедиться на последующем примере).

Определение 3.4.3. Среднеквадратичное отклонение (стандартное отклонение) случайной величины – это квадратный корень из дисперсии:

$$\sigma(\xi) = \sqrt{D(\xi)}.$$

Среднеквадратичное отклонение характеризует отклонение случайной величины от ее математического ожидания.

Пример 3.4.2. Вернемся к предыдущему примеру. Там мы нашли неутешительное математическое ожидание $E(\xi) = -0{,}5$ нашей игры, и сейчас нам предстоит вычислить её дисперсию двумя способами: по определению и по первому свойству «дисперсия – это матожидание квадрата минус квадрат матожидания». Для поиска дисперсии первым способом составми таблицу:

ξ_i	-5	2,5	10
$P(\xi_i)$	0,5	0,4	0,1
$\xi_i - E(\xi)$	-4,5	3	10,5
$(\xi_i - E(\xi))^2$	20,25	9	110,25
$(\xi_i - E(\xi))^2 \cdot P(\xi_i)$	10,125	3,6	11,025

Суммируя значения последней строки, получаем дисперсию:

$$D(\xi) = 10{,}125 + 3{,}6 + 11{,}025 = 24{,}75.$$

Чтобы найти дисперсию по первому свойству, необходимо вычислить величины:

$$E(\xi^2) = \sum_{i=1}^{3} P(\xi_i)\xi_i^2 = 0{,}5 \cdot 25 + 0{,}4 \cdot 6{,}25 + 0{,}1 \cdot 100 = 25.$$

$$E(\xi)^2 = \left(\sum_{i=1}^{3} P(\xi_i)\xi_i\right)^2 = (0{,}5 \cdot (-5) + 0{,}4 \cdot 2{,}5 + 0{,}1 \cdot 10)^2 = (-0{,}5)^2 = 0{,}25.$$

Тогда, по формуле $D(\xi) = E(\xi^2) - E(\xi)^2$ получим такой же результат, как и в первом случае:

$$D(\xi) = 25 - 0{,}25 = 24{,}75.$$

Из этого примера видно, что последний способ вычисления дисперсии действительно проще, чем подсчет дисперсии по определению.

Размерность дисперсии – рубли в квадрате, поэтому сложно по дисперсии оценить разброс выйгрыша или проигрыша в игре. А вот по стандартному отклонению уже становится яснее, насколько сильно разбросаны эти значения относительно вычесленного матожидания:

$$\sigma(\xi) = \sqrt{D(\xi)} = \sqrt{24{,}75} \approx 5.$$

3.5 Функция и плотность распределения

Определение 3.5.1. Функцией распределения случайной величины ξ называется функция $F_\xi : \mathbb{R} \to [0, 1]$ такая, что $F_\xi(x) = P(\xi < x)$.

Свойства функции распределния:

1. $F_\xi(x)$ монотонно неубывающая функция.
2. $\lim\limits_{x\to-\infty} F_\xi(x) = 0$, $\lim\limits_{x\to+\infty} F_\xi(x) = 1$.
3. $F_\xi(x)$ непрерывна слева: $\forall\, x \in \mathbb{R} \lim\limits_{t\to x-0} F_\xi(t) = F_\xi(x)$.
4. $F_\xi(x)$ имеет предел справа.
5. $F_\xi(x)$ имеет разрывы только первого рода, и их не более чем счетное количество.

Теорема 3.5.1. *Для того, чтобы функция $F : \mathbb{R} \to [0, 1]$ являлась функцией распределения, необходимо и достаточно, чтобы она удовлетворяла следующим условиям:*

1. *$F(x)$ монотонно неубывает на $\mathbb{R}$.*
2. *$\lim\limits_{x\to-\infty} F(x) = 0$ и $\lim\limits_{x\to+\infty} F(x) = 1$.*
3. *$F(x)$ непрерывна слева: $\forall\, x \in \mathbb{R} \lim\limits_{t\to x-0} F(t) = F(x)$.*

Определение 3.5.2. Распределение случайной величины ξ называется дискретным, если ξ принимает конечное или счетное число значений $x_1, x_2, \ldots$ таких, что:

$$p_k = P(\xi = x_k) > 0, \quad \sum_k p_k = 1.$$

Определение 3.5.3. Распределение P случайной величины ξ назыввется абсолютно непрерывным, если существует такая неотрицательная функция $f(x)$ (которая называется плотностью распределения случайной величины ξ), что для любого борелевского множества B существует интеграл Лебега от этой функции $f(x)$, такой, что:

$$P(x \in B) = \int\limits_B f(x)dx.$$

Теорема 3.5.2. *Если функция $f : \mathbb{R} \to \mathbb{R}$ удовлетворяет условиям*

1. $\forall\, x \in \mathbb{R}\ f(x) \geq 0$,

2. $\int\limits_{-\infty}^{+\infty} f(x)dx = 1$,

то существует распределение $F(x)$ такое, что $f(x)$ является его плотностью.

3.6 Основные дискретные распределения

Рассмторим последовательность из n независимых испытаний с двумя исходами: A и $\overline{A}$, которые назовем соответственно «успех» и «неудача», причем $P(A) = p \in (0,1)$, $P(\overline{A}) = 1 - p$. Такая схема испытаний называется схемой Бернулли, а сам опыт – опыт Бернулли.

Из комбинаторных соображений нетрудно получить, что при проведении опыта G по схеме Бернулли вероятность $P_n(k)$ события $A_n(k)$, состоящего в том, что при n повторениях опыта G событие A произойдет ровно k раз, равна:

$$P_n(k) = C_n^k p^k (1-p)^{n-k}.$$

Определение 3.6.1. Дискретная случайная величина ξ с реализациями $x_k = k$, $k = \overline{0,n}$ имеет биномиальное распределение с параметрами n – количество возможных исходов и $p \in (0,1)$, что записывается как $\xi \sim Bin(n,p)$, если вероятность события $\xi = x_k$ определяется формулой Бернулли:

$$P(\xi = x_k) = P_k = C_n^k p^k (1-p)^{n-k} = C_n^k p^k q^{n-k},\ \ q = 1-p.$$

Здесь $C_n^k = \dfrac{n!}{k!(n-k)!} = \dbinom{n}{k}$ – биномиальные коэффициенты, то есть коэффициенты в разложении бинома Ньютона:

$$(a+b)^n = \sum_{k=0}^{n} C_n^k a^k b^{n-k} = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}.$$

Математическое ожидание и дисперсия биномиального распределния равны $E(\xi) = np$, $D(\xi) = npq = np(1-p)$.

Пример 3.6.1. Пусть монету подбрасывают три раза. Требуется найти ряд распределения (закон распределения, записанный в порядке возрастания случайной величины) числа ξ выпавших гербов.

Случайная величина ξ распределена по биномиальному закону с параметрами $n = 3$ и $p = 1/2$, поэтому ξ может принимать значения 0, 1, 2, 3 с вероятностями

$$P_0 = C_3^0 \left(\frac{1}{2}\right)^3 = \frac{1}{8},\ P_1 = P_2 = C_3^1 \left(\frac{1}{2}\right)^3 = \frac{3}{8},\ P_3 = C_3^3 \left(\frac{1}{2}\right)^3 = 1 - (P_0 + P_1 + P_2) = \frac{1}{8}.$$

Таким образом, получаем следующий ряд распределения числа выпавших гербов:

ξ	0	1	2	3
P	1/8	3/8	3/8	1/8

Определение 3.6.2. Биноминальное распределение $Bin(1,p)$ с параметрами $n = 1$ и $p \in (0,1)$ называется распределением Бернулли и обозначается $Ber(p)$.

Определение 3.6.3. Дискретная случайная величина ξ с реализациями $x_k = k$, $k \in \mathbb{N}$ имеет распределение Пуассона с параметром $\lambda > 0$, что записывается как $\xi Pois(\lambda)$, если

$$P(\xi = x_k) = P_k = \frac{\lambda^k}{k!} e^{-\lambda}.$$

Математическое ожидание и дисперсия распределния Пуассона равны между собой и равны параметру λ: $E(\xi) = D(\xi) = \lambda$.

Теорема 3.6.1 (Пуассона). *Пусть $n \to \infty$ и $p \to 0$ и при этом выполнено $np = \lambda =$ const. Тогда*

$$\lim_{n\to\infty} P_n(k) = \frac{\lambda^k}{k!} e^{-\lambda}, \text{ где } P_n(k) = C_n^k p^k (1-p)^{n-k}, \ k = \overline{0, n}.$$

Определение 3.6.4. Геометрическое распределение – распределение дискретной случайной величины, равной количеству испытаний случайного эксперимента до наблюдения первого «успеха», то есть $\xi \sim Geom(n, p)$, если $P(\xi = k) = p(1-p)^k$, $k \in \mathbb{N}$.

3.7 Основные непрерывные распределения

Определение 3.7.1. Случайная величина ξ распределена на отрезке $[a, b]$ равномерно, что записыватся как $\xi \sim R(a, b)$, если плотность вероятности ξ имеет вид

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & x \in [a, b] \\ 0, & x \notin [a, b] \end{cases}$$

Определение 3.7.2. Случайная величина ξ имеет экспоненциальное (показательное) распределение с параметром $\lambda > 0$, что обозначается как $\xi \sim Exp(\lambda)$, если плотность вероятности ξ имеет вид:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0, & x \le 0 \end{cases}$$

Определение 3.7.3. Случайная величина ξ имеет нормальное (гауссовское) распределение с параметрами m и $\sigma^2 > 0$, что обозначается как $\xi \sim N(m, \sigma^2)$, если плотность вероятности ξ имеет вид:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}}.$$

3.8 Парная линейная регрессия

Определение 3.8.1. Ковариация двух случайных величин ξ и η, определенных в одном и том же вероятностном пространстве – это

$$\operatorname{cov}(\xi, \eta) = E[(\xi - E(\xi))(\eta - E(\eta))].$$

Ковариация – обобщение понятия дисперсии. Действительно, дисперсия есть ковариация случайной величины с самой собой: $\operatorname{cov}(\xi, \xi) = E[(\xi - E(\xi))(\xi - E(\xi))] = E[(\xi - E(\xi))^2] = D(\xi)$.

Утверждение 3.8.1. *Для ковариации справедливо следующее соотношение:*

$$\operatorname{cov}(\xi, \eta) = E(\xi\eta) - E(\xi)E(\eta).$$

Доказательство. Как и в доказательстве аналогичного свойства для дисперсии, используем линейность матожидания:

$$\operatorname{cov}(\xi,\eta) = E[(\xi - E(\xi))(\eta - E(\eta))] = E[\xi\eta - \xi E(\eta) - \eta E(\xi) + E(\xi)E(\eta)] = \\ E(\xi\eta) - E(\xi)E(\eta) - E(\eta)E(\xi) + E(\xi)E(\eta) = E(\xi\eta) - E(\xi)E(\eta).$$

□

Определение 3.8.2. Линейный коэффициент корреляции Пирсона двух случайных величин ξ и η – это

$$r(\xi,\eta) = \frac{\operatorname{cov}(\xi,\eta)}{\sigma(\xi)\sigma(\eta)}.$$

Линейный коэффициент корреляции изменяется от минус одного до одного и показывает, насколько сильна линейная зависимость величин: если $r(\xi,\eta)$ близок к еденице, то между случайными величинами ξ и η прямая линейная пропорциональность (чем больше ξ, тем больше η), а если $r(\xi,\eta)$ порядка минус одного, то связь между ξ и η линейная, но с тем свойством, что чем больше ξ, тем меньше η.

Рассмотрим следующую задачу (задачу линейной парной регрессии). Пусть у объекта есть пара признаков (x, y). Пусть так же нам удалось набрать статистику: зависимость одного признака от другого, вид которой изображен на рисунке 1.

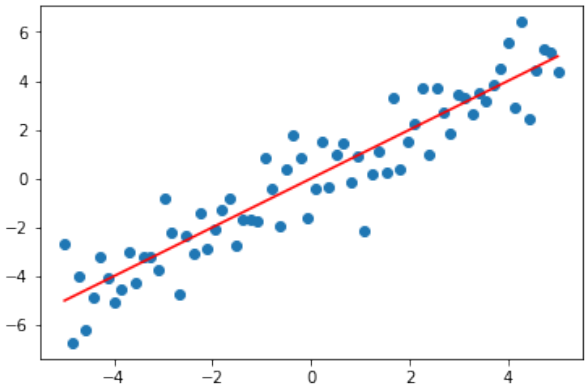

Рис. 1: Иллюстрация парной регрессии.

Из графика видно, что, хотя точки и не ложатся на прямую, все же можно провести прямую, которая будет отражать зависимость $y(x)$. Встает вопрос – как провести эту прямую? Давайте из всех прямых выделим такую, которая лучше всего будет аппроксимировать нашу зависимость. Чтобы это сделать, надо определиться с так называемой функцией потерь, которая будет количественно отражать, насколько хорошо проведена прямая. Первое, что приходит на ум – просто взять среднее от суммы модулей расстояний от каждой точки до прямой (Mean Absolute Error):

$$L = \frac{1}{n}\sum_{i=1}^{n} |(y(x_i) - \hat{y}(x_i))|,$$

где $\hat{y}(x_i) = kx_i + b$ – искомая прямая, а $y(x_i) = kx_i + b + \varepsilon_i$ – то, что нам удалось померить (данные, на основе которых мы будем строить уравнение регрессии). Такая функция ошибок тоже используются, но чаще в качестве штрафа за неправильное предсказание берут сумму квадратов расстояний от каждой точки до прямой (Mean Squared Error). Этот выбор обусловлен теоремой Гаусса-Маркова.

Теорема 3.8.1 (Гаусса-Маркова). *Пусть в модели парной регрессии наблюдения y связаны с x зависимостью: $y_i = kx_i + b + \varepsilon_i$, и при этом выполнены следующие условия:*

1. *Модель данных правильно специфицирована*
2. *Все x_i детерминированы и не все равны между собой*
3. *Ошибки не носят систематического характера, то есть $E(\varepsilon_i) = 0 \ \forall\, i$*
4. *Дисперсия всех ошибок одинакова*
5. *Ошибки некоррелированы, то есть $\operatorname{cov}(\varepsilon_i, \varepsilon_j) = 0 \ \forall\, i, j$*

Тогда в этих условиях оценки метода наименьших квадратов (МНК) оптимальны в классе линейных несмещённых оценок для поиска уравнения регресси $y = kx + b$.

Таким образом, наша функция ошибок имеет вид:

$$L(k, b) = \frac{1}{n}\sum_{i=1}^{n}(y(x_i) - \hat{y}(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}(y(x_i) - kx_i - b)^2.$$

Заметим, что функция L не имеет максимума – как бы «плохо» мы не провели прямую (какое бы большое значение не принимала бы L), можно провести ее еще «хуже» (сделать значение L еще больше). А значит, если мы найдем экстремум функции L, то это будет минимум, а это как раз то, что нам нужно – в этом случае сумма растояний от точек до прямой будет минимальна.

Из математического анализа известно, что функция нескольких переменных принимает свое экстремальное значение только в тех точках, где ее градиент равен нулю. Из этого условия получаем систему линейных уравнений:

$$\begin{cases} \dfrac{\partial L}{\partial k} = 0, \\ \dfrac{\partial L}{\partial b} = 0; \end{cases} \Rightarrow \begin{cases} -\dfrac{2}{n}\displaystyle\sum_{i=1}^{n}(y_i - kx_i - b)x_i = 0, \\ -\dfrac{2}{n}\displaystyle\sum_{i=1}^{n}(y_i - kx_i - b) = 0; \end{cases}$$

По свойству линейности суммы, имеем:

$$\begin{cases} \dfrac{k}{n}\displaystyle\sum_{i=1}^{n}x_i^2 + \dfrac{b}{n}\sum_{i=1}^{n}x_i = \dfrac{1}{n}\sum_{i=1}^{n}x_i y_i, \\ \dfrac{k}{n}\displaystyle\sum_{i=1}^{n}x_i + b = \dfrac{1}{n}\sum_{i=1}^{n}y_i; \end{cases}$$

Вводя обозначение среднего арифметического $\overline{\alpha} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\alpha_i$, получим:

$$\begin{cases} k \cdot \overline{x^2} + b \cdot \overline{x} = \overline{xy}, \\ k \cdot \overline{x} + b = \overline{y}; \end{cases}$$

Выражая из второго уравнения b и подставляя его в первое, находим k:

$$k = \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2}.$$

Заметим, что если $E(x) = \overline{x}$, то можно записать: $k = \dfrac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2} = \dfrac{\mathrm{cov}(x,y)}{D(x)}$.

Зная k, находим b:

$$b = \overline{y} - k \cdot \overline{x} = \overline{y} - \overline{x} \cdot \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2}.$$

Искомое уравнение парной регресси имеет вид:

$$\hat{y}(x) = \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2} \cdot x + \overline{y} - \overline{x} \cdot \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2}.$$

3.9 Задача об оптимальном линейном прогнозе

Обобщим рассмотренную в предыдущем пункте задачу. Пусть в результате некоторого опыта мы, имея возможность наблюдать за случайной величиной ξ, хотим предсказывать другую случайную величину η, тоже связанную с этим эксперементом, то есть хотим построить оценку $\hat{\eta} = f(\xi)$. Чтобы найти лучшую оценку $\hat{\eta}$, логично в качестве функции ошибки выбрать $E((\hat{\eta} - \eta)^2)$, и минимизировать ее.

В общем виде такая задача достаточно трудна, но если рассматривать в качестве f только линейные функции, то этот аналог вариационной задачи можно решить аналитически. Действительно, пусть $\hat{\eta} = f(\xi) = a\xi + b$, тогда функция ошибок будет иметь вид:

$$F(a,b) = E((a\xi + b - \eta)^2) = a^2 E(\xi^2) + b^2 + E(\eta^2) + 2abE(\xi) - 2aE(\xi\eta) - 2bE(\eta).$$

Используя необходимое условие экстремума, запишем:

$$\begin{cases} \dfrac{\partial F}{\partial a} = 2aE(\xi^2) + 2bE(\xi) - 2E(\xi\eta) = 0, \\ \dfrac{\partial F}{\partial b} = 2b + 2aE(\xi) - 2E(\eta) = 0 \end{cases}$$

Решая систему, находим неизвестные параметры a и b:

$$a = \frac{\mathrm{cov}(\xi,\eta)}{D(\xi)}, \quad b = E(\eta) - aE(\xi).$$

Это решение обеспечивает минимум функционала $F(a,b)$ (это можно проверить, используя достаточное условие экстремума для функции нескольких переменных).

Подставляя найденные коэффициенты в равенство $\hat{\eta} = a\xi + b$ и преобразуя, получим, что оптимальная оценка в классе линейных имеет вид:

$$\hat{\eta} = E(\eta) + \sigma(\eta) r(\xi,\eta) \frac{\xi - E(\xi)}{\sigma(\xi)}.$$

3.10 Множественная линейная регрессия

На практике редко используется парная регрессия – обычно один из параметров, который мы хотим предсказать (он называется целевой переменной) зависит не от одного фактора, а от многих. Обобщим решение задачи регрессии на этот случай.

Пусть у нас есть целевая переменная y, которую мы хотим научиться предсказывать, и k параметров $(x_1, x_2, \ldots x_k)$, от которых y зависит линейно: $y_i = \omega_0 + \omega_1 x_{i1} + \omega_2 x_{i2} + \ldots + \omega_k x_{ik} + \varepsilon_i$. Пусть так же нам удалось собрать n штук наблюдений – как ведет себя y в

зависимости от параметров. Аналогично задаче парной регресси мы хотим предсказать $\hat{y}$ так, чтобы сумма квадратов расстояний от y до $\hat{y}$ была минимальной:

$$L(\omega_0, \omega_1, \dots \omega_k) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \omega_0 - \omega_1 x_{i1} - \omega_2 x_{i2} - \dots - \omega_k x_{ik})^2 \to \min.$$

Введем обозначения:

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{pmatrix}, \; y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \; \omega = \begin{pmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_k \end{pmatrix}.$$

Тогда задача минимизации функции L превращается в задачу нахождения минимума квадрата длины вектора $y - X\omega$:

$$L = \frac{1}{n}|y - X\omega|^2 = \frac{1}{n}(y - X\omega)^T(y - X\omega) \to \min.$$

Посчитаем градиент функции L:

$$\operatorname{grad} L = \begin{pmatrix} -\frac{2}{n}\sum_{i=1}^{n}(y_i - \omega_0 - \omega_1 x_{i1} - \dots - \omega_k x_{ik}) \\ -\frac{2}{n}\sum_{i=1}^{n} x_{i1}(y_i - \omega_0 - \omega_1 x_{i1} - \dots - \omega_k x_{ik}) \\ \vdots \\ -\frac{2}{n}\sum_{i=1}^{n} x_{ik}(y_i - \omega_0 - \omega_1 x_{i1} - \dots - \omega_k x_{ik}) \end{pmatrix} = -\frac{2}{n}X^T(y - X\omega).$$

Приравнивая градиент к нулю, получим аналитическое решение:

$$-\frac{2}{n}X^T(y - X\omega) = 0,$$

$$X^T X\omega = X^T y,$$

$$\omega = (X^T X)^{-1} X^T y.$$

3.11 Принцип максимального правдоподобия

Чтобы проиллюстрировать метод максимального правдоподобия, рассмторим следущий пример.

Предположим, что мы решили выяснить, сколько людей помнят со школы формулу дискриминанта квадратного уравнения, и решили провести эксперимент. Мы опросили 50 человек и оказалось, что формулу помнят всего 13 человек. Разумно предположить, что вероятность того, что следующий опрошенный знает формулу будет равна $\frac{13}{50} = 0{,}26$. Оказывается, что такая интуитивно понятная оценка не просто хороша, но еще и является оценкой максимального правдоподобия.

Действительно, подобный опрос с бинарным исходом (помнит – не помнит) представляет собой реализацию испытания по схеме Бернулли с неизвестным параметром p – вероятностью того, что респондент помнит формулу. Случайная величина ξ для каждого опрошенного может принимать два значения: 1 – если человек помнит формулу, и 0 – в

противном случае. Эта случайная величина будет иметь следующую функцию распределения (здесь x – реализация ξ в отдельном опыте):

$$F_\xi(p, x) = p^x(p-1)^{1-x}, \;\; x \in \{0, 1\}.$$

После того как мы опросили 50 человек, у нас есть вектор $\vec{x}$ размерности 50 – вектор реализации нашей случайной величины в данном эксперименте. Очевидно, что опросы всех 50 человек независимы, а значит, вероятность того, что в ходе эксперимента случайная величина ξ была реализована именно как вектор $\vec{x}$, есть просто произведение вероятностй всех реализаций ξ (для оценки неизвестного параметра p не важен порядок элементов в векторе $\vec{x}$, а важно лишь количество 0 и 1):

$$P(\vec{x} \mid p) = \prod_{i=1}^{50} p^{x_i}(1-p)^{1-x_i}, \;\; x_i \in \{0, 1\}.$$

С помощью составленного выражения можно найти интрересующий нас параметр p. Мы точно знаем, что результат опроса – вектор $\vec{x}$, с другой стороны мы показали, что вероятность именно такой реализации случайной величины ξ зависит от неизвестного параметра p по выписанной выше формуле. Идея метода максимального правдоподобия заключается в том, чтобы найти такой параметр p, при котором вероятность реализации случайной величины именно как $\vec{x}$ максимальна.

Чтобы было легче максимизировать функцию $P(\vec{x} \mid p)$ по p прологарифмируем это выражение (применение монотонного преобразования не изменит решение, но упростит вычисления):

$$\log P(\vec{x} \mid p) = 13 \log p + 37 \log(1-p).$$

Далее найдем экстремум данной функции:

$$\frac{\partial}{\partial p}\left(\log P(\vec{x} \mid p)\right) = \frac{\partial}{\partial p}\left(13 \log p + 37 \log\left(1-p\right)\right) = \frac{13}{p} - \frac{37}{1-p} = 0.$$

$$p = \frac{13}{50}.$$

3.12 Принцип максимального правдоподобия и линейная регрессия

Давайте рассмотрим линейную регрессию с вероятностной точки зрения, и попробуем приминить рассуждения, описанные в предыдущем пункте. Модель линейной регрессии остается такой же:

$$\vec{y} = X\vec{\omega} + \vec{\varepsilon},$$

но будем считать, что случайные ошибки распределены нормально:

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

Тогда можно записать:

$$f(y_i \mid X, \vec{w}) = \sum_{j=1}^{m} w_j X_{ij} + \mathcal{N}(0, \sigma^2) = \mathcal{N}\left(\sum_{j=1}^{m} w_j X_{ij}, \sigma^2\right).$$

Так как примеры независимы (ошибки не скоррелированы – одно из условий теоремы Гаусса-Маркова), то полное правдоподобие данных – это произведение функций плотности $f(y_i \mid X, \vec{w})$. Рассмотрим логарифм правдоподобия, что позволит нам перейти от произведения к сумме:

$$\log P\left(\vec{y} \mid X, \vec{w}\right) = \log \prod_{i=1}^{n} \mathcal{N}\left(\sum_{j=1}^{m} w_j X_{ij}, \sigma^2\right) = \sum_{i=1}^{n} \log \mathcal{N}\left(\sum_{j=1}^{m} w_j X_{ij}, \sigma^2\right) =$$
$$-\frac{n}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - \vec{w} \cdot \vec{x}_i\right)^2 .$$

Мы хотим найти гипотезу максимального правдоподобия, то есть нам нужно максимизировать выражение $P(\vec{y} \mid X, \vec{w})$, а это то же самое, что максимизация его логарифма. Таким образом:

$$\hat{w} = \arg\max_w P(\vec{y} \mid X, \vec{w}) = \arg\max_w \left(-\frac{n}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - \vec{w} \cdot \vec{x}_i\right)^2\right) =$$
$$\arg\max_w \left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - \vec{w} \cdot \vec{x}_i\right)^2\right) = \arg\max_w -\mathcal{L}(X, \vec{y}, \vec{w}) = \arg\min_w \mathcal{L}(X, \vec{y}, \vec{w}).$$

Таким образом, максимизация правдоподобия данных – это то же самое, что и минимизация среднеквадратичной ошибки (при справедливости указанных выше предположений). Получается, что квадратичная функция ошибки является следствием того, что ошибки распределены нормально.

3.13 Задача классификации

Ранее было рассмотрено решение задачи регрессии с помощью линейной модели. Оказывается, что с помощью аналогичного подхода можно решать задачу классификации.

Рассмотрим задачу бинарной классификации: есть два класса (которые мы обозначим как $+1$ и -1), причем гарантируется, что существует гиперплоскость, разбивающая пространство признаков на два полупространства так, чтобы все объекты одного класса лежали в одном полупространстве, а объекты другого класса – в другом (рисунок 2). В таких случаях говорят, что выборка линейно разделима.

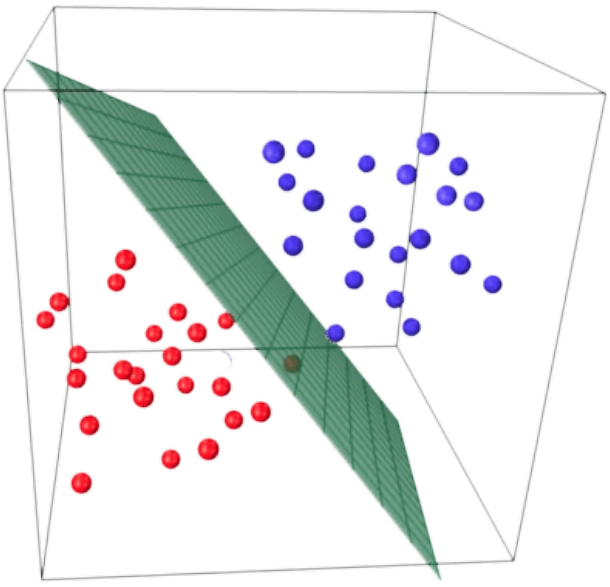

Рис. 2: Иллюстрация к задаче бинарной классификации.

Мы хотим научиться отделять объекты одного класса от объектов другого класса на основе обучающей выборки – пар $(y_i, \vec{x}_i)$, где y_i – метка классса ($+1$ или -1), а $\vec{x}_i$ – вектор

признаков данного объекта. Один из самых простых линейных классификаторов получается на основе регрессии следующим образом:

$$a(\vec{x}) = \text{sign}(\vec{w} \cdot \vec{x}),$$

где

- $\vec{x} \in \mathbb{R}^n$ – вектор признаков объекта (вместе с единицей)
- $\vec{\omega}$ – веса в линейной модели (вместе со смещением)
- $a : \mathbb{R}^n \to \{+1, -1\}$ – искомая зависимость метки класса от вектора признаков

Кроме линейной регресси существует популярная модель логистической регрессии, которая тоже является частным случаем линейного классификатора, но обладает хорошим свойством прогнозировать вероятность P_+ отнесения примера $\vec{x}$ к классу $+1$:

$$P_+ = P\left(y = 1 \mid \vec{x}, \vec{w}\right).$$

Определение 3.13.1. Шансом события A называется отношение вероятности того, что событие произойдет, к вероятности того, что событие A не произойдет:

$$\text{OR}(A) = \frac{P(A)}{P(\overline{A})} = \frac{P(A)}{1 - P(A)} = \frac{P_+}{1 - P_+}.$$

Заметим, что $\text{OR} \in (0, \infty)$, а значит $\log \text{OR} \in \mathbb{R}$. Именно поэтому в модели логистической регрессии с помощью МНК прогнозируется именно логарифм шанса:

$$\log \text{OR} = \vec{w} \cdot \vec{x}.$$

Таким образом, вероятности принадлежности бъекта к классу $+1$ или -1 вычисляется следующим образом:

$$P_+(\vec{x}) = \frac{OR}{1 + OR} = \frac{e^{\vec{w} \cdot \vec{x}}}{1 + e^{\vec{w} \cdot \vec{x}}} = \frac{1}{1 + e^{-\vec{w} \cdot \vec{x}}} = \sigma(\vec{w} \cdot \vec{x}).$$

$$P_-(\vec{x}) = P\left(y = -1 \mid \vec{x}, \vec{w}\right) = 1 - \sigma(\vec{w} \cdot \vec{x}) = \sigma(-\vec{w} \cdot \vec{x}).$$

Оба этих выражения можно объединить в одно:

$$P\left(y = y_i \mid \vec{x_i}, \vec{w}\right) = \sigma(y_i \vec{w} \cdot \vec{x_i}).$$

Теперь рассмотрим правдоподобие выборки, а именно, вероятность наблюдать данный вектор $\vec{y}$ у выборки X (выборка размера m). В предположении, что объекты выборки независимы и из одного распределения, получаем:

$$P\left(\vec{y} \mid X, \vec{w}\right) = \prod_{i=1}^{m} P\left(y = y_i \mid \vec{x_i}, \vec{w}\right).$$

Так как сумму оптимизировать намного проще, чем произведение, рассмотрим логарифм правдоподобия:

$$\log P\left(\vec{y} \mid X, \vec{w}\right) = \log \prod_{i=1}^{m} \sigma(y_i \vec{w} \cdot \vec{x_i}) = \sum_{i=1}^{m} \log \frac{1}{1 + e^{-y_i \vec{w} \cdot \vec{x_i}}} = -\sum_{i=1}^{m} \log(1 + e^{-y_i \vec{w} \cdot \vec{x_i}}).$$

Значит, в даном случае принцип максимального правдоподобия приводит к минимизации так называемой логистической функции потерь по всем объектам выборки:

$$\mathcal{L}(X, \vec{y}, \vec{w}) = \sum_{i=1}^{m} \log(1 + e^{-y_i \vec{w} \cdot \vec{x_i}}) \to \min_w.$$

3.14 Логистическая регрессия и logloss

Рассмотрим с вероятностной точки зрения схему испытаний Бернулли по подбрасыванию монеты, которая выпадает с вероятностью p решкой и $q = 1 - p$ орлом. Пусть мы провели серию опытов из n бросков, в котором монетка k раз выпала решкой.

Достаточно очевидно (это можно вывести из комбинаторных рассуждений или просто воспользоваться тем фактом, что такая случайная величина как количество выпадений решки имеет распределение Бернулли), что вероятность того, что при n подбрасываниях такая монета выпадет ровно k раз решкой это

$$P(k) = C_n^k p^k (1-p)^{n-k}.$$

Чтобы найти неизвестный параметр p можно воспользоваться принципом максимально провдоподобия. Для это нам надо подобрать такое p, чтобы вероятность наблюдения именно такого исхода опыта, который мы получили, был максимален, то есть найти

$$\tilde{p} = \arg\max P(k) = \arg\max \log P(k).$$

$$\tilde{p} = \arg\max(\log C_n^k + k\log p + (n-k)\log(1-p)) = \arg\max(k\log p + (n-k)\log(1-p)).$$

$$\tilde{p} = \frac{1}{n}\arg\max\left(\frac{k}{n}\log p + \left(1 - \frac{k}{n}\right)\log(1-p)\right).$$

Оказывается, что функция

$$L = -t\log p - (1-t)\log(1-p)$$

очень хорошо подходит в качестве функции потерь для решения задач бинарной классификации. Здесь $p \in (0,1)$ – предсказанная вероятность принадлежности объекта к одному из двух классов, а $t \in \{0,1\}$ – «таргетное» (целевое) значение вероятности. Эту функцию называют бинарной кросс-энтропией или перекресной энтропией.

На самом деле это та же логистическая функция потерь (часто так же ее называют logloss), которую мы рассмотрели в предыдущем пункте: заменим метки классов с t на y по правилу $0 \to -1$ и $1 \to 1$. Ошибка на одном объекте может быть записана следующим образом:

$$L = -\log p^t (1-p)^{1-t}.$$

Покажем, что

$$-\log p^t(1-p)^{1-t} = \log(1 + e^{-y\vec{w}\cdot\vec{x}}) = -\log\sigma(y\vec{w}\cdot\vec{x}).$$

Учитывая, что $p = \sigma(\vec{w}\cdot\vec{x})$, достаточно показать, что

$$\sigma^t(\vec{w}\cdot\vec{x})(1-\sigma(\vec{w}\cdot\vec{x}))^{1-t} = \sigma(y\vec{w}\cdot\vec{x}),$$

что легко проверяется подстановкой $t = 0, y = -1$ и $t = 1, y = 1$.

Почему же не использовать в качестве функции потерь для логистической регрессии обычный MSE $S = (p-t)^2$? Оказывается, что на практике при обучении нейронных сетей возникает проблема, которая назыается затухание градиента. Чтобы итеративно настраивать веса, нам необходимо уменьшать значение функции потерь: $S \to \min_w$. Это делается с помощью алгоритма градиентного спуска, а именно веса w изменяются по правилу:

$$\vec{w}_{i+1} = \vec{w}_i - \alpha\frac{\partial S}{\partial w}.$$

Так как занчение производной логистической функции $p'(z) = \sigma'(z) = \sigma(z)(1-\sigma(z))$ не может привышать 0,25, а при очень больших по модулю значениях аргумнента будет стремиться к нулю, то очевидно, что при попадании в эту область значения весов практически перестанут меняться:

$$\frac{\partial S}{\partial w} = \frac{\partial S}{\partial p}\frac{\partial p}{\partial \sigma}\frac{\partial \sigma}{\partial w} = 2(p-t)p(1-p)x.$$

В случае использования логистической функции потерь такой ситуации не происходит, и значение градиента прямо пропорционально разности между предсказанным и целевым значением вероятности:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial p}\frac{\partial p}{\partial \sigma}\frac{\partial \sigma}{\partial w} = \left(-\frac{t}{p} + \frac{1-t}{1-p}\right)p(1-p)x = (p-t)x.$$

3.15 Задача многоклассовой классификации и кросс-энтропия

Для задачи многоклассовой классификации используют обобщение логистической функции, а именно функцию Softmax, выходной вектор которой интерпретируется как вероятность принадлежности объекта к соответствующему классу:

$$Softmax(y_i) = Sm(y_i) = Sm_i = \frac{e^{y_i}}{\sum\limits_{j=1}^{n} e^{y_j}} = p_i.$$

Так же, для настройки весов, понадобятся значения производной:

$$\frac{\partial Sm_i}{\partial y_j} = \begin{cases} -Sm_i Sm_j = -p_i p_j, & i \neq j \\ Sm_i(1-Sm_i) = p_i(1-p_i), & i = j \end{cases}$$

В качестве функции потерь для обобщенной логистической функции используется обобщение logloss, а именно кросс-энтропия:

$$L = -\sum_{i=1}^{n} t_i \log p_i,$$

где p_i – предсказанная вероятность принадлежности объекта к соответствующему классу, а $t \in \{0,1\}$ – «таргетное» (целевое) значение вероятности. Для задачи мнгоклассовой классификации используется именно такая функция потерь по тем же причинам, почему для бинарной классификации используется logloss. Чтобы убедиться в том, что Softmax действительно подходящая функция потерь, найдем ее производную:

$$\frac{\partial L}{\partial y_j} = \frac{\partial}{\partial y_j}\left(-\sum_{i=1}^{n} t_i \log p_i\right) = \frac{\partial}{\partial y_j}\left(t_j \log p_j\right) - \frac{\partial}{\partial y_j}\left(\sum_{i \neq j} t_i \log p_i\right) =$$

$$= -\frac{t_j}{p_j}\frac{\partial p_j}{\partial y_j} - \sum_{i \neq j}\frac{t_i}{p_i}\frac{\partial p_i}{\partial y_j} = -\frac{t_j}{p_j}p_j(1-p_j) - \sum_{i \neq j}\frac{t_i}{p_i}(-p_i p_j) = -t_j + t_j p_j + p_j \sum_{i \neq j} t_i = -t_j + p_j \sum_{i=1}^{n} t_i = p_j - t_j.$$

4 Нейронные сети

4.1 Модель искусственного нейрона

Модель искусственного нейрона схематично может быть прдедставлена следующим образом:

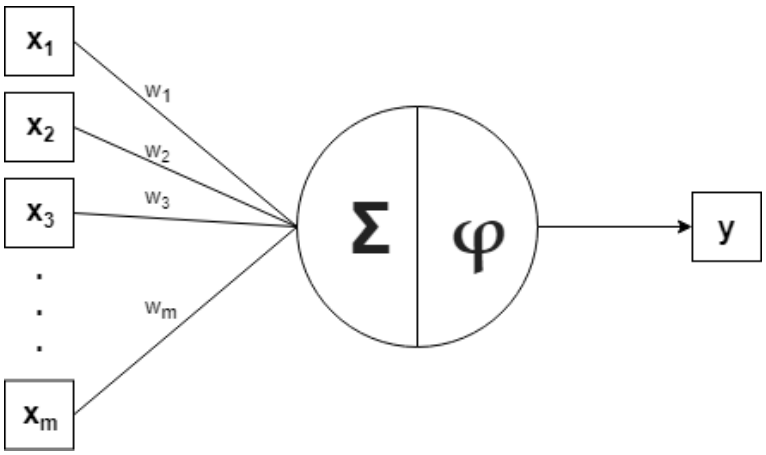

Рис. 3: Модель искусственного нейрона.

Здесь x_i – входные значения, подающиеся на вход нейрону, $y = y(x)$ – выход нейрона. Сам же нейрон представляет из себя композицию сумматорной и активационной функций. Сумматорная функция вычисляет взвешанную сумму входов, используя параметры нейрона (их называют весами), а активационная функция, применяя к выходу сумматорной функции некоторую функцию активации, возвращает итоговый выход нейрона. Таким образом, выход нейронна можно записать в виде:

$$y(x) = \varphi(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_m x_m) = \varphi\left(\omega_0 + \sum_{k=1}^{m} \omega_k x_k\right).$$

Параметр ω_0 называют смещением (bias) нейрона.

Ценность такой модели состоит в том, что она способна обучаться посредством изменения весов. Обучение нейрона возможно, когда сущетсвует обучающий набор – множество пар $(\vec{x}_i, \hat{y}_i)$ входных параметров и правильных ответов. Алгоритм обучения нейрона, известный как дельта-правило, достаточно прост, но мы не будем его обсуждать ввиду того, что на данный момент он представляет скорее исторический, чем практический интерес.

В качестве функции активации принято использовать нелинейные дифференцируемые функции. Нелинейность используется потому, что композиция линейных функций – линейная функция, и, следовательно, совокупность нейронов с линейной функцией активации не будет принципиально отличаться от одного нейрона с такой же функцией.

Пример 4.1.1. Приведем простой пример, когда комозиция нейронов с нелинейной пороговой функцией активации Хевисайда

$$f(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

может решать задачу, которую невозможно решить используя один нейрон с такой функцией активации.

Реализуем булевы ($x_1, x_2 \in \{0, 1\}$) функции как нейроны:

$$NO(x_1) = f\left(0{,}5 - x_1\right),$$

$$AND(x_1, x_2) = f\left(-1{,}5 + x_1 + x_2\right),$$
$$OR(x_1, x_2) = f(-1 + 2x_1 + 2x_2).$$

Нетрудно убедиться, что с помощью одного такого нейрона никак нельзя реализовать операцию исключающего или XOR. При этом, достаточно легко проверить, что с помощью конъюкции, дизъюнкции и отрицания эта функция может быть реализована следующим образом: [1]

$$XOR(x_1{,}x_2) = OR[AND(NO(x_1), x_2), AND(x_1, NO(x_2))].$$

Такой способ называется суперпозицией (в данном случае двухслойной) функций AND, OR, NO.

Другой подход в машинном обучении, а именно конструирование новых признаков, так же помогает решить эту задачу:

$$XOR(x_1, x_2) = f(x_1 + x_2 - 2x_1x_2 - 0.5).$$

Оказывается, что в задаче бинарной классификации, всегда, когда в пространстве признаков существует гиперплоскость, отделяющая объекты одного класса от другого, совокупность рассмотренных нами нейронов с пороговой функцией активации, способна решить задачу классификации за конечное число шагов. Об этом говорит известная теорема о сходимости обучения перцептрона при линейно разделимой обучающей выборке.

4.2 Пример нейронной сети для решения задачи регрессии

Концепция нейронных сетей подразумевает объеденение нейронов, рассмотренных в предыдущем пункте, определенным образом. Мы рассмотрим простейшую полносвязную нейронную сеть, которая после обучения будет способна аппроксимировать любую вещественнозначную функцию одного вещественного аргумента.

Архитектура нашей сети будет следующей (рисунок 4). Один нейрон на входном слое, n нейронов на скрытом слое, и один нейрон на выходном слое, что разумно, так как сеть на вход принимает одно число, и ответом сети тоже является одно число. При этом в качестве активационной функции для всех нейроннов скрытого слоя мы будем использовать сигмоиду. Тогда, по теореме Цыбенко, возможна аппроксимация любой гладкой функции с любой точностью. У выходного нейрона не будет функции активации (то есть функция активации последнего нейрона есть простая линейная функция), так как мы решаем задачу регрессии, а значит, нам необходимо уметь предсказывать ответ в диопазоне $(-\infty, \infty)$.

Нетрудно убедиться, что ответом нейронной сети на вход (число) x будет число: [2]

$$y(x) = B\sigma(Ax + \alpha) + \beta,$$

где $A = \begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix}^T$ – вектор весов нейронов скрытого слоя, $\alpha = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n \end{pmatrix}^T$ – вектор смещений нейронов скрытого слоя, $B = \begin{pmatrix} b_1 & b_2 & \dots & b_n \end{pmatrix}$ – вектор весов нейрона выходного слоя, а β – смещение нейрона выходного слоя.

Теперь нам надо обучить сеть, то есть подобрать $3n + 1$ параметров (именно столько параметров суммарно содержат матрицы A, α, B и β). Делать мы это будем с помощью

[1] Так же можно реализовать исключающее или как $XOR(x_1{,}x_2) = f(-0{,}5 + OR(x_1{,}x_2) - AND(x_1, x_2))$.

[2] Здесь и далее применение к матрице некоторой функции означает применение этой функции ко всем элементам матрицы.

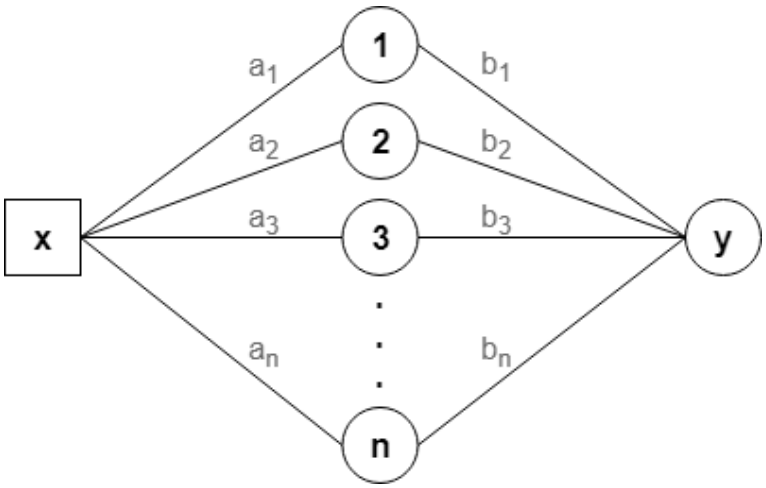

Рис. 4: Простейшая нейронная сеть с одним скрытым слоем.

стахостического градиентного спуска по батчам, а значит, нам нужна функция потерь, которую мы будем оптимизировать. Самая популярная функция потерь для задач регрессии – среднеквадратичная ошибка (MSE):

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(B\sigma(Ax_i + \alpha) + \beta - \hat{y}_i)^2.$$

Здесь $y_i = y(x_i)$ – ответ сети на вход x_i, а $\hat{y}_i$ – правильный ответ для задачи регресси на вход x_i (напомним, что мы рассматриваем обучение с учителем, то есть у нас есть так называемое обучающее множество – пары $(x_i, \hat{y}_i)$, состоящие из предиктора и значения $\hat{y}(x_i)$, где $\hat{y}$ – искомая зависимость). N – это количество обучающих примеров в батче.[3]

Напомним, что производная функции $f : \mathbb{R}^k \to \mathbb{R}$ по матрице $A = (a_{ij}) \in \underset{m\times n}{\mathbf{M}}$ – это матрица, состоящая из частных производных функции f по элементам матрицы A:

$$\frac{\partial f}{\partial A} = \begin{pmatrix} \frac{\partial f}{\partial a_{11}} & \frac{\partial f}{\partial a_{12}} & \cdots & \frac{\partial f}{\partial a_{1n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial a_{m1}} & \frac{\partial f}{\partial a_{m2}} & \cdots & \frac{\partial f}{\partial a_{mn}} \end{pmatrix}.$$

Чтобы, используя градиентый спуск, итеративно обновлять веса нейронной сети, нам необходимо найти выражения для частных производных функции потерь по всем весам. Это легко сделать, используя правило цепи (правило дифференцирования сложной функции):

$$\frac{\partial L}{\partial A} = \frac{2}{N}\sum_{i=1}^{N}(B\sigma(Ax_i + \alpha) + \beta - \hat{y}_i) \cdot \sigma(Ax_i + \alpha) \odot (1 - \sigma(Ax_i + \alpha))x_i,$$

$$\frac{\partial L}{\partial \alpha} = \frac{2}{N}\sum_{i=1}^{N}(B\sigma(Ax_i + \alpha) + \beta - \hat{y}_i) \cdot \sigma(Ax_i + \alpha) \odot (1 - \sigma(Ax_i + \alpha)),$$

$$\frac{\partial L}{\partial B} = \frac{2}{N}\sum_{i=1}^{N}(B\sigma(Ax_i + \alpha) + \beta - \hat{y}_i) \cdot \sigma(Ax_i + \alpha)^T,$$

$$\frac{\partial L}{\partial \beta} = \frac{2}{N}\sum_{i=1}^{N}(B\sigma(Ax_i + \alpha) + \beta - \hat{y}_i).$$

[3]При $N = 1$ это будет обычный стахостический градиентый спуск.

Знак $\odot$ означает поэлементное матричное умножение. [4]

Таким образом, мы можем итеративно обучать нашу нейронную сеть, просматривая весь обучающий набор, желательно, несколько раз.

[4] В линейной алгебре эта операция называется произведением Адамара.

5 Список литературы

1. С. М. Никольский – Курс математического анализа.
2. Д. В. Беклемишев – Курс аналитической геометрии и линейной алгебры.
3. Д. В. Беклемишев – Дополнительные главы линейной алгебры.
4. А. И. Кострыкин – Введение в алгебру. Часть 1.
5. А. Н. Ширяев – Вероятность.
6. А. И. Кибзун, Е. Р. Горяинова, А. В. Наумов, А. Н. Сиротин – Теория вероятностей и математическая статистика.
7. Джоэл Грас – Data Science. Наука о данных с нуля.
8. Тарик Рашид – Создаем нейронную сеть.
9. Франсуа Шолле – Глубокое обучение на Python.
10. Орельен Жерон – Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow.
11. Michael Nielsen – Neural Networks and Deep Learning.
12. Бенджио Иошуа, Гудфеллоу Ян, Курвилль Аарон – Глубокое обучение.
13. С. Николенко, А. Кадурин, Е. Архангельская – Глубокое обучение. Погружение в мир нейронных сетей
14. Саймон Хайкин – Нейронные сети.