Векторы и матрицы

Как превратить признаки в координаты и применить одно правило сразу ко многим объектам?

Цветок описывают длиной и шириной лепестка. Программа должна сравнить его с другими цветами и получить один ответ. Векторы и матрицы помогут записать эти действия так, чтобы не потерять порядок признаков.

1. Несколько признаков одного объекта

Одно сообщение можно описать двумя числами: длиной текста и числом ссылок. Для другого объекта удобно хранить длину и ширину лепестка. **Вектор** собирает такие координаты в упорядоченный набор. Запись $x = (2, 3)$ сообщает две величины в заранее выбранном порядке.

Порядок обязателен. Если первая координата означает длину, а вторая ширину, векторы (2; 3) и (3; 2) описывают разные измерения. Перед расчётом подпишите смысл и единицы каждой позиции. Вектор сам по себе не хранит объяснение, что такое два и три. Его должен знать человек или программа, которая готовит признаки.

Векторы одинаковой длины можно складывать по координатам: (2; 3) плюс (1; −2) даёт (3; 1). Умножение на число меняет каждую координату: удвоенный (2; 3) равен (4; 6). Число в таком действии называют скаляром. Геометрически положительный множитель сохраняет направление стрелки и меняет длину.

Сравнивать векторы разных размеров по этой операции нельзя: для лишней координаты не найдётся пары. Если у одного объекта заполнены два признака, а у другого три, сначала нужно согласовать описание. Добавить пропущенный признак нулём без объяснения означало бы придумать измерение.

Пример

Для цветка координаты могут быть «длина лепестка, ширина лепестка» в сантиметрах. Для сообщения «число символов, число ссылок». Одинаковые числа в этих двух пространствах имеют разный смысл.

> У вектора (5; 3) первую координату увеличили на 2, а вторую уменьшили на 1. Какова сумма новых координат?

2. Длина и расстояние

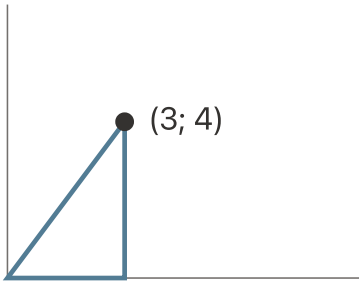
Вектор (3; 4) можно нарисовать стрелкой от начала координат к точке с горизонтальным смещением три и вертикальным четыре. Стрелка становится гипотенузой прямо-

угольного треугольника. По теореме Пифагора её длина равна квадратному корню из суммы квадратов координат: $\sqrt{3^2 + 4^2} = 5$.

Для расстояния между двумя точками сначала найдите разности соответствующих координат. Между (1; 2) и (4; 6) разность равна (3; 4), поэтому расстояние снова пять. Направление вычитания здесь не меняет расстояния: знаки разностей исчезают при возведении в квадрат.

В опыте одна координата равна трём, другая меняется. Предскажите длину при второй координате ноль и при второй координате четыре. Первая длина равна трём, вторая пяти. Когда вторая координата растёт, стрелка удлиняется, хотя её горизонтальная часть сохраняется.

Проверьте свой прогноз



Расстояние от начала координат: √(3² + 4²) = 5,00.

Горизонтальная часть остаётся равной 3.

Горизонтальная координата равна 3. Ползунок меняет вертикальную координату и длину стрелки.

Интерактивный эксперимент — в электронной версии урока.

Расстояние зависит от масштаба координат. Если только длину лепестка перевести из сантиметров в миллиметры, её численные различия вырастут в десять раз и получат больший вес. Геометрический объект не изменился, а правило численного сравнения изменилось. Поэтому перед поиском соседей проверяют единицы и способ масштабирования признаков.

> Точки имеют координаты (2; 1) и (5; 5). Чему равно евклидово расстояние между ними?

{ƒ} Обозначения

Символ	Смысл
x_i	координата под номером i
$\|\|x\|\|$	длина вектора
$x - y$	покоординатная разность

Эпик. **Контрпример: единицы измерения выбирают соседа**

Утверждение «расстояние зависит от масштаба» звучит безобидно, пока не меняет ответ. Пусть цветок-запрос имеет длину лепестка 2 см и ширину 10 мм, а два известных цветка — А с размерами (3 см; 10 мм) и Б с размерами (2 см; 12 мм). В этих единицах расстояние до А равно 1, до Б равно 2: ближайший сосед — А.

Переведём ширину в сантиметры: запрос (2; 1,0), А (3; 1,0), Б (2; 1,2). Расстояние до А по-прежнему 1, до Б стало 0,2: ближайший сосед — Б. Данные не изменились, изменилась только запись, а ответ перевернулся. Формула честно считает то, что ей дали; сломалось молчаливое предположение, что единица по каждой оси значит одно и то же.

Общий случай: если разность по ширине умножить на множитель s, расстояние до Б станет $2s$, а до А останется 1. Они сравняются при $s = 0{,}5$, то есть когда ширину измеряют в единицах по 2 мм. Любой множитель меньше половины делает соседом Б, любой больше — А. Выбор единиц — это выбор весов признаков, сделанный незаметно.

Правильный ход — не искать «истинные» единицы, а привести признаки к общему масштабу: вычесть среднее и разделить на разброс по каждой оси. Тогда единица по любой оси означает «одно типичное отклонение». Так делают перед поиском соседей, кластеризацией и обучением сетей; без этого признак с крупными числами диктует ответ.

> Запрос описан парой чисел (длина 2; ширина 10), объект А — (3; 10), объект Б — (2; 12). Все значения ширины умножают на множитель s, длину не трогают. При каком s евклидовы расстояния от запроса до А и до Б совпадут?

3. Взвесить координаты

Допустим, два признака влияют на итог по-разному. Первый умножается на четыре, второй на минус один. Для объекта (2; 3) получаем `2 * 4 + 3 * (-1) = 5`. Такой расчёт называется **скалярным произведением**: перемножить соответствующие координаты и сложить произведения.

Веса записывают отдельным вектором $w = (4, -1)$. Тогда краткая запись $w \cdot x$ означает тот же расчёт. Отрицательный вес делает вклад второго признака отрицательным, а не игнорирует его. Если увеличить второй признак на единицу при неизменном первом, сумма уменьшится на один.

Прибавим сдвиг $b = 2$. Линейное правило $w \cdot x + b$ даст семь. В машинном обучении веса определяют вклад признаков, а сдвиг позволяет менять результат даже при нулевых признаках. Все эти параметры пока можно выбрать руками и проверить их действие на одном объекте.

Скалярное произведение имеет и геометрический смысл. Перпендикулярные ненулевые векторы дают ноль; одинаково направленные дают положительный результат. Например, (1; 0) и (0; 1) имеют произведение ноль. Но величина произведения зависит также от длин векторов. Для сравнения только направления используют нормировку, которую позже разберём в сюжете о похожести.

> Для x = (3; 2), w = (2; −1) и сдвига b = 1 найдите w·x + b.

Заметка

Сумма произведений координат возвращает одно число. Умножение каждого элемента одного массива на соответствующий элемент другого вернуло бы несколько чисел. Это разные операции.

Профи. **Скалярное произведение и угол**

Почему нуль означает прямой угол, а не просто «ничего общего»? Раскроем квадрат длины разности по координатам. Для $x = (x_1; x_2)$ и $y = (y_1; y_2)$:

$$\|x - y\|^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2 = \|x\|^2 + \|y\|^2 - 2\,x \cdot y.$$

Перекрёстные слагаемые $-2x_1y_1 - 2x_2y_2$ — это удвоенное скалярное произведение со знаком минус. С другой стороны, в треугольнике со сторонами $\|x\|$, $\|y\|$ и $\|x - y\|$ теорема косинусов даёт $\|x - y\|^2 = \|x\|^2 + \|y\|^2 - 2\|x\|\|y\|\cos\theta$. Сравнивая две записи, получаем

$$x \cdot y = \|x\|\,\|y\|\cos\theta.$$

Три следствия. При прямом угле косинус равен нулю, и произведение тоже. Для $x = (3; 4)$ и $y = (4; 3)$ произведение равно 24, длины по 5, значит $\cos\theta = 24/25 = 0{,}96$: стрелки почти совпадают. Наконец, само число $x \cdot y$ не показывает похожесть направлений: у пар $(100; 0) \cdot (1; 1) = 100$ и $(1; 0) \cdot (1; 1) = 1$ угол один и тот же. Типичная ошибка — сравнивать объекты по величине скалярного произведения при разных длинах векторов: длинный вектор выигрывает любое сравнение. Поэтому для похожести произведение делят на обе длины и получают косинус угла.

> Длина вектора x равна 5, длина вектора y равна 5, их скалярное произведение равно 1. Чему равен квадрат длины разности $x - y$?

4. Матрица хранит несколько объектов

Вектор описывает один объект. Несколько одинаково устроенных векторов можно положить строками в **матрицу**. В матрице `[[1, 2], [3, 4], [5, 6]]` три строки и два столбца: три объекта, по два признака у каждого. Размер записывают как «3 на 2».

Первый индекс элемента выбирает строку, второй столбец. В математических записях индексы часто начинаются с единицы, в Python с нуля. Поэтому элемент третьей строки второго столбца равен шести, но обращение к нему в массиве Python записывается как `[2, 1]`. До чтения индекса установите соглашение.

Умножение матрицы на вектор весов применяет одно правило ко всем строкам. Для весов (2; 1) получаем три суммы: `1*2 + 2*1 = 4`, `3*2 + 4*1 = 10`, `5*2 + 6*1 = 16`. Результат содержит одно число на объект. Никакого отдельного способа умножения здесь нет: повторяется знакомое скалярное произведение.

Транспонирование меняет строки и столбцы местами. После него матрица размера 3 на 2 получает размер 2 на 3. Значения сохраняются, но их положение меняется. Если исходные строки означали объекты, после транспонирования объекты будут лежать по столбцам. Смысл осей следует переподписать.

Ссылки

NumPy: основы массивов связывает форму массива, строки, столбцы и транспонирование с конкретным кодом.

5. Проверить размеры до вычисления

Матрицу размера $m \times n$ можно умножить на матрицу размера $n \times k$. Внутренние размеры совпадают: каждая строка первой имеет столько элементов, сколько столбец второй. Результат имеет размер $m \times k$. Чтобы найти его элемент, берём одну строку слева и один столбец справа, перемножаем пары и складываем.

Для матрицы 2 на 3 и матрицы 3 на 4 произведение имеет размер 2 на 4. Обратный порядок не подходит: у матриц 3 на 4 и 2 на 3 внутренние размеры четыре и два различаются. Даже когда оба порядка допустимы, результаты обычно разные. Матричное умножение нельзя переставлять как умножение обычных чисел.

В упражнении используются только списки Python. Допишите сумму двух произведений и получите ответ для каждого объекта. Сначала проверьте одну строку вручную. Когда результат содержит три числа, это дополнительно подтверждает форму: три исходных объекта получили по одному результату.

Применить веса к трём объектам

Цель: Для каждой строки вычислите сумму признаков с весами 2 и 1.

```
rows = [[1, 2], [3, 4], [5, 6]]
w = [2, 1]
result = []
for row in rows:
    score = row[0]  # TODO: учтите оба признака и веса
    result.append(score)
print(f"Результаты: {result}")
```

Профи. **Матрица на матрицу: порядок как у функций**

Элемент произведения AB в строке i и столбце j — скалярное произведение строки i матрицы A на столбец j матрицы B. Для A со строками (1; 2), (0; 3) и B со строками (3; 1), (7; 2) верхний левый элемент равен $1 \cdot 3 + 2 \cdot 7 = 17$, верхний правый $1 \cdot 1 + 2 \cdot 2 = 5$. Всего четыре такие суммы.

Почему порядок важен, видно из смысла. Применим B к вектору-столбцу x, а затем A к результату: получится $A(Bx)$, и это то же, что $(AB)x$. Произведение матриц — композиция двух правил, причём правое применяется первым, как в записи $f(g(x))$. Переставить A и B — значит поменять порядок действий, а «удвоить, затем прибавить» и «прибавить, затем удвоить» дают разные результаты.

Типичная ошибка — перемножить матрицы поэлементно, как две таблицы: $1 \cdot 3$, $2 \cdot 1$ и так далее. В NumPy так работает знак `*`, а матричное произведение записывается через `@`. Ошибку ловят размеры: поэлементное умножение требует одинаковых форм, а матричное — совпадения внутренних размеров. Для A размера 2 на 3 и B размера 3 на 2 поэлементное произведение не определено вовсе.

История

Матричная запись позволяет описывать системы линейных уравнений единообразно. MIT OpenCourseWare: линейная алгебра продолжает эту тему. В нашем опыте уравнений ещё нет: матрица лишь собирает несколько одинаковых расчётов.

Подумай

Проверьте веса (0; 1): результат должен совпасть со вторым столбцом. Это простой контроль порядка признаков и весов без длинной арифметики.

Перемножить две матрицы

Цель: Допишите сумму по внутреннему индексу и получите AB и BA.

```
A = [[1, 2], [0, 3]]
B = [[3, 1], [7, 2]]

def matmul(P, Q):
    rows, inner, cols = len(P), len(Q), len(Q[0])
    R = [[0] * cols for _ in range(rows)]
    for i in range(rows):
        for j in range(cols):
            R[i][j] = P[i][0] * Q[0][j]  # TODO: сложите произведения по
всем k
    return R

C = matmul(A, B)
D = matmul(B, A)
print(f"AB = {C}")
print(f"BA = {D}")
```

Эпик. **Порядок скобок и цена вычислений**

Умножение матрицы $m \times n$ на матрицу $n \times k$ требует $m \cdot n \cdot k$ умножений: по n на каждый из $m \cdot k$ элементов результата. Это число растёт быстрее, чем кажется. Пусть A и B имеют размер 1000 на 1000, а x — столбец из 1000 чисел. Произведения $(AB)x$ и $A(Bx)$ дают один и тот же вектор, но $(AB)x$ стоит $10^9 + 10^6$ умножений, а $A(Bx)$ — всего $2 \cdot 10^6$: примерно в пятьсот раз дешевле. Оценка порядка: при миллиарде умножений в секунду это секунда против двух миллисекунд.

Отсюда правило для нейросетей, где выход имеет вид $W_3W_2W_1x$: слои применяют к вектору по очереди, справа налево, и никогда не перемножают матрицы весов заранее. Ответ тот же, цена другая.

> Матрицы A и B имеют размер 470 на 470, x — столбец из 470 чисел. Во сколько раз больше умножений требует $(AB)x$, чем $A(Bx)$? Ответ с точностью до сотых.

Где это встретится дальше

Смысл координат и размеры определяют расчёт.
Взвешенная сумма одной строки превращается
в матричное правило для нескольких объектов.

Вычислить расстояние и скалярное произведение.

Различить объект, признак, строку и столбец.

Проверить допустимость матричного умножения по размерам.

Домашняя работа

1. Найдите евклидову длину вектора (5; 12).

2. Матрица имеет строки (2; 1) и (0; 3). Умножьте её на вектор-столбец (4; 2).
3. ★ Найдите ненулевой вектор, перпендикулярный (2; 3).

Источники

- MIT: линейная алгебра (https://ocw.mit.edu/courses/18-06-linear-algebra-spring-2010/).
- NumPy: основы массивов (https://numpy.org/doc/stable/user/absolute_beginners.html).
- NumPy: матричное произведение (https://numpy.org/doc/stable/reference/generated/numpy.matmul.html).

Повторить эту тему