Основы линейной алгебры. Классификация: начало

Маша Шеянова, masha.shejanova@gmail.com

Линейная алгебра на коленке

Линал рулит!

- Data Science — это немного матана, немного кода и много линейной алгебры
- Хотите понимать, что тут написано?))

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}}$$

- Welcome to linear algebra!

What is what

Вектор, матрица, размерность. R^n, C^n.

Матрица nxn:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

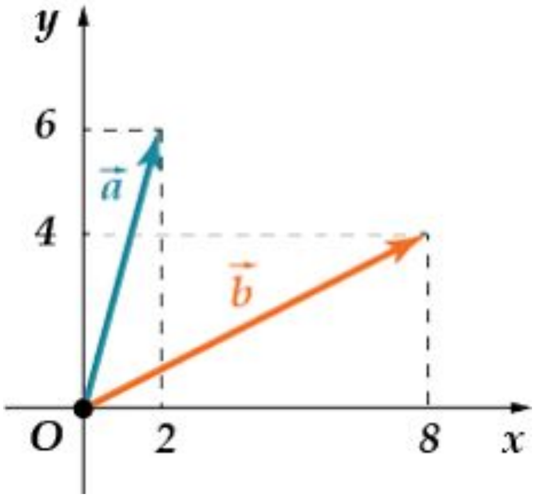
Вектор, размерность 4: (a, b, c, d)

Вектор

Как вы помните, на вектор можно смотреть “с разных углов”:

- набор чисел, например
- стрелочка на координатной плоскости
- точка на координатной плоскости

Всё это, по сути, разные представления одного и того же.



Что можно делать с векторами

- складывать / вычитать (поэлементно)

  (2, 3) + (-1, 10) = (1, 13)

- умножать на число (скаляр)

  4 x (2, 3) = (8, 12)

- считать длину (модуль) вектора:

$$\| X \| = \sqrt{\alpha_1{}^2 + \alpha_2{}^2 + \ldots + \alpha_n{}^2}$$

- считать скалярное произведение векторов

Скалярное произведение

Скалярным произведением векторов

$$X = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \dots \\ \alpha_n \end{pmatrix}, Y = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \dots \\ \beta_n \end{pmatrix}$$

называется число

$$(X, Y) = \alpha_1\beta_1 + \alpha_2\beta_2 + \ldots + \alpha_n\beta_n$$

Расстояние между векторами

- евклидово расстояние

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}.$$

- косинусная мера (косинус угла между векторами)

$$\cos(X \hat{\ } Y) = \frac{(X, Y)}{\| X \| \cdot \| Y \|} = \frac{\alpha_1 \beta_1 + \ldots + \alpha_n \beta_n}{\sqrt{\alpha_1^2 + \ldots + \alpha_n^2} \sqrt{\beta_1^2 + \ldots + \beta_n^2}}$$

Матрица

На матрицу тоже можно смотреть по-разному:

- таблица с числами
- набор векторов
- преобразование пространства

Чтобы понять, почему последнее представление тоже работает, нужно чуть глубже разобраться в линейной алгебре.

Что можно делать с матрицами

- складывать / вычитать
- умножать на число (скаляр)
- транспонировать
- перемножать
- умножать на вектор

Транспонирование

Транспонировать матрицу — значит “перевернуть” её. Взять каждый столбец и положить набок, сделать строкой. (источник картинки)

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}^T = \begin{pmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{pmatrix}$$

Умножение матриц

(Источник картинки)

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

$$AB = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{pmatrix}$$

При матричном умножении, каждая **строка первой матрицы** (слева) “умножается” на каждый **столбец второй матрицы** (справа).

“Умножается” — значит, берётся скалярное произведение векторов.

Умножение матриц: свойства

- ассоциативность

  (A * B) * C = A * (B * C)

- дистрибутивность

  A(B + C) = AB + AC

- не коммутативность

  A * B != B * A

Умножение на вектор

Это в принципе то же самое, что и умножение двух матриц. Только одна из матриц — размерности 1 x n. (источник картинки)

$$\begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + cy + e \\ bx + dy + f \\ 0 + 0 + 1 \end{bmatrix}$$

Какие матрицы бывают

Диагональная:

$$A_n = \begin{pmatrix} a & 0 & \cdots & 0 & 0 \\ 0 & a & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a & 0 \\ 0 & 0 & \cdots & 0 & a \end{pmatrix}$$

Единичная (обозначается E):

1 0 0
0 1 0
0 0 1

Верхне-треугольная:

$$U_5 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 3 & 6 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

Вырожденная:

внизу — строчки с нулями

Обратная матрица

Пускай A — некоторая матрица. Матрица B отбратна A если A * B = E.

Например, так:

$$\begin{bmatrix} 4 & 7 \\ 2 & 6 \end{bmatrix}\begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.4 \end{bmatrix} = \begin{bmatrix} 4\times0.6 + 7\times-0.2 & 4\times-0.7 + 7\times0.4 \\ 2\times0.6 + 6\times-0.2 & 2\times-0.7 + 6\times0.4 \end{bmatrix}$$

$$= \begin{bmatrix} 2.4 - 1.4 & -2.8 + 2.8 \\ 1.2 - 1.2 & -1.4 + 2.4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Где ботать линал

Здесь я рассказала **самое-самое начало**. Этого хватит, чтобы понимать формулы с предыдущего слайда, но в основы линала входят ещё:

- *понятие линейности*, *линейные преобразования*
- *определитель (determinant)* — важная характеристика матрицы
- *линейное пространство и его базис*
- *метод Гаусса* — как решать любые системы линейных уравнений
- и многое другое

Если хотите глубже понимать линейную алгебру, можно пройти курс на khan academy или курс на coursera от вышки. А ещё есть 3Blue1Brown.

Классификация

Популярные алгоритмы классификации

- k ближайших соседей (kNN)
- **наивный Байес**
- деревья решений
- логистическая регрессия
- метод опорных векторов (SVM)

Наивный Байес

Вспомним формулу Байеса

$$P(H_i / A) = \frac{P(H_i) \cdot P(A / H_i)}{P(A)}$$

Вот на этой идее и держится Байесовский классификатор:

насколько вероятно, что метка Hi возникла при условии объекта А.

Здесь не происходит никакого обучения, только подсчёт вероятностей!

Наивный Байес ≠ байесовские методы!

Наивный Байес — самый простой классификатор в машинном обучении.

Байесовские методы — нестандартный подход к нейросетям, по которому защищают диссертации и ведут целые курсы!

Метрики качества

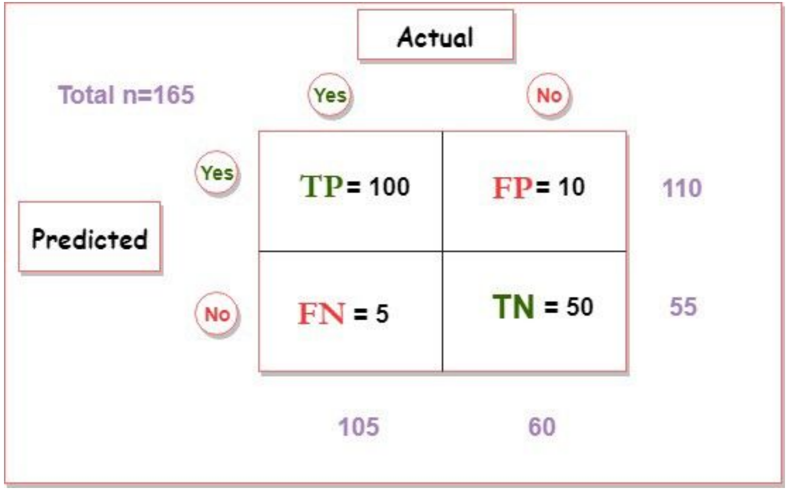
Any ideas?

Accuracy. Confusion matrix.

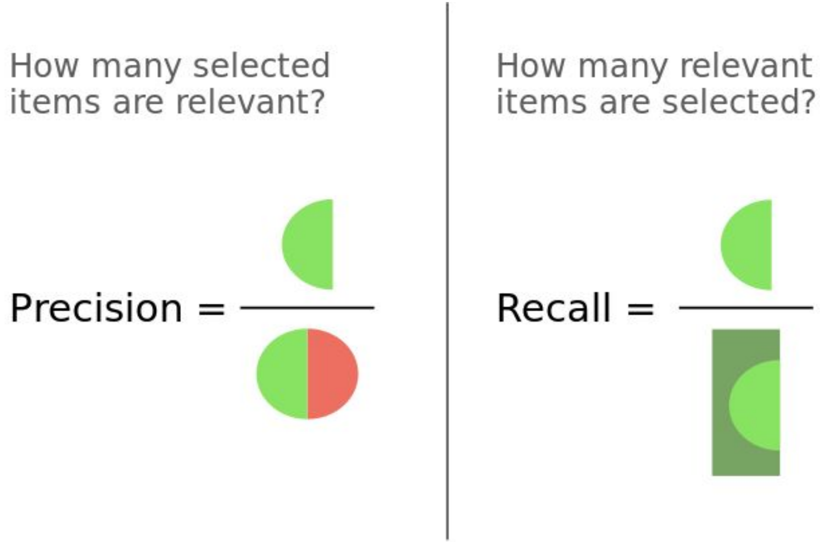
Метрика accuracy — самая простая оценка классификации: поделить все правильные ответы классификатора на количество всех ответов.

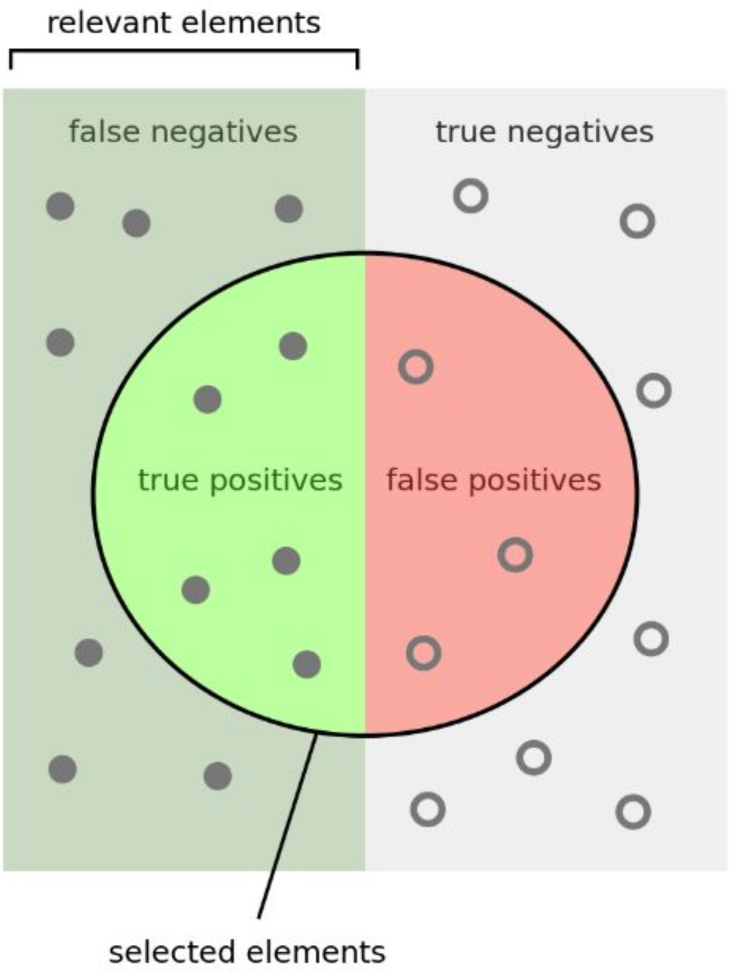
Достоинства: простота. Недостатки: Плохо работает при высокой априорной вероятности у одного из классов.

Confusion matrix ➔



Точность и полнота





Precision (точность): не отхватить лишнего

Recall (полнота): ничего не упустить

Точность и полнота

Когда важна точность (не отхватить случайно ничего лишнего):

- ?..

Когда важна полнота (ничего не забыть):

- диагностика болезней
- поиск террористов
- ???

f1-мера

Но что, если важно и то, и то?

Можно было бы просто посчитать среднее между точностью и полнотой. Но тогда очень плохие результаты будут get away.

f1 — среднее гармоническое точности и полноты

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Векторизаторы

One-hot encoding

Упорядочиваем значения категориальной переменной… и превращаем её в столько бинарных переменных, сколько у неё было значений.

значение
french
english
italian
russian

is_french	is_english	is_italian	is_russian
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Теперь можно подбирать коэффициенты!

Мешок слов

А теперь сделаем такой вектор для каждого слова в тексте… и сложим!

мягкий ➔ (1, 0)

котик ➔ (0, 1)

“Мягкий, мягкий котик” ➔ (2, 1)

Такой подход называется моделью “мешок слов”.

Почему? Нам не важен порядок!

Ресурсы

Посмотреть / почитать

Почитать

- про presicion, recall и f-меру (англ)
- то же самое по-русски (рус)

Посмотреть

- курс на khan academy про линал (англ)
- курс на coursera про линал (внезапно, рус)
- 3blue1Brown, Essence of linear algebra (англ)

Термины на английском языке

dot product — скалярное произведение

precision — точность

recall — полнота