

Дополнительная лекция



- Операции с матрицами
- Производная и градиент
- Градиентный спуск
- Матричные разложения
- Понижение размерности

Даниил Корбут

Специалист по Анализу Данных



Даниил Корбут
ML Engineer
Deliveroo

Окончил магистратуру ФИБТ
МФТИ в 2020.

Работал в Яндекс.Алиса, Insilico
Medicine, Microsoft, PicsArt.

Сейчас работаю на позиции ML
Engineer в Deliveroo.

Скалярное произведение векторов

Скалярное произведение векторов (dot product по англ.) - это скаляр (число), полученное в результате перемножения длин векторов на косинус угла между ними.

$$\vec{a} \cdot \vec{b} = |\vec{a}| \cdot |\vec{b}| \cdot \cos(\alpha)$$

Если известны координаты векторов, то скалярное произведение можно посчитать по формуле:


$$\vec{a} \cdot \vec{b} = x_a \cdot x_b + y_a \cdot y_b$$

где $\vec{a}(x_a; y_a)$ и $\vec{b}(x_b; y_b)$ вектора в двумерном пространстве

Транспонирование матрицы

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

Транспонирование матрицы
— это замена строк на
столбцы.


$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} A_{1,1} & A_{2,1} & A_{3,1} \\ A_{1,2} & A_{2,2} & A_{3,2} \end{bmatrix}$$

Транспонирование матрицы можно рассматривать как отображение матрицы относительно главной диагонали.

Обратная матрица

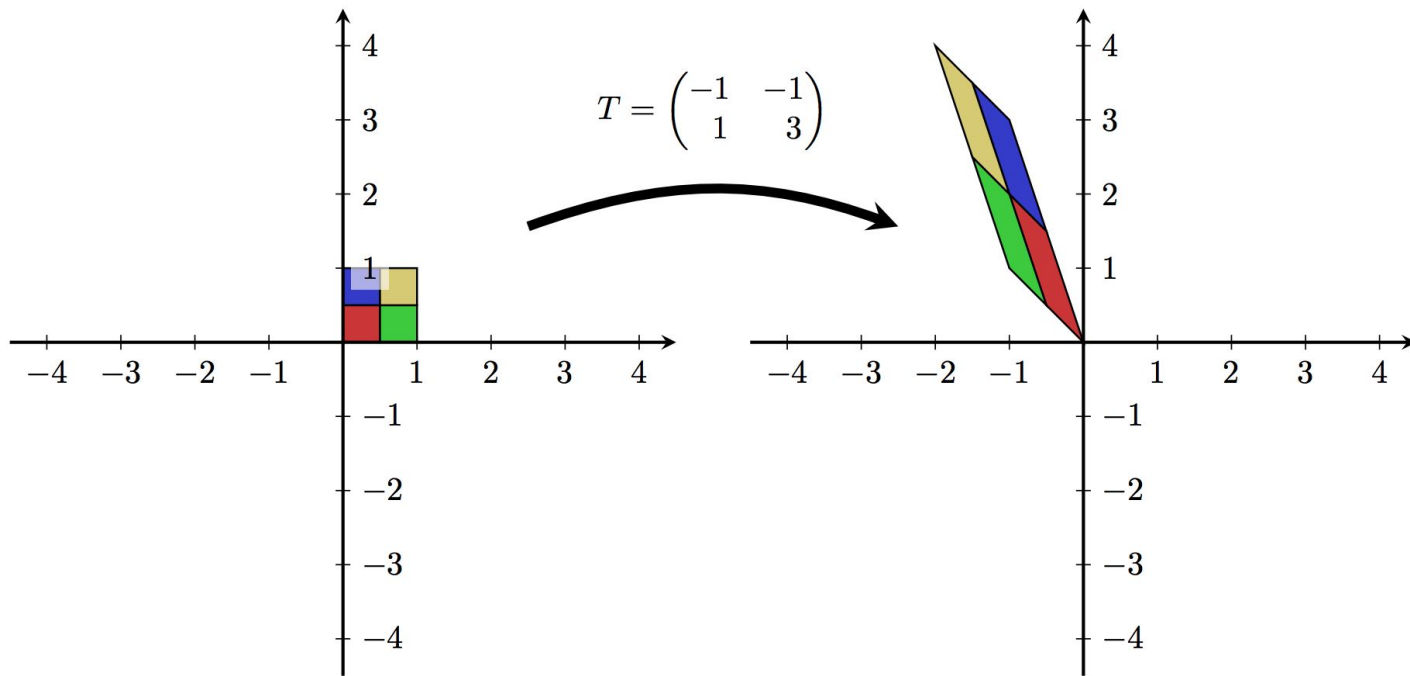
Обратная матрица к данной — это матрица при перемножении которой с текущей матрицей получается единичная матрица.

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

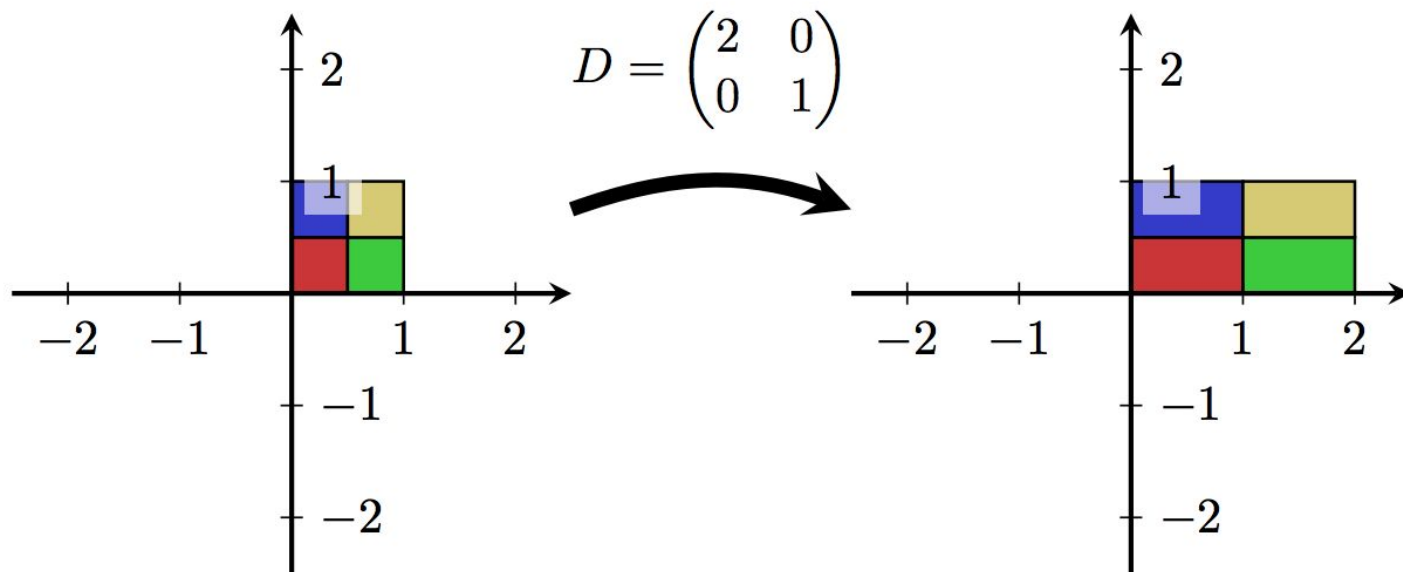
Например:

$$B = \begin{bmatrix} 2 & 5 & 7 \\ 6 & 3 & 4 \\ 5 & -2 & -3 \end{bmatrix} \quad B^{-1} = \begin{bmatrix} 1 & -1 & 1 \\ -38 & 41 & -34 \\ 27 & -29 & 24 \end{bmatrix}$$
$$B \cdot B^{-1} = \begin{bmatrix} 2 & 5 & 7 \\ 6 & 3 & 4 \\ 5 & -2 & -3 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 1 \\ -38 & 41 & -34 \\ 27 & -29 & 24 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}$$

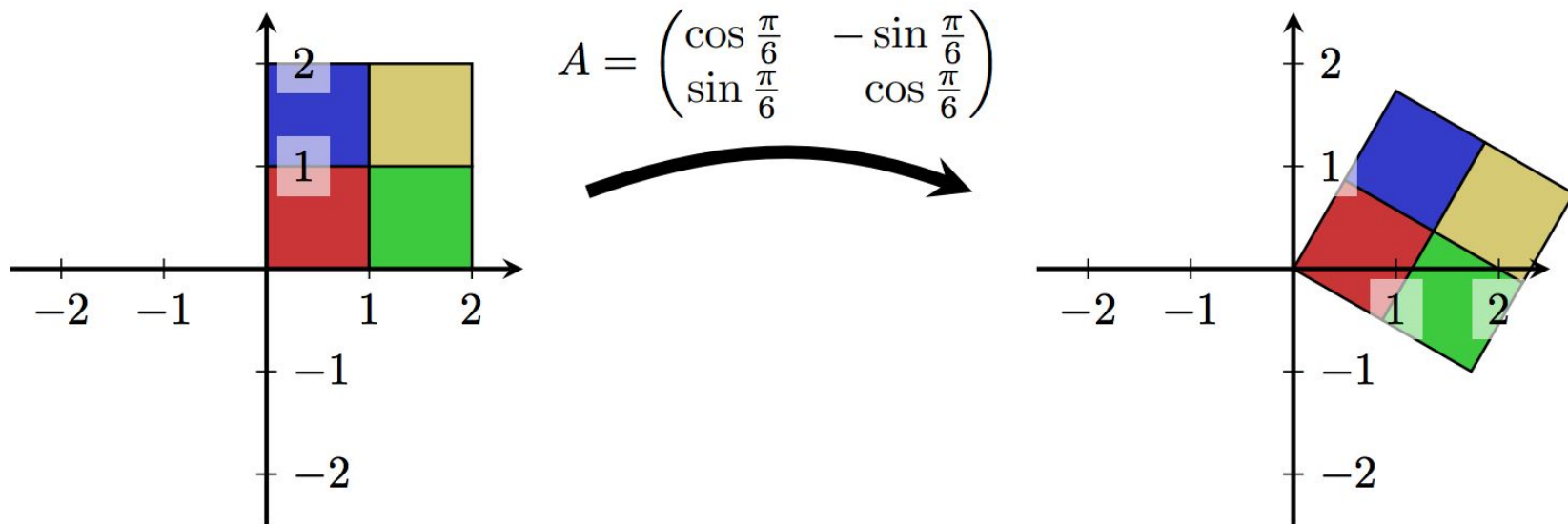
Типы матриц (преобразование пространства)



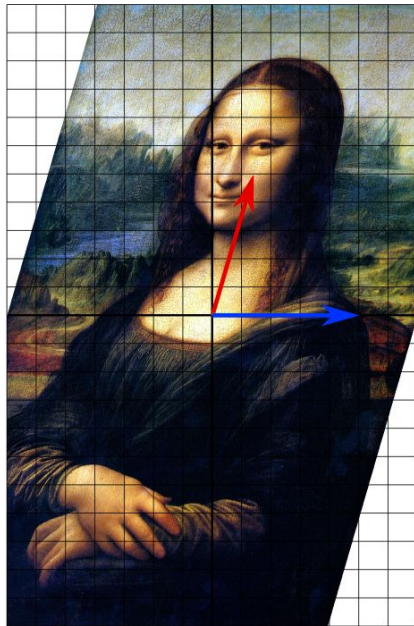
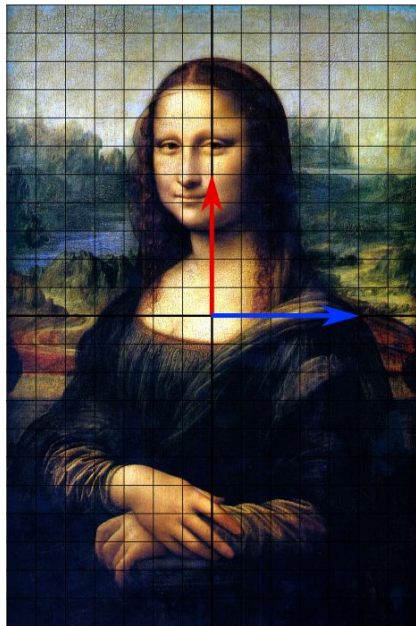
Типы матриц (преобразование пространства)



Типы матриц (преобразование пространства)



Собственные векторы и собственные значения



Собственный вектор преобразования A

$$AX = \lambda X$$

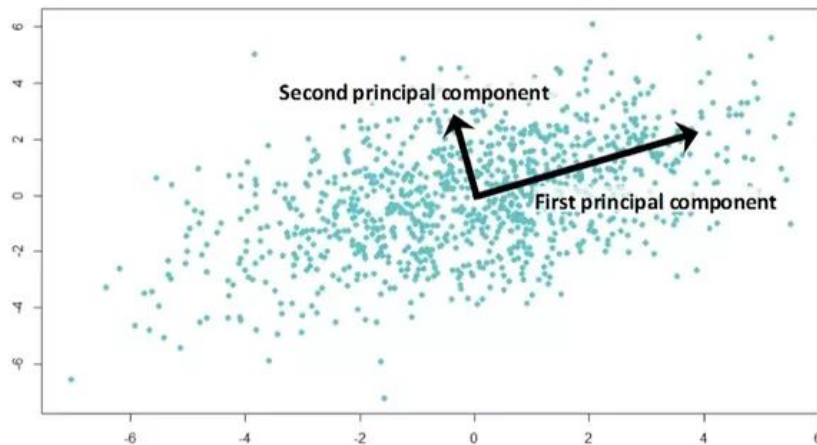
X - собственный вектор (ненулевой!)
 λ - собственное значение

!

У матрицы $n \times n$ не более n
собственных значений

Понижение размерности: РСА

- 1) Вычисляем матрицу ковариаций признаков
- 2) Находим собственные вектора матрицы ковариаций
- 3) Первые k векторов соответствующих k максимальным собственным значениям - компоненты нашего разложения



<https://medium.com/@sadatnazrul/the-dos-and-donts-of-principal-component-analysis-7c2e9dc8cc48>

Плюсы: возможность регуляции получаемой размерности (добавлении компонент по одной в зависимости от объяснённой дисперсии); скорость алгоритма; интерпретация

Минусы: линейность, предположение об ортогональности

Матричные разложения (спектральное разложение)

Разложение матрицы - представление в виде произведения некоторых других, обладающих интересными свойствами.

Пример: спектральное разложение X

$$X = S^T \cdot D \cdot S$$



X - симметричная, S - ортогональная, D - диагональная из собственных значений X .

Часто встречаются квадратичные формы

$$f(y) = y^T X y$$

с помощью спектрального разложения приводим к более простому виду:

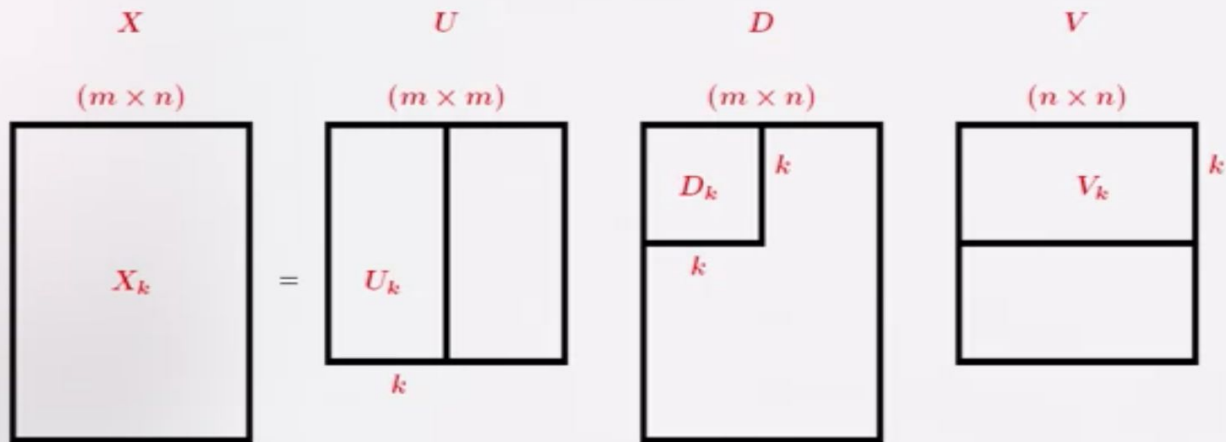
$$f(y) = y^T \cdot S^T \cdot D \cdot S \cdot y = (S \cdot y)^T \cdot D \cdot (S \cdot y) = z^T \cdot D \cdot z = \sum_{i=1}^n \lambda_i z_i^2,$$

Матричные разложения (сингулярное разложение)

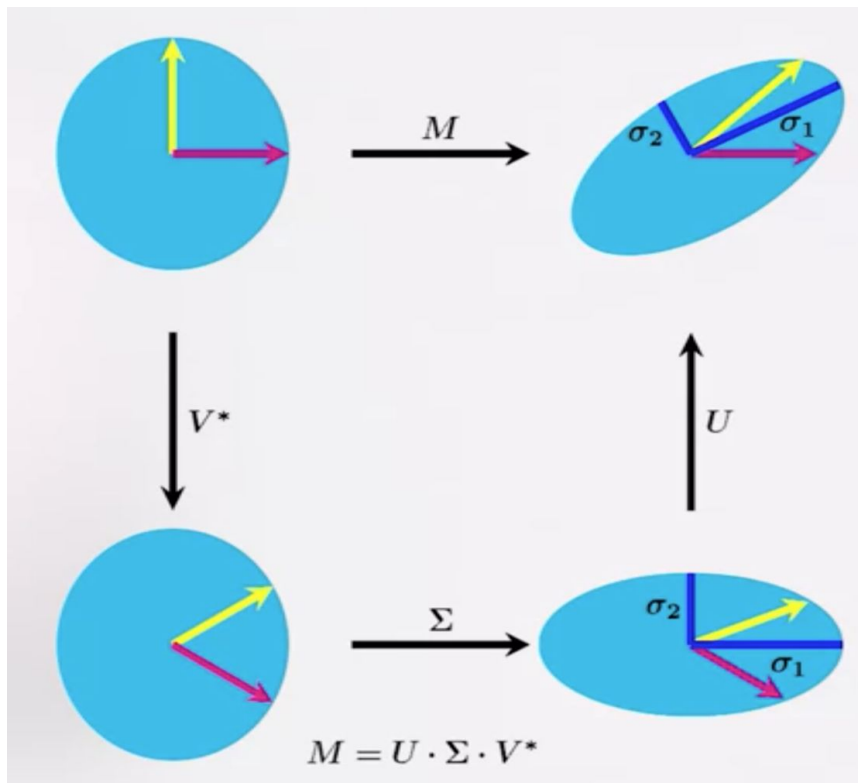
Но это была симметричная матрица, что в случае произвольной?

$$\triangleright X = UDV$$

$\triangleright U, V$ — ортогональные, D —
диагональная



Матричные разложения (сингулярное разложение)



Сингулярное разложение представляет линейное преобразование в виде композиции: вращения, растяжения по осям, вращения.

Матричные разложения (приближение матрицей меньшего ранга)

- Матрица задаёт отображение, ранг в какой-то степени мера “сложности” отображения
- Ранг - максимальное количество линейно независимых столбцов или строк
- Ранг - максимальный размер подматрицы с ненулевым определителем



$\text{rank}(X) \leq \min(n, m)$, если X - матрица $m \times n$

Пусть $X = AB$, A размера (m, k) , B размера (k, n)

Пусть также $k < m$, $k < n$

Что можно сказать о ранге X ?

Матричные разложения (приближение матрицей меньшего ранга)

Зачем приближать матрицу матрицей меньшего ранга?

Мы предполагаем, что матрица преобразования X на самом деле более простая.

Что значит приблизить

$$\triangleright X \approx X' = UV^T$$

$$\triangleright U - m \times k, V - n \times k$$

Просто наилучшее приближение по
норме: $\|X - UV^T\| \rightarrow \min$

Матричные разложения (приближение матрицей меньшего ранга)

Что значит приблизить

$$\triangleright X \approx X' = UV^T$$

$$\triangleright U - m \times k, V - n \times k$$

Просто наилучшее приближение по
норме: $\|X - UV^T\| \rightarrow \min$

$$\|X\|_F = \sqrt{\sum_{i,j} x_{ij}^2}$$

Итоговая задача выглядит так:

$$U, V = \underset{U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times k}}{\operatorname{argmin}} \sum_{i,j} (x_{ij} - u_i^T v_j)^2$$

Матричные разложения (пример применения)

- › Пусть X – матрица признаков объектов
- › Тогда U – матрица новых признаков
- › При $k < n$ преобразование признаков понижает размерность пространства
- › По U с максимальной возможной точностью восстанавливаются исходные признаки X

Матричные разложения (пример применения)

- › Пусть X – матрица с оценками x_{ij} , поставленными пользователем i фильму j
- › Некоторые значения матрицы неизвестны
- › $x_{ij} \approx \widehat{x_{ij}} = u_i v_j$, где u_i отражает интересы пользователя, а v_j – признаковое описание фильма
- › Идея: настроим u_i и v_j на известных x_{ij} , а неизвестные спрогнозируем
- › Будем рекомендовать фильмы, для которых спрогнозирована высокая оценка

Что делать с пропущенными значениями?

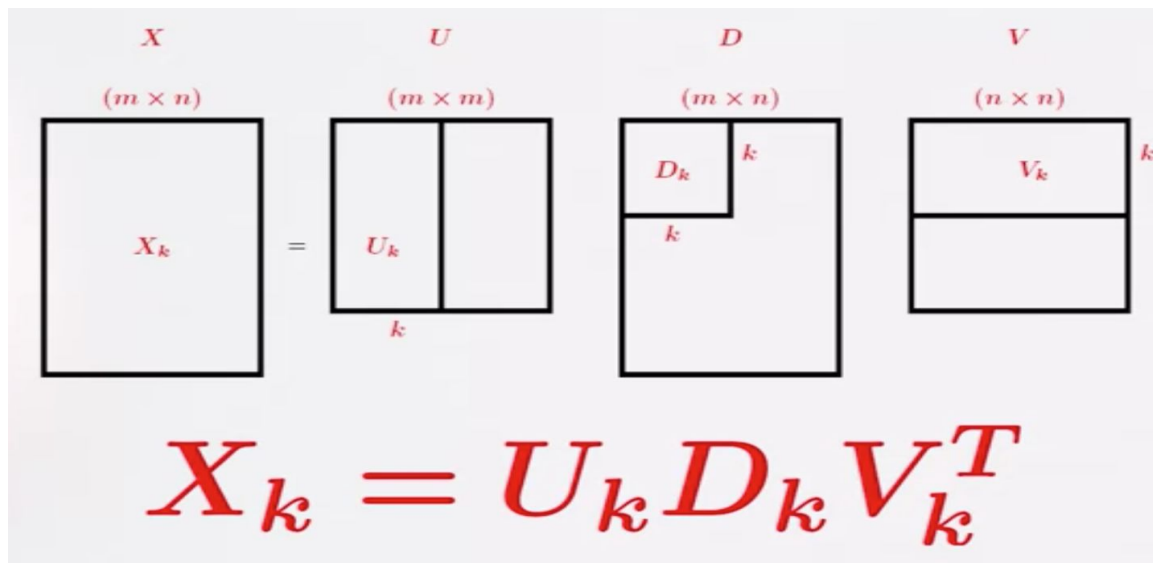
Матричные разложения (пример применения)

- › Пусть X – матрица с оценками x_{ij} , поставленными пользователем i фильму j
- › Некоторые значения матрицы неизвестны
- › $x_{ij} \approx \widehat{x_{ij}} = u_i v_j$, где u_i отражает интересы пользователя, а v_j – признаковое описание фильма
- › Идея: настроим u_i и v_j на известных x_{ij} , а неизвестные спрогнозируем
- › Будем рекомендовать фильмы, для которых спрогнозирована высокая оценка

$$U, V = \underset{U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times k}}{\operatorname{argmin}} \sum_{i, j: x_{ij} \neq 0} (x_{ij} - u_i^T v_j)^2$$

Матричные разложения (связь SVD и низкорангового приближения)

$$\hat{X} = \underset{\text{rg } \hat{X} \leq k}{\operatorname{argmin}} \|X - \hat{X}\| \quad X = U \cdot D \cdot V^T$$



Усечённый SVD

Матричные разложения (связь SVD и низкорангового приближения)

Оказывается, X_k - наилучшее приближение матрицы X матрицей ранга $\leq k$ по норме Фробениуса!

$$X_k = U_k D_k V_k^T$$

$$\hat{X}_k = \operatorname{argmin}_{\operatorname{rg}(\hat{X}) \leq k} ||X - \hat{X}||_F$$

$$||X - \hat{X}||_F = \sqrt{\sum_{i,j} (x_{ij} - \hat{x}_{ij})^2}$$

Матричные разложения (рекомендательные системы)

› Вариант 1 (не очень правильно, но просто): сделать SVD, матрицу $U_k D_k$ использовать как матрицу профилей пользователей, а матрицу V_k как матрицу профилей фильмов, произведение профилей – прогноз оценки фильма

› Вариант 2 (более правильно, но нужно более глубоко вникнуть в метод): Не будем никак использовать SVD, а просто подберем U и V , минимизируя функционал

Производная функции

Производная - мгновенная скорость роста функции в заданной точке.

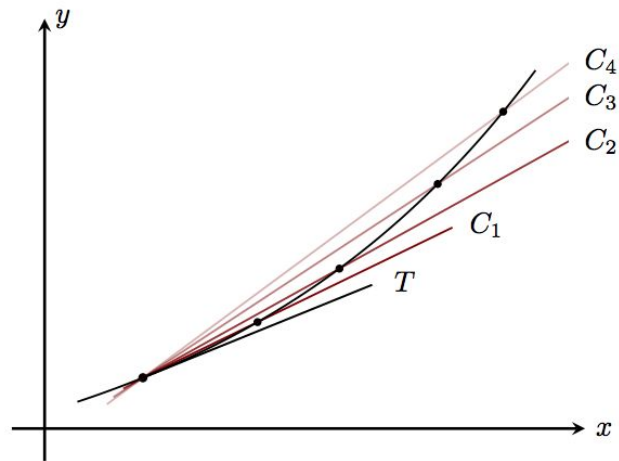
$$\frac{f(x + \Delta x) - f(x)}{\Delta x} = k.$$

Давайте посмотрим на линейную функцию $y=kx+b$

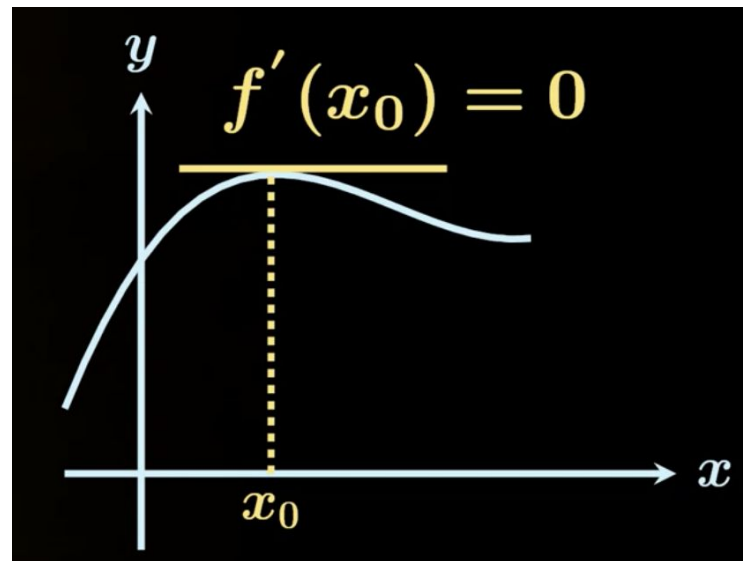
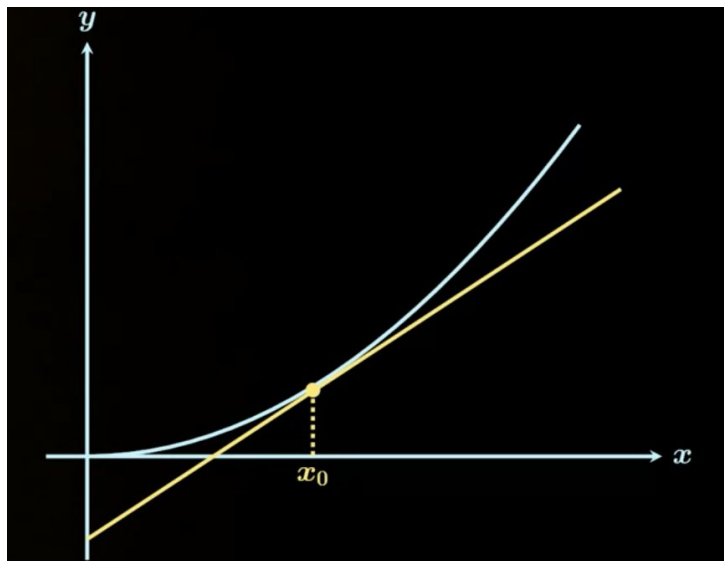
Как понять скорость роста для произвольной функции? **Предел!**

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

Гладкие функции - функции, производная которых непрерывна.

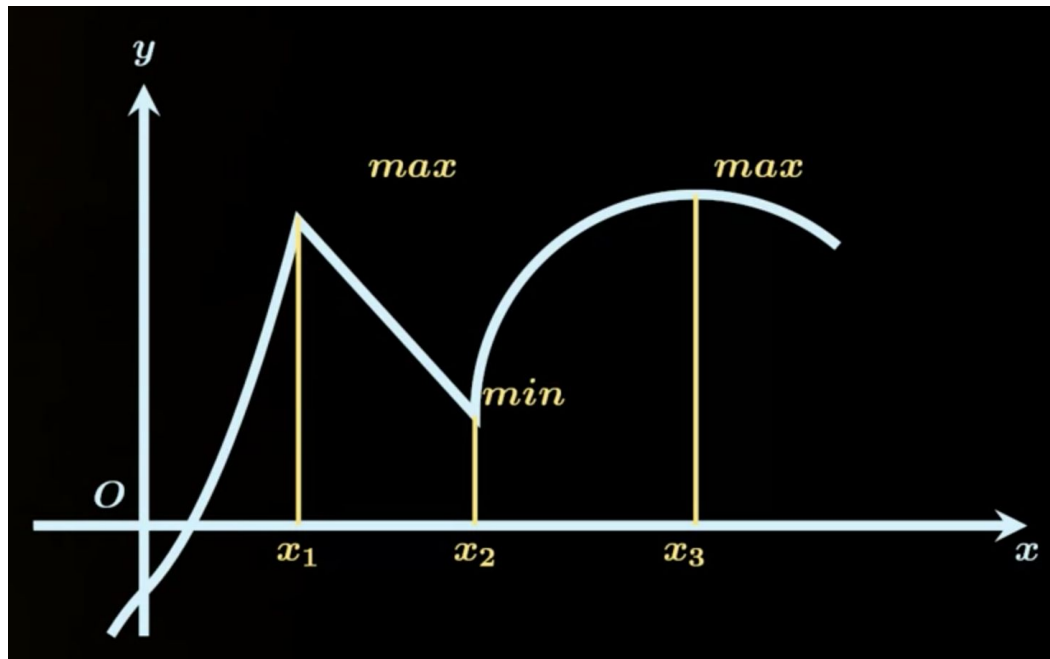
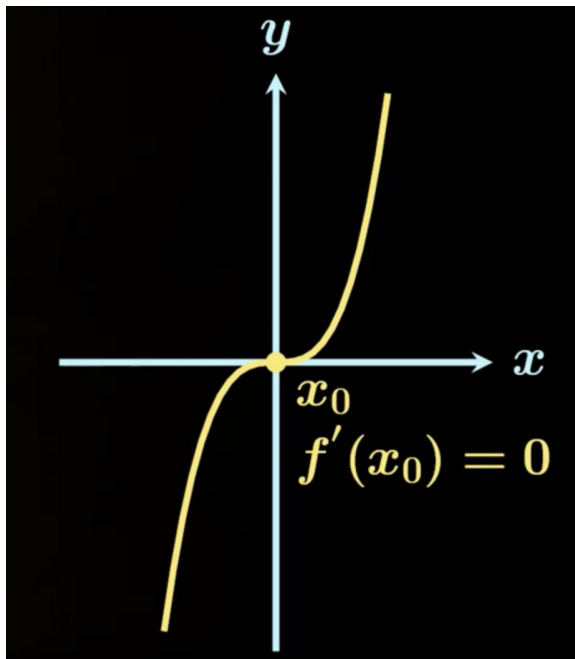


Экстремум функции и производная



В точках локальных экстремумов производная (если она определена (!), пример дальше) обязана равняться нулю.
Это **необходимое** условие.

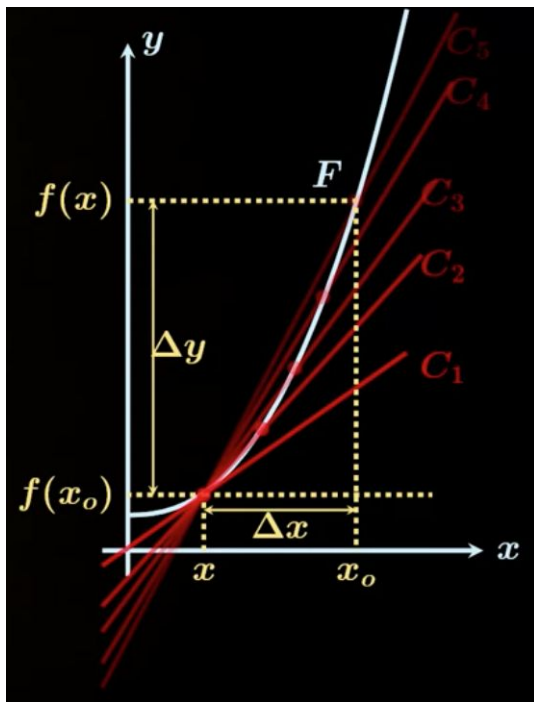
Экстремум функции и производная



Однако равенство нулю производной **не является достаточным** условием локального экстремума. Также производная может быть вовсе не определена в точках локальных экстремумов.

Функция нескольких переменных

Из одномерного случая помним: геометрический смысл производной - угловой коэффициент касательной.



Как посчитать производную функции нескольких переменных?

$$\triangleright \frac{\Delta f}{\Delta x} \xrightarrow{\Delta x \rightarrow 0} f'_x, y \text{ — фиксирован}$$

$$\triangleright \frac{\Delta f}{\Delta y} \xrightarrow{\Delta y \rightarrow 0} f'_y, x \text{ — фиксирован}$$

Частная производная



Частная производная — обобщение понятия производной на случай функции нескольких переменных.



Частная производная функции $f(x, y)$ по x определяется как производная по x , взятая в смысле функции одной переменной, при условии постоянства оставшейся переменной y .

$$f'_x(x, y) = \lim_{h \rightarrow 0} \frac{f(x + h, y) - f(x, y)}{h}, \quad f'_y(x, y) = \lim_{h \rightarrow 0} \frac{f(x, y + h) - f(x, y)}{h}.$$

Градиент и линии уровня функции

Если $f(x_1, \dots, x_n)$ — функция n переменных x_1, \dots, x_n , то n -мерный вектор из частных производных:

$$\text{grad } f = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)$$



называется **градиентом функции**.



Линией уровня функции называется множество точек, в которых функция принимает одно и то же фиксированное значение. Оказывается, что **градиент перпендикулярен линии уровня**.

Градиент в задачах оптимизации

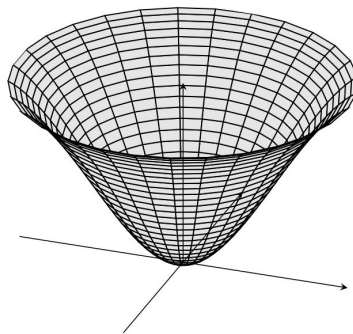
Задачей оптимизации называется задача по нахождению экстремума функции, например минимума:

$$f(x_1, \dots, x_n) \rightarrow \min$$

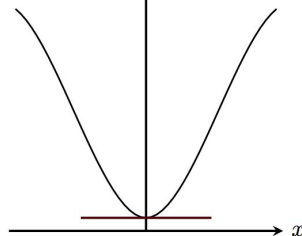
Такая задача часто встречается в приложениях, например при выборе оптимальных параметров рекламной компании, а также в задачах классификации.

Градиент в задачах оптимизации

Но не всегда задачу можно решать аналитически. В таком случае используется численная оптимизация. Наиболее простым в реализации из всех методов численной оптимизации является метод градиентного спуска.



$$\left. \frac{\partial f}{\partial x} \right|_{(0,0)} = 0$$



$$\left. \frac{\partial f}{\partial y} \right|_{(0,0)} = 0$$

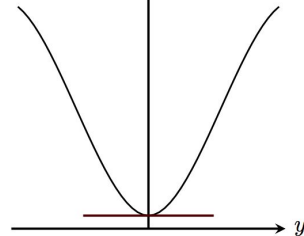


Рис. 2: Функция двух переменных достигает минимума в начале координат.

Градиентный спуск

Это итерационный метод. Решение задачи начинается с выбора начального приближения $\vec{x}^{[0]}$

После вычисляется приблизительное значение \vec{x}^1

Затем \vec{x}^2

и так далее...

! $\vec{x}^{[j+1]} = \vec{x}^{[j]} - \gamma^{[j]} \nabla F(\vec{x}^{[j]}),$ где $\gamma^{[j]}$ — шаг градиентного спуска.

Идея: идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla F$.

Градиентный спуск

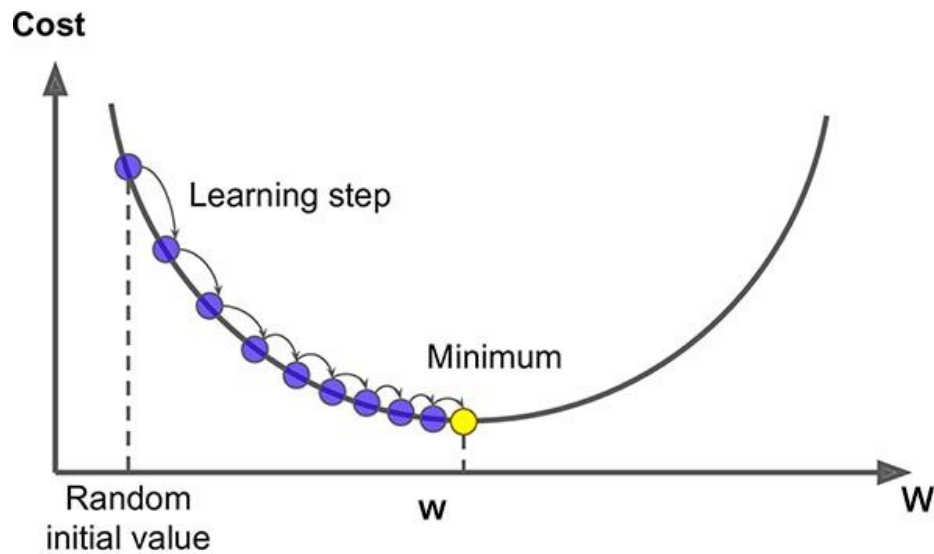
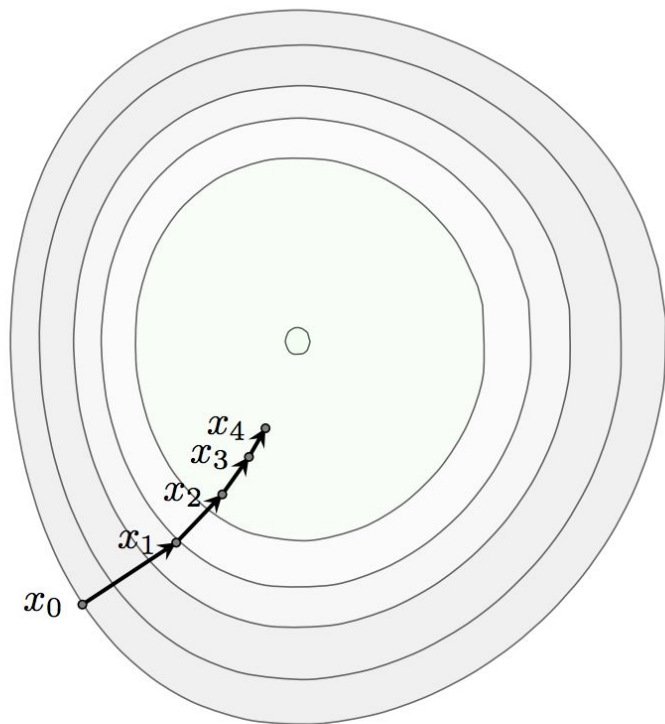


Рис. 3: Градиентный спуск

Список полезной литературы

- <https://habr.com/ru/post/319144/>
- <https://habr.com/ru/post/318970/>
- <https://blog.statsbot.co/recommendation-system-algorithms-ba67f39ac9a3>
- <https://habr.com/ru/company/surfingbird/blog/139863/>
- <https://habr.com/ru/post/304214/>
- http://mathprofi.ru/chastnye_proizvodnye_primery.html
- https://www.youtube.com/watch?v=glsWNZSPK9w&ab_channel=SKILLUP
- https://function-x.ru/return_matrix.html
- <https://habr.com/ru/post/147807/>
- <https://habr.com/ru/post/359016/>