Дополнительная лекция

Вектора и операции над ними. Операции с матрицами. Производная и градиент. Градиентный спуск. Дифференциальная эволюция. Матричные разложения. Понижение размерности.

Даниил Корбут
Специалист по Анализу Данных

Даниил Корбут
ML Engineer,
Deliveroo

Окончил магистратуру ФИВТ МФТИ в 2020.
Работал в Яндекс.Алиса, Insilico Medicine, Microsoft, PicsArt.

Сейчас работаю на позиции ML Engineer в Deliveroo.

НЕТОЛОГИЯ

Векторы - применение

Признак - отображение из множества объектов в множество допустимых значений этого признака.
Если задано множество объектов и некоторый набор признаков, для каждого объекта можно построить его **признаковое описание — вектор**, составленный из значений этого набора признаков на данном объекте.

Так как алгоритмы не умеют работать с текстовыми описаниями, картинками явно, мы вынуждены переводить их в векторные/матричные представления, для которых подключается математический аппарат для работы с ними.

Векторы

Вектор — упорядоченный конечный список чисел.
Вектора обычно записываются как вертикальный список, например:

$$\begin{bmatrix} -1.1 \\ 0.0 \\ 3.6 \\ -7.2 \end{bmatrix} \qquad \begin{pmatrix} -1.1 \\ 0.0 \\ 3.6 \\ -7.2 \end{pmatrix}$$

Вектор может быть записан также в следующем виде:

$$(-1.1, 0.0, 3.6, -7.2)$$

Скалярное произведение векторов

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

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

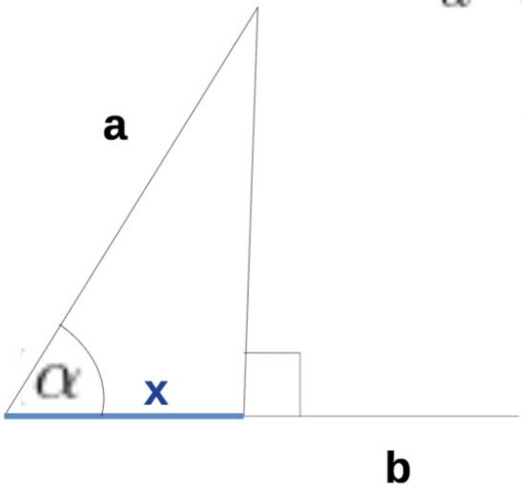
Если известны координаты векторов, то скалярное произведение можно посчитать по формуле:

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

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

Проекция одного вектора на другой

Длина вектора x, полученного в результате проекции вектора a на вектор b, равна делению скалярного произведения вектора **a** на вектор **b** на длину b.



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

$$cos(\alpha) = \frac{|x|}{|a|}$$

$$|x| = cos(\alpha) \cdot |a|$$

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

$$|x| = \frac{\overrightarrow{a} \cdot \overrightarrow{b}}{|b|}$$

Нормированные пространства

Для обобщения понятия длины вектора используется понятие нормы. Функция $\|\cdot\| : V \to \mathbb{R}$ называется нормой в векторном пространстве V, если для нее выполняются аксиомы нормы:

1. $\|x\| = 0 \iff x = 0$ (Нулевую норму имеет только нулевой вектор)
2. $\forall x, y \in L : \|x + y\| \leq \|x\| + \|y\|$ (Неравенство треугольника)
3. $\forall \alpha \in \mathbb{R}\ \forall x \in V : \|\alpha x\| = |\alpha| \|x\|$ (Условие однородности)

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}, \qquad \|x\|_1 = \sum_{i=1}^{n} |x_i|.$$

Метрические пространства

Понятие расстояние обобщается с помощью понятия метрики. Пусть X — некоторое множество, а числовая функция $d : X \times X \to \mathbb{R}$, которая называется метрикой, удовлетворяет следующим условиям:

1. $d(x,\ y) = 0 \Leftrightarrow x = y$ (аксиома тождества).
2. $d(x,\ y) = d(y,\ x)$ (аксиома симметрии).
3. $d(x,\ z) \leqslant d(x,\ y) + d(y,\ z)$ (неравенство треугольника).

Любое нормированное пространство можно превратить в метрическое, определив функцию расстояния $d(x,\ y) = \|y - x\|$. Например, Евклидова и Манхэттенская метрики имеют вид:

$$\rho_2(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}, \qquad \rho_1(x,y) = \sum_{i=1}^{n}|x_i - y_i|$$

Транспонирование матрицы

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

Транспонирование матрицы — это замена строк на столбцы.

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

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:

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

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

Перемножение матриц

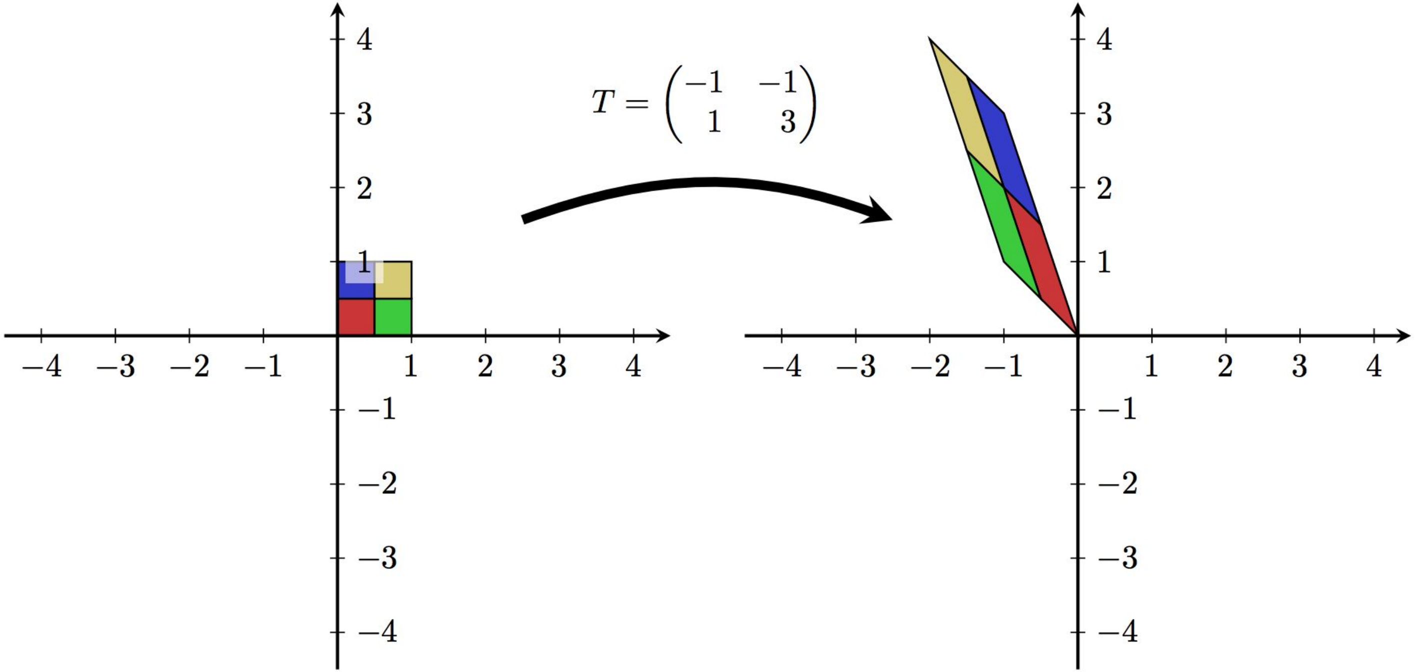
Даны 2 матрицы: A и B. Умножение матрицы A на B можно выполнить, если количество столбцов матрицы A равно количеству строк матрицы B.

$$A \cdot B = \begin{pmatrix} 1 & 2 & 2 \\ 3 & 1 & 1 \end{pmatrix} \begin{pmatrix} 4 & 2 \\ 3 & 1 \\ 1 & 5 \end{pmatrix} = \begin{pmatrix} 1 \cdot 4 + 2 \cdot 3 + 2 \cdot 1 & 1 \cdot 2 + 2 \cdot 1 + 2 \cdot 5 \\ 3 \cdot 4 + 1 \cdot 3 + 1 \cdot 1 & 3 \cdot 2 + 1 \cdot 1 + 1 \cdot 5 \end{pmatrix} = \begin{pmatrix} 12 & 14 \\ 16 & 12 \end{pmatrix}$$

$$B \cdot A = \begin{pmatrix} 4 & 2 \\ 3 & 1 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} 1 & 2 & 2 \\ 3 & 1 & 1 \end{pmatrix} =$$

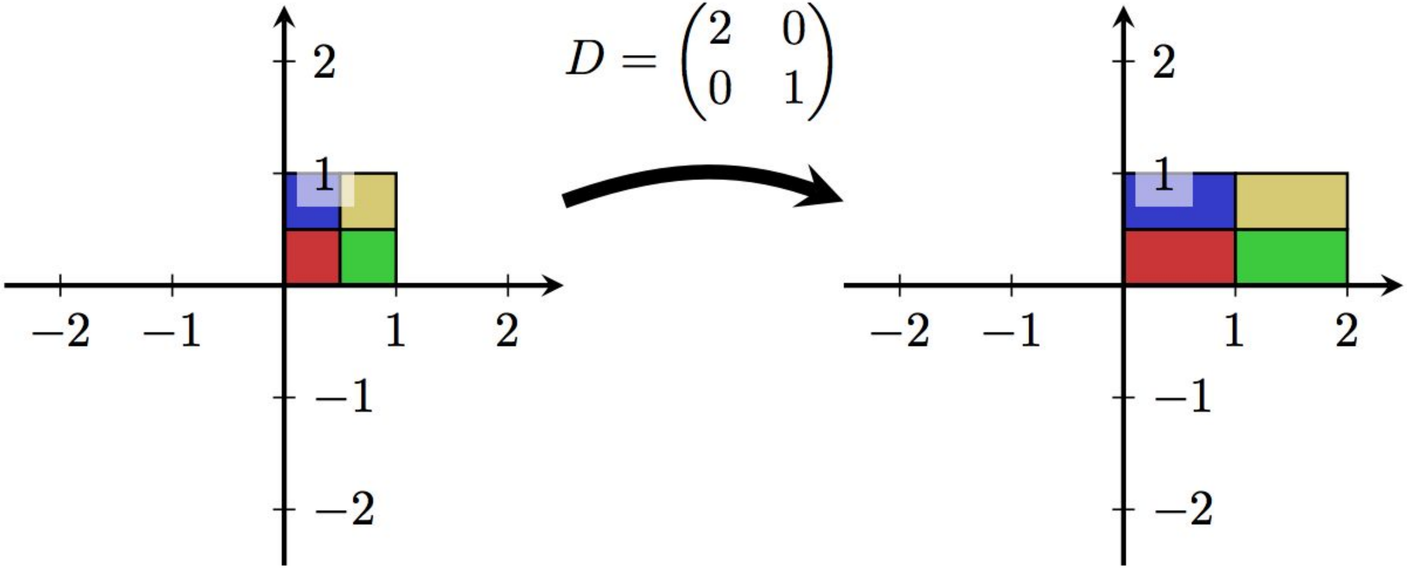
$$\begin{pmatrix} 4 \cdot 1 + 2 \cdot 3 & 4 \cdot 2 + 2 \cdot 1 & 4 \cdot 2 + 2 \cdot 1 \\ 3 \cdot 1 + 1 \cdot 3 & 3 \cdot 2 + 1 \cdot 1 & 3 \cdot 2 + 1 \cdot 1 \\ 1 \cdot 1 + 5 \cdot 3 & 1 \cdot 2 + 5 \cdot 1 & 1 \cdot 2 + 5 \cdot 1 \end{pmatrix} = \begin{pmatrix} 10 & 10 & 10 \\ 6 & 7 & 7 \\ 16 & 7 & 7 \end{pmatrix}$$

Типы матриц (преобразование пространства)



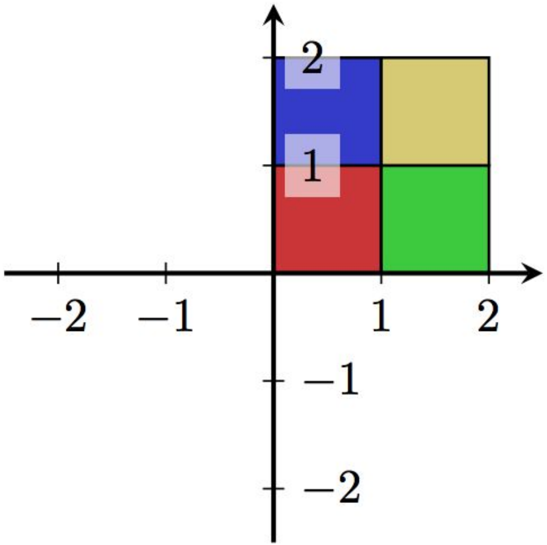
https://www.coursera.org/learn/mathematics-and-python

Типы матриц (преобразование пространства)

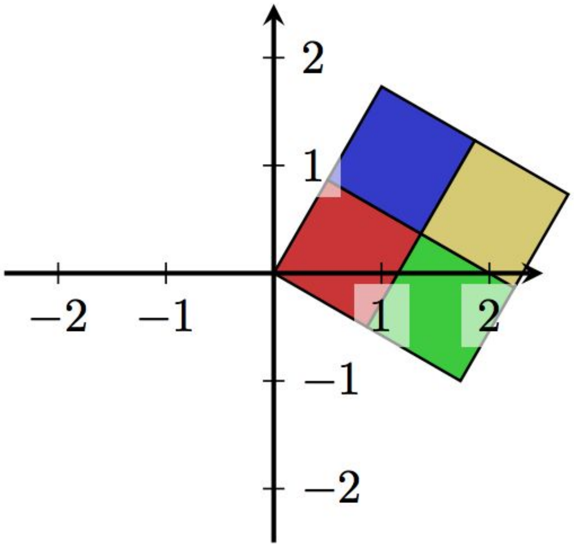


https://www.coursera.org/learn/mathematics-and-python

Типы матриц (преобразование пространства)

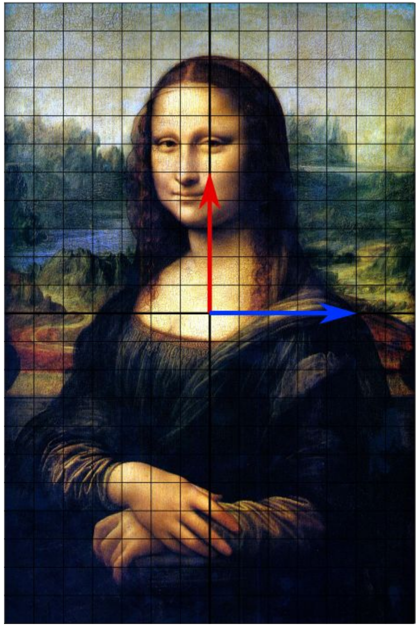


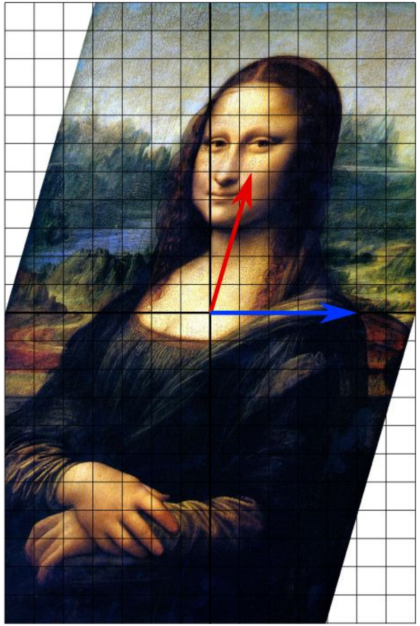
$$A = \begin{pmatrix} \cos\frac{\pi}{6} & -\sin\frac{\pi}{6} \\ \sin\frac{\pi}{6} & \cos\frac{\pi}{6} \end{pmatrix}$$



https://www.coursera.org/learn/mathematics-and-python

Собственные векторы и собственные значения





Собственный вектор преобразования A

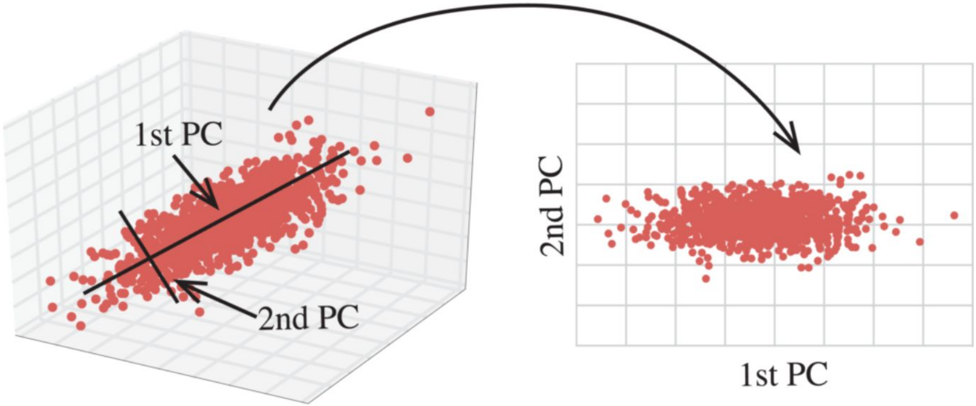
$$AX = \lambda X$$

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

!

У матрицы **n x n** не более **n** собственных значений

Понижение размерности: PCA



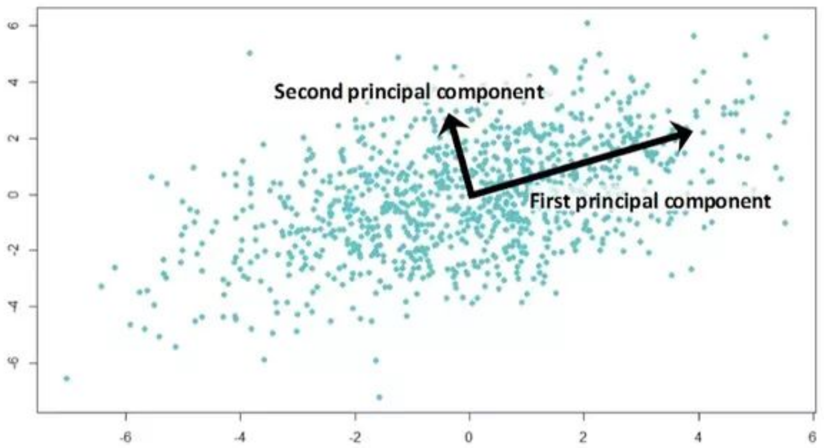
Один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации.

https://habr.com/ru/post/304214/

НЕТОЛОГИЯ

Понижение размерности: PCA

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

НЕТОЛОГИЯ

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

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

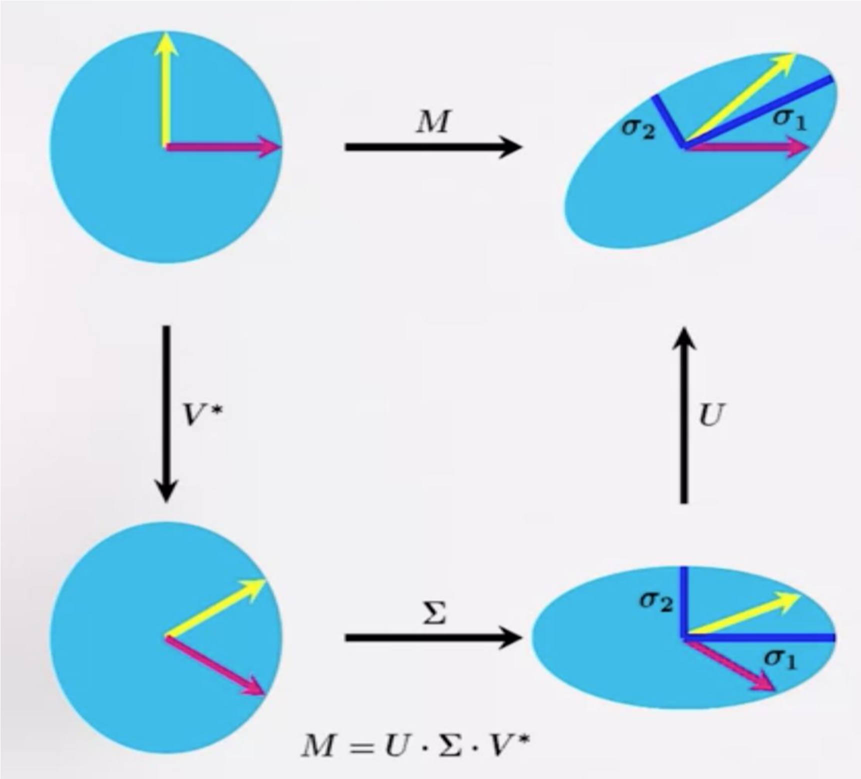
с помощью спектрального разложения приводим к более простому виду:

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

Матричные разложения (сингулярное разложение)

Но это была симметричная матрица, что в случае произвольной?

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

! rank(X) <= min(n, m), если X - матрица m x 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

- $X \approx X' = UV^T$
- $U - m \times k, V - n \times k$

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

Матричные разложения (приближение матрицей меньшего ранга)

Что значит приближить

- $X \approx X' = UV^T$
- $U - m \times k, V - n \times k$

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

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

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?

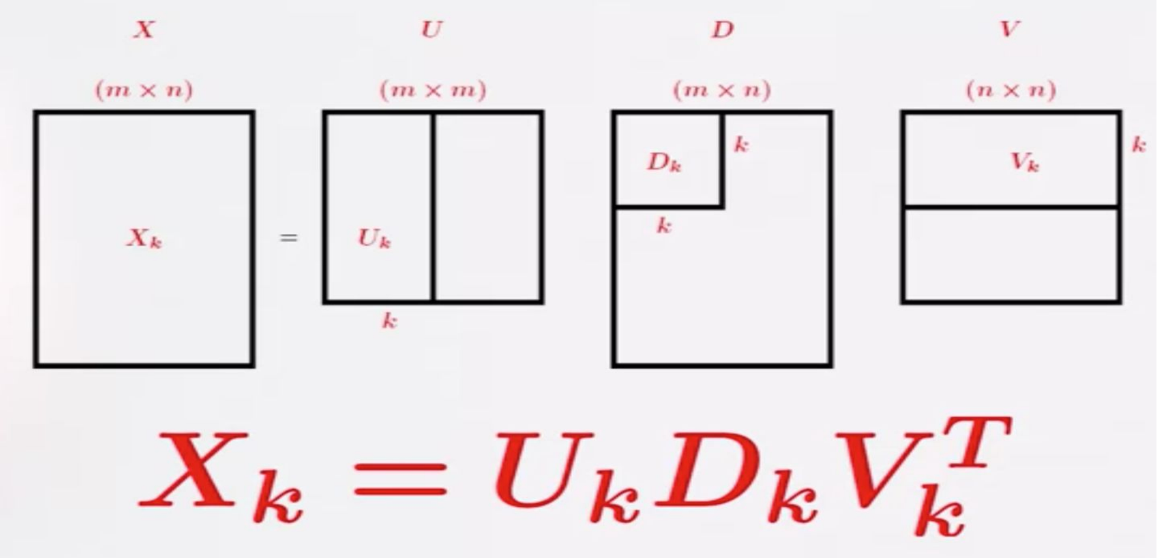
Матричные разложения (пример применения)

- 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)

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



Усечённый SVD

Матричные разложения (связь SVD и низкорангового приближения)

Оказывается, Xk - наилучшее приближение матрицы X матрицей ранга <= k по норме Фробениуса!

$$X_k = U_k D_k V_k^T$$

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

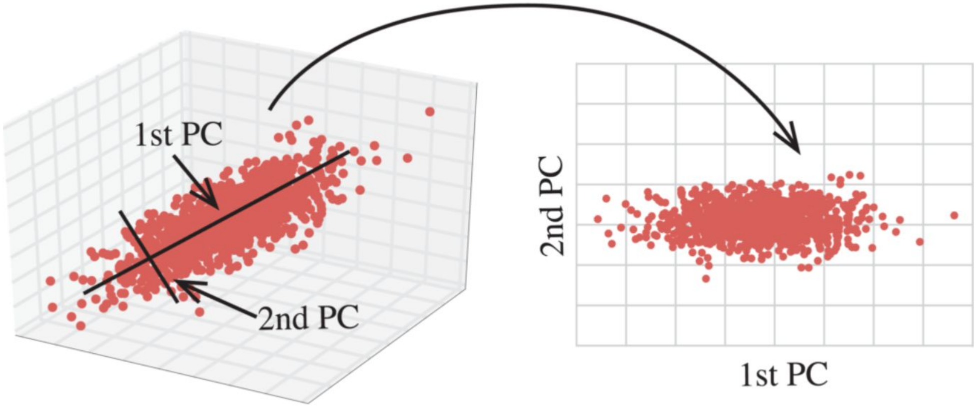
$$||X - \hat{X}||_F = \sqrt{\sum_{i,j} (x_{ij} - \hat{x}_{ij})^2}$$

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

Понижение размерности: PCA



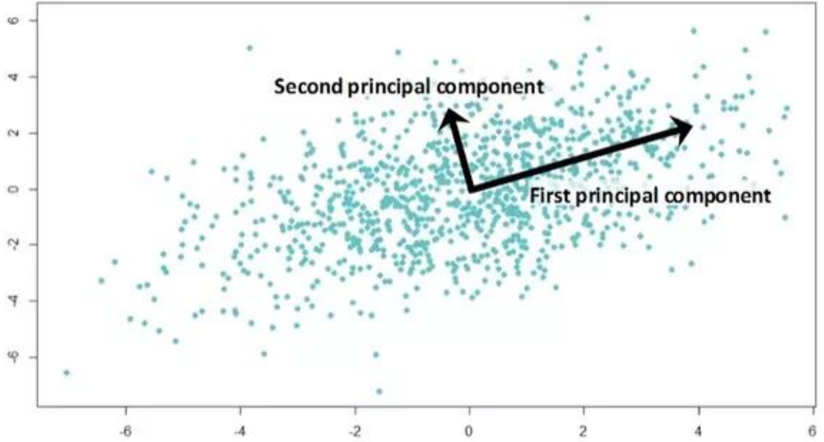
Один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации.

https://habr.com/ru/post/304214/

НЕТОЛОГИЯ

Понижение размерности: PCA

1) Вычисляем матрицу коввариаций 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

Производная функции

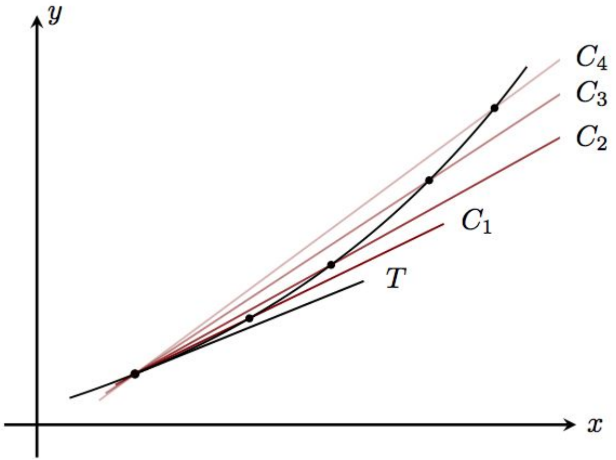
Производная - мгновенная скорость роста функции в заданной точке.

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

Давайте посмотрим на линейную функцию y=kx+b

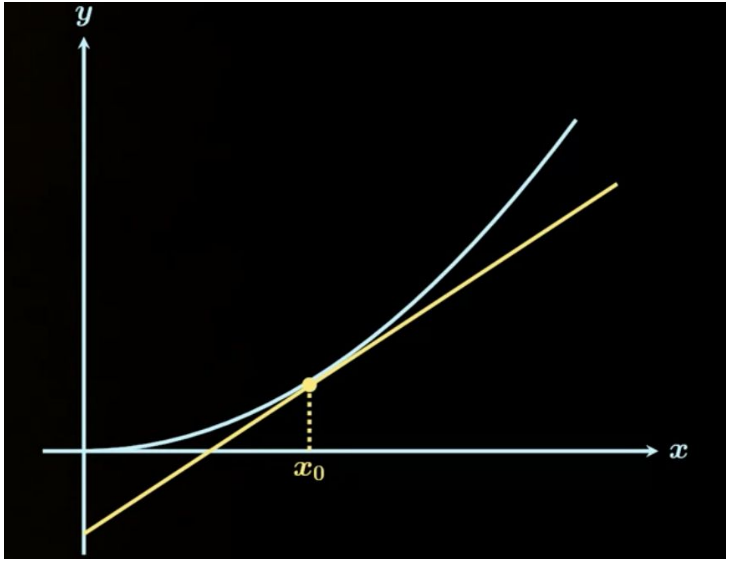
Как понять скорость роста для произвольной функции? **Предел!**

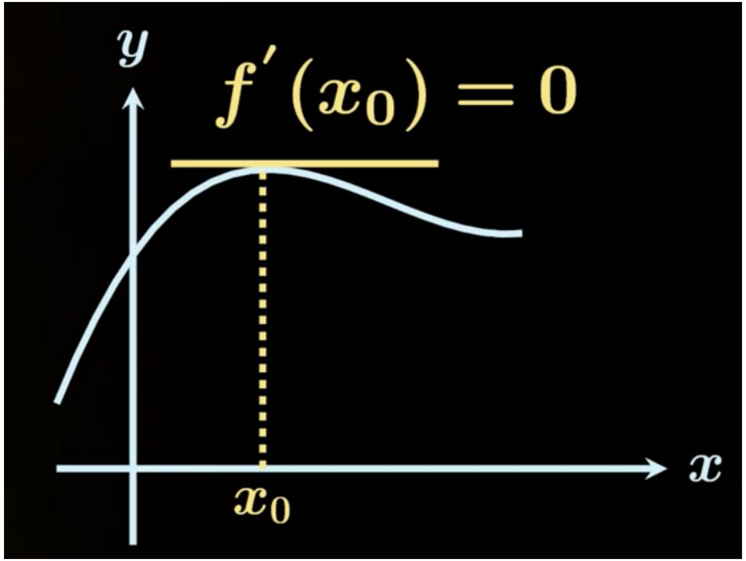
$$f'(x)=\lim_{\Delta x\to 0}\frac{f(x+\Delta x)-f(x)}{\Delta x}.$$



Гладкие функции - функции, производная которых непрерывна.

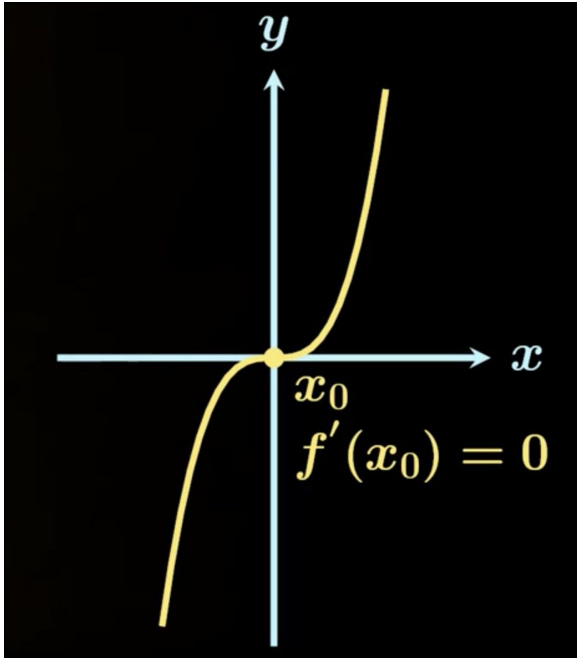
Экстремум функции и производная

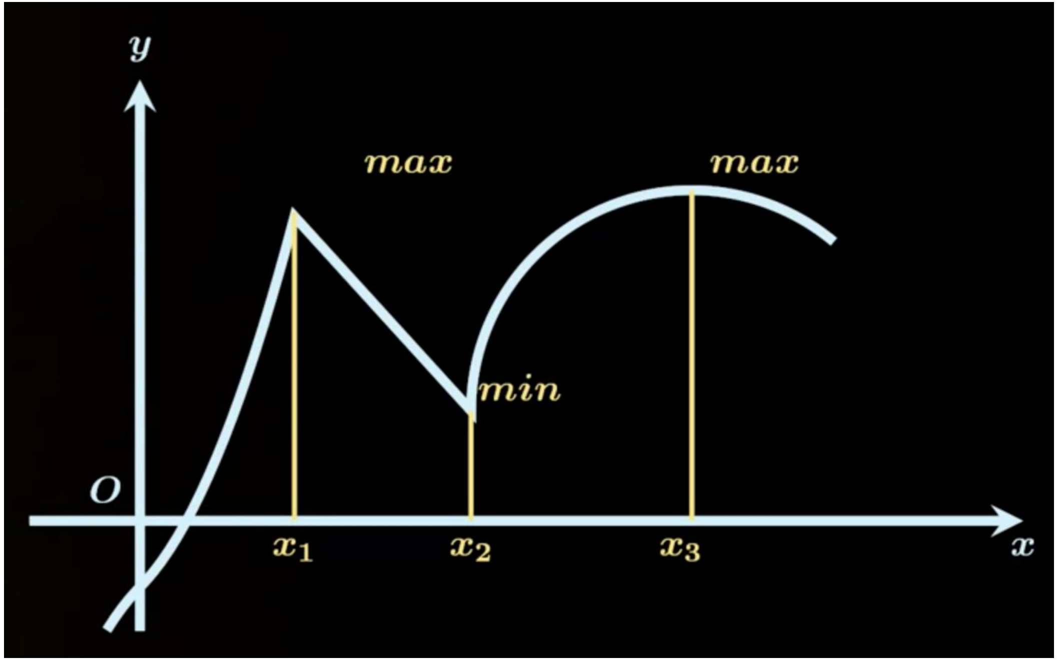




В точках локальных экстремумов производная (если она определена (!), пример дальше) обязана равняться нулю.
Это **необходимое** условие.

Экстремум функции и производная

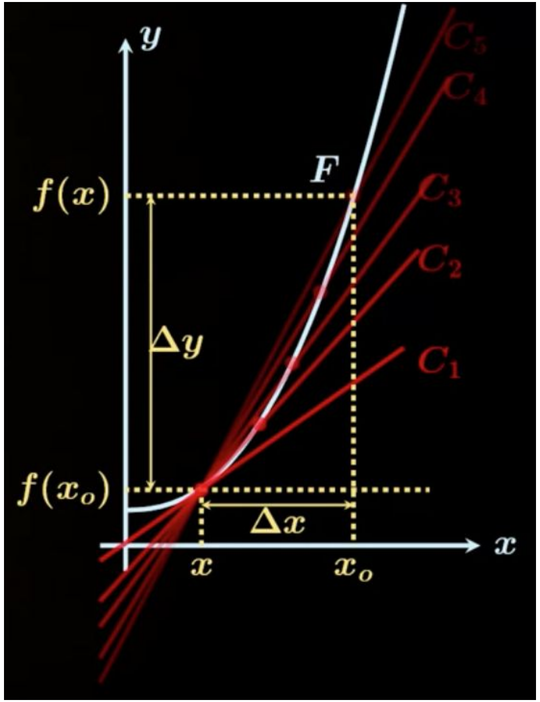




Однако равенство нулю производной **не является достаточным** условием локального экстремума. Также производная может быть вовсе не определена в точках локальных экстремумов.

Функция нескольких переменных

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



Как посчитать производную функции нескольких переменных?

- $\frac{\Delta f}{\Delta x} \xrightarrow{\Delta x \to 0} f'_x$, y — фиксирован
- $\frac{\Delta f}{\Delta y} \xrightarrow{\Delta y \to 0} f'_y$, x — фиксирован

https://www.coursera.org/learn/mathematics-and-python

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.

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

https://www.coursera.org/learn/mathematics-and-python

Градиент и линии уровня функции

Если f(x1, . . . , xn) — функция n переменных x1, ..., xn, то n-мерный вектор из частных производных:

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

!

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:

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

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.

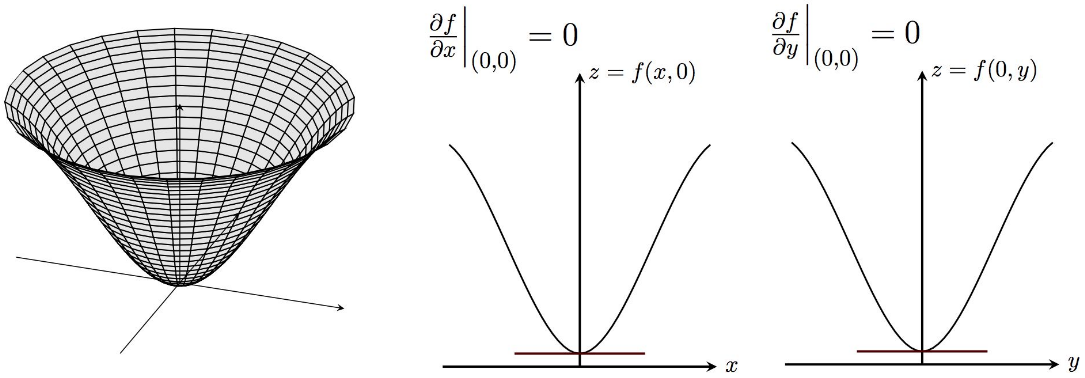


Рис. 2: Функция двух переменных достигает минимума в начале координат.

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…

!

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

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

Градиентный спуск

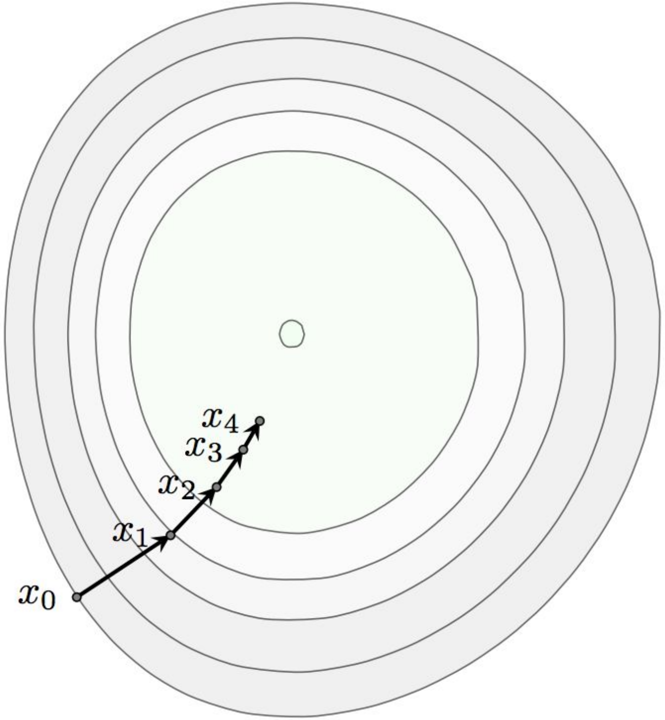
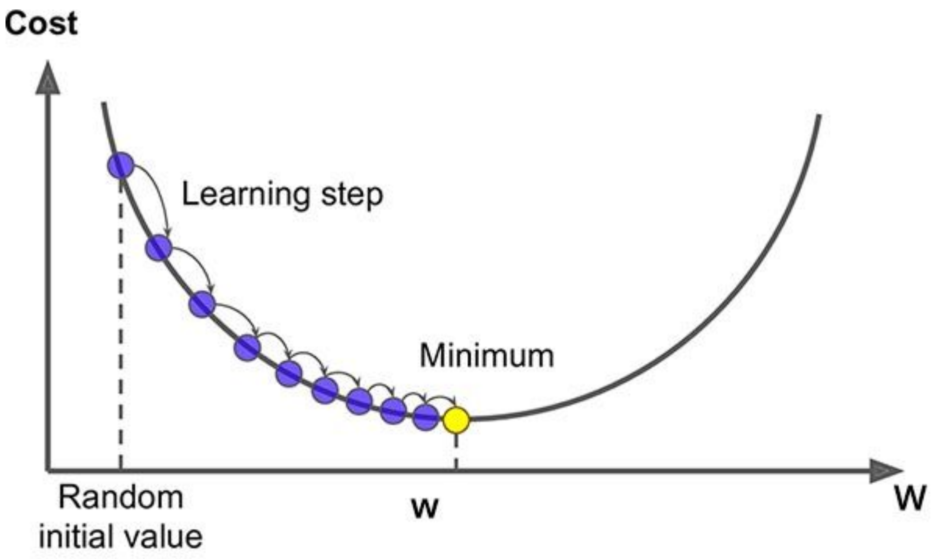


Рис. 3: Градиентный спуск



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/