Семинар 2

Математика для ML. Вычислительная линейная алгебра.

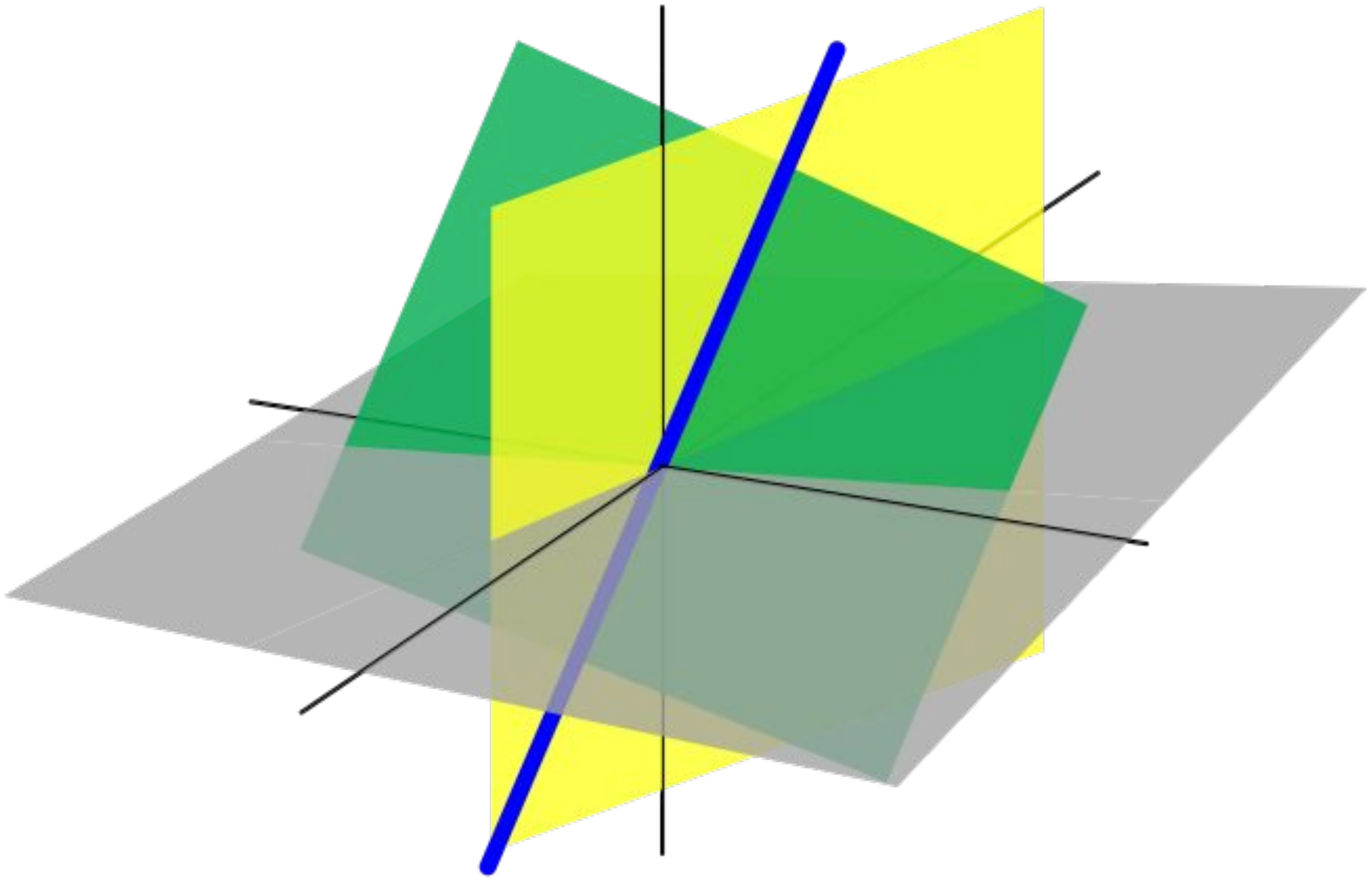
Сегодня мы…

- …узнаем, как линейная алгебра стала самым важным для компьютерных приложений разделом прикладной математики двадцатого века.
- …вспомним её основные понятия.
- …познакомимся с основными матричными разложениями. Научимся выводить PCA из интуитивных соображений.
- …осознаем рекомендательную систему на основе коллаборативной фильтрации наподобие тех, что раньше использовались в рекламе Яндекса.

План

- Мотивировка.
- Повторение: Основные понятия линейной алгебры.
- Матричные разложения.

  Вывод PCA.
- Коллаборативная фильтрация.



Мотивировка

Линейная алгебра — язык данных, понятный вашему компьютеру

За последние 30 лет мы научили машины

- Смотреть и видеть;
- Слушать, слышать и говорить в ответ;
- Сочинять стихи и песни,
- Прогнозировать котировки бирж;
- Находить раковые опухоли

и многое другое.

Диапазон приложений машинного обучения поражает воображение человека.

Тем более удивительным кажется тот факт, что на всё это способна обычная вычислительная машина.

Ликбез: компьютер это очень глупая коробка:) Это я вам как программист говорю.

Он понимает только нули и единицы, но делает это очень хорошо.

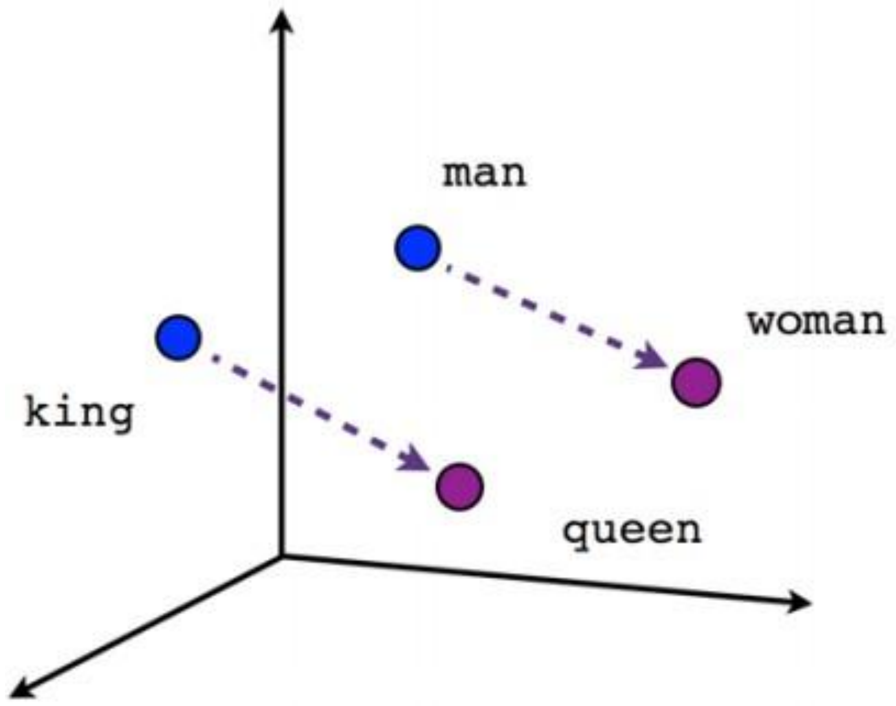
Настолько хорошо, что если из нулей и единиц составить числа, то он сможет за секунду обработать больше чисел, чем вы за всю свою жизнь.

Какой отсюда следует вывод?

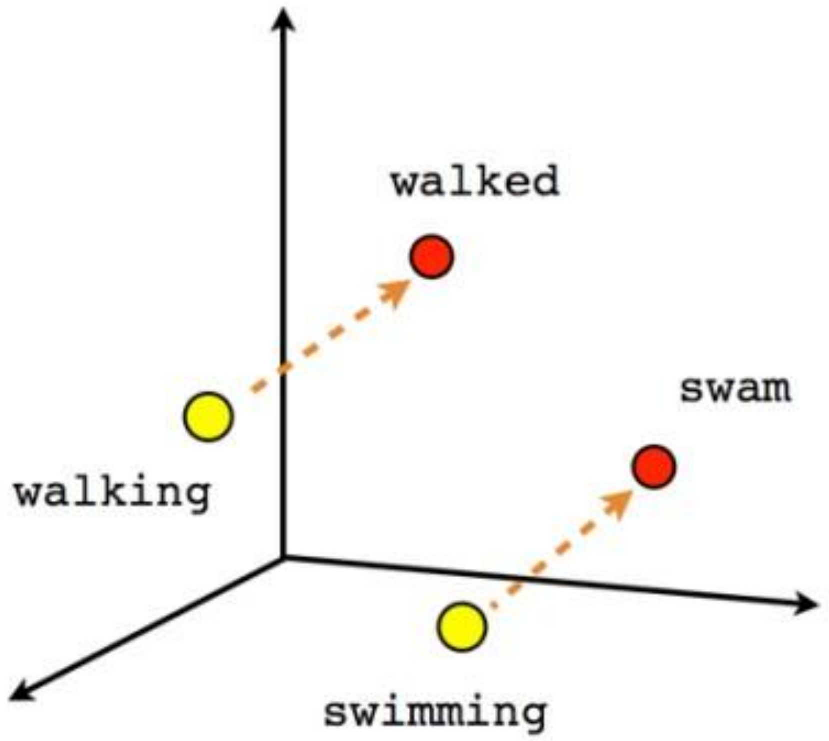
Фото, видео, звуки вашего голоса, стихи Маяковского — всё это нужно выразить на языке чисел.

И возможным это делает раздел математики под названием “линейная алгебра”.

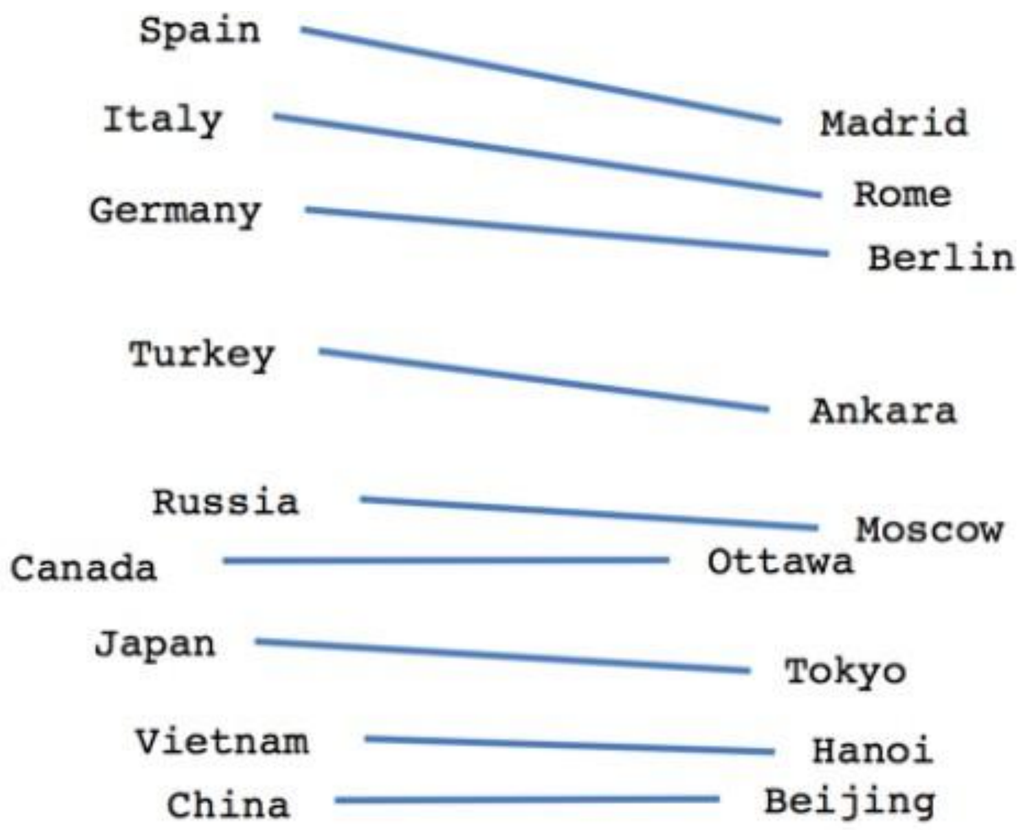
Линейная алгебра сделала возможной арифметику слов и…



Male-Female

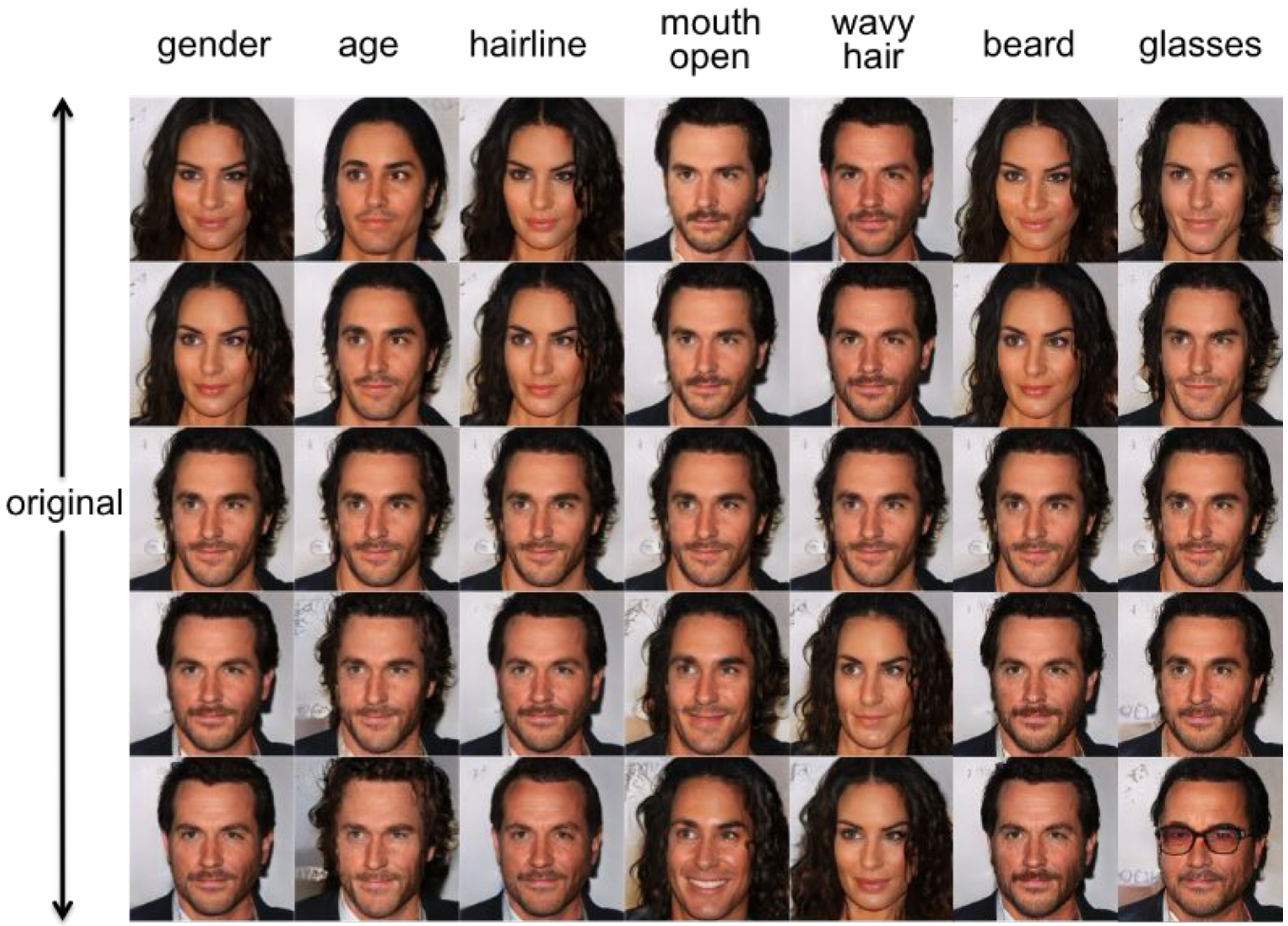


Verb tense



Country-Capital

…генерацию лиц с заданными свойствами, и многое другое



Линейная алгебра это язык, на котором задачу понимает машина.

Наша сегодняшняя задача — стать переводчиками с одного языка на другой.

Линейная алгебра

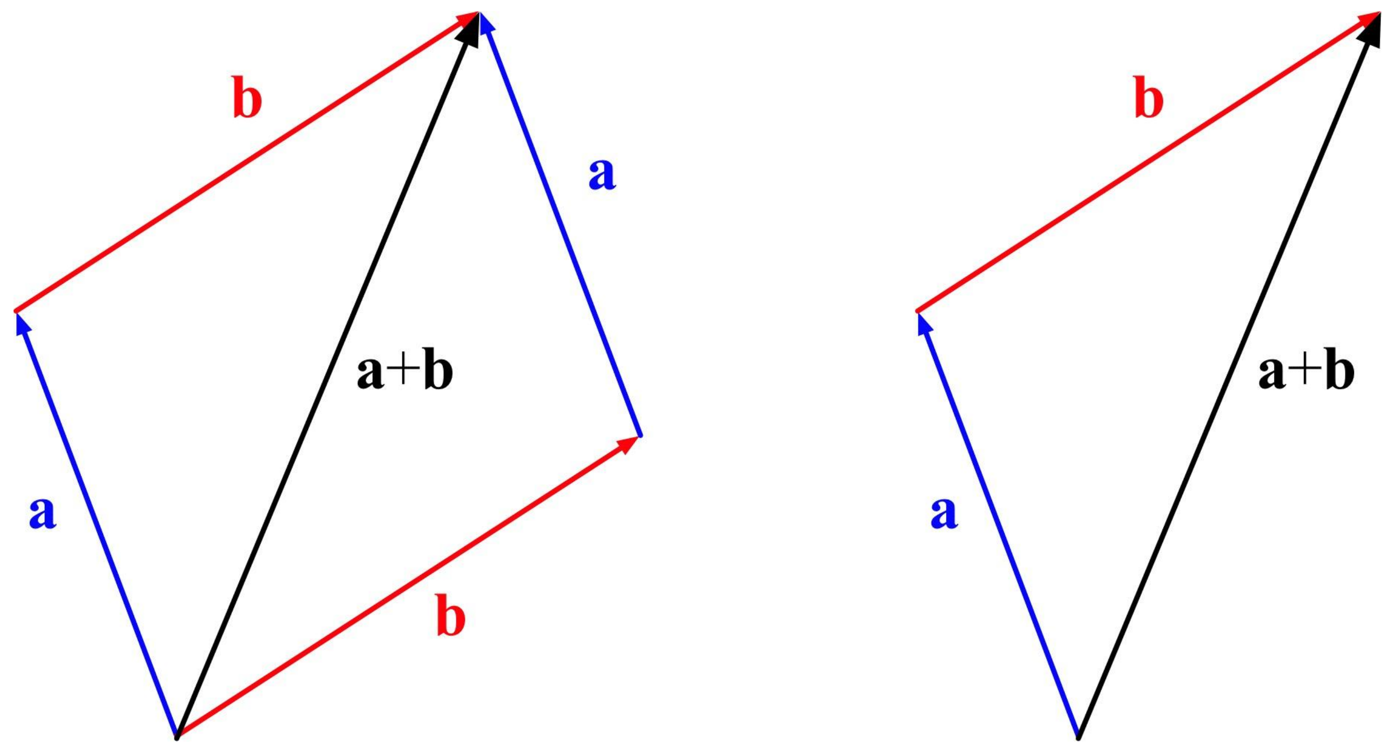
Повторение основных понятий

Линейное пространство

Основополагающим понятием линейной алгебры является понятие линейного пространства.

Пример: множество вещественных векторов из n компонент (например (0, 0, 0, …, 0))

Что с ними можно делать? Складывать (в любом порядке, в любой последовательности), умножать на вещественные числа.



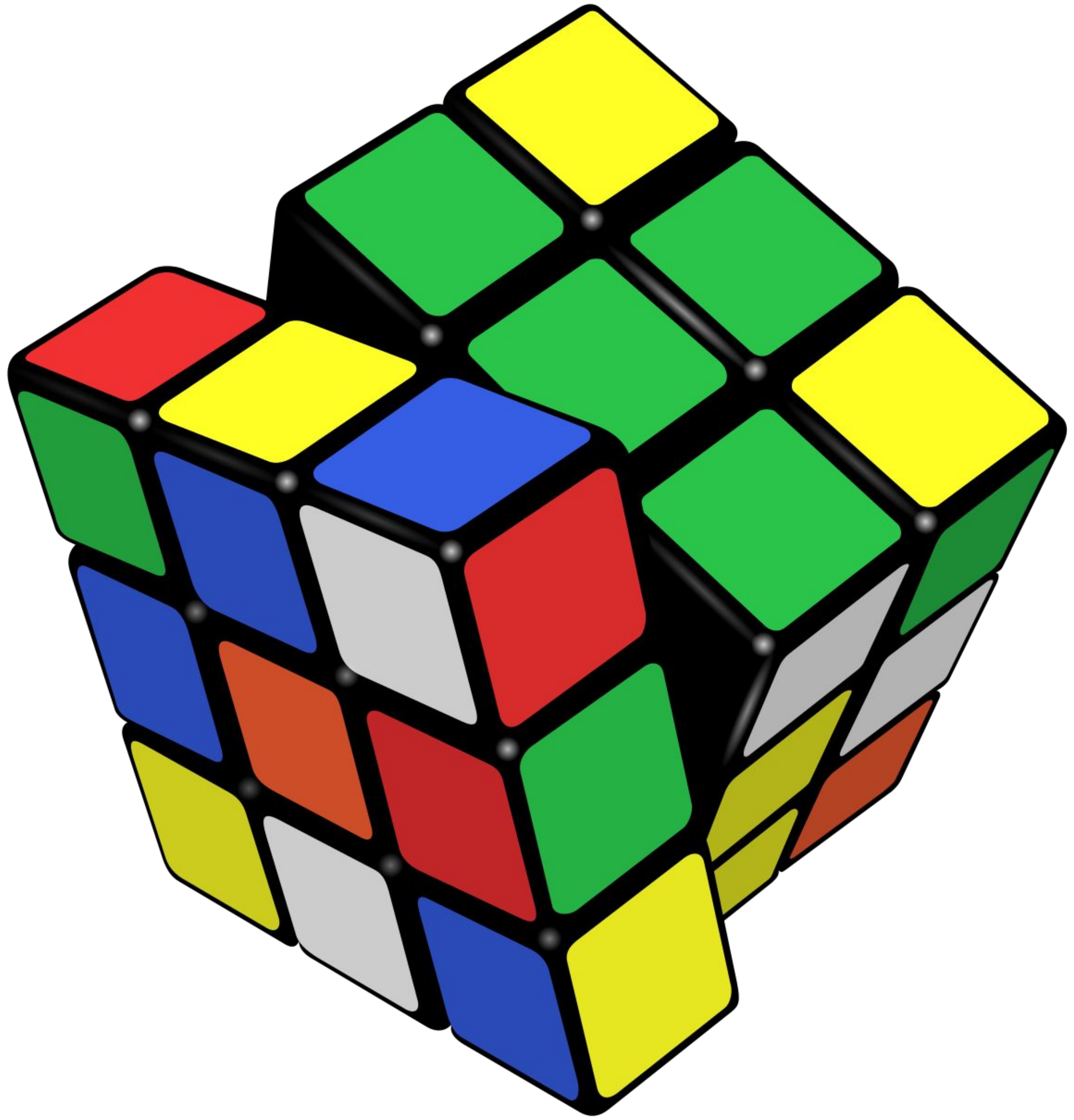
Линейное пространство

По жизни вам придётся иметь дело преимущественно с вещественными векторами, матрицами и тензорами.

Тем не менее, линейные пространства на этом не исчерпываются.

В общем случае, линейное пространство состоит из абелевой группы векторов (с операцией сложения), поля скаляров и операции умножения вектора на скаляр, согласованной со сложением векторов при помощи аксиом дистрибутивности.

Эти условия легко проверить.

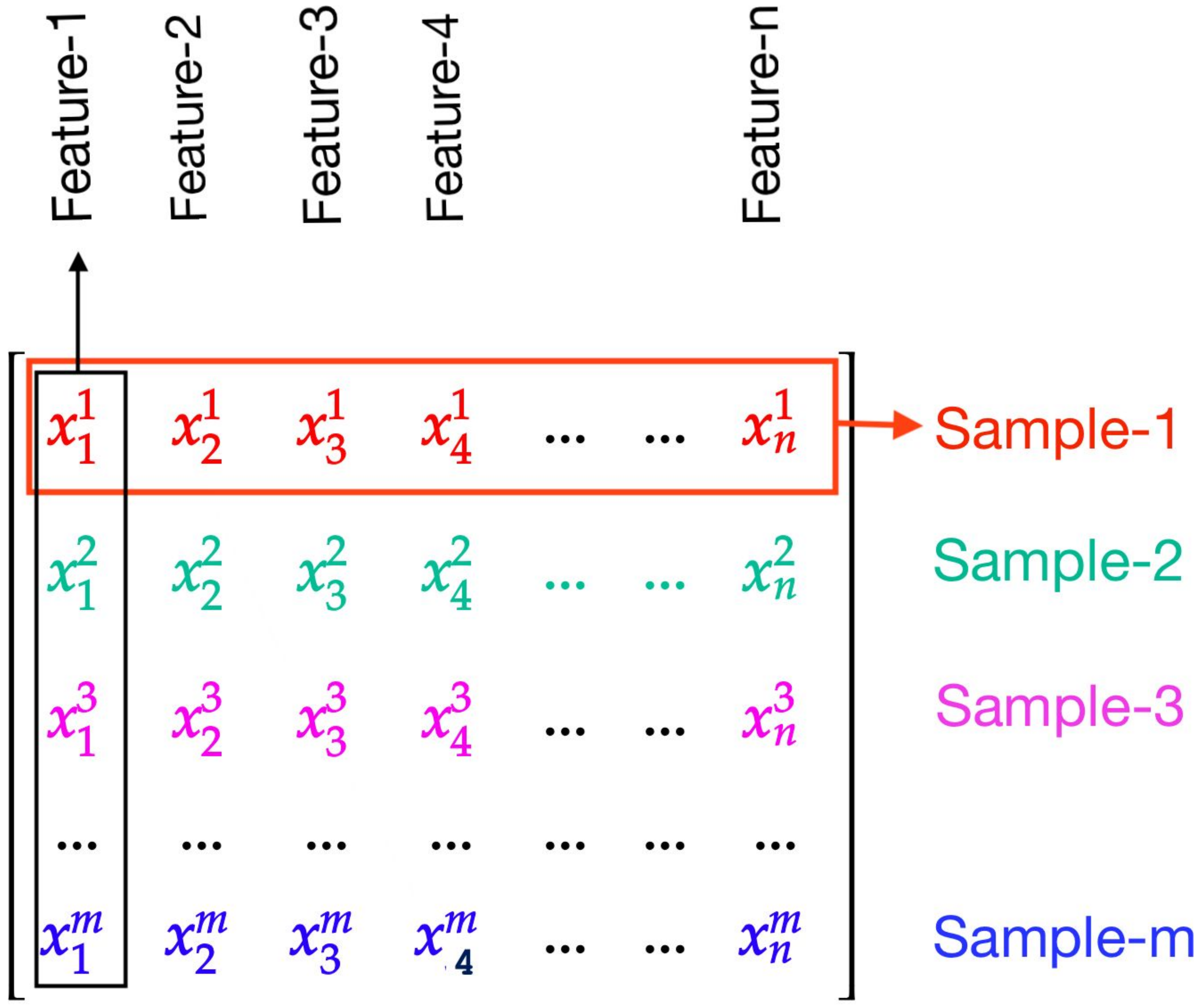


Примеры полезных линейных пространств

- **Пространство матриц одинаковой размерности.**
- **Пространство тензоров (“многомерных” матриц) одинаковой размерности.**
- **Пространство многочленов (обычных или тригонометрических).**
- **Пространство решений однородной системы линейных уравнений, алгебраических или дифференциальных.**
- **Пространство двоичных векторов с операцией сложения по модулю 2 и операцией умножения на 0 и 1.**

На этом занятии мы сосредоточимся на вещественных векторах, матрицах и связи между ними.

Поскольку именно с ними в большинстве своем работает Data Scientist



Важные понятия линейной алгебры

Определение: *(Матрица) Представление чисел в виде таблицы.*

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad \begin{pmatrix} 1 & 2 & -1 \\ 3 & 4 & 0 \end{pmatrix} \qquad \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

Определение: *(Транспонированная матрица) Зеркально отображенная относительно диагонали матрица.*

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \qquad \begin{pmatrix} 1 & 2 & -1 \\ 3 & 4 & 0 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 \\ 2 & 4 \\ -1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

Важные понятия линейной алгебры

<u>Определение:</u> *(Единичная матрица) Квадратная матрица, у которой на диагонали стоят единицы, а на всех остальных местах нули. (обычно обозначают I или E)*

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

<u>Определение:</u> *(Диагональная матрица) Матрица, у которой на диагонали стоят любые числа, а на всех остальных местах нули.*

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 5 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 10 \end{pmatrix} \qquad \begin{pmatrix} -7 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 70 \end{pmatrix}$$

Важные понятия линейной алгебры

Умножение матрицы на вектор:

$$\begin{pmatrix} 1 & 5 \\ 3 & 8 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}$$

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

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

Умножение матрицы на матрицу:

$$\begin{pmatrix} -1 & 3 \\ 4 & 0 \end{pmatrix} \cdot \begin{pmatrix} -1 & 4 \\ 3 & 0 \end{pmatrix} = \begin{pmatrix} 10 & -4 \\ -4 & 16 \end{pmatrix}$$

$$\begin{pmatrix} -1 & 3 & 1 \\ 4 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} =$$

Важные понятия линейной алгебры

<u>Определение:</u> *(Обратная матрица) Такая матрица, при умножение на которую, исходная матрица станет единичной. (обозначают верхним индексом -1)*

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

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad A^{-1} = \begin{pmatrix} -2 & 1 \\ 3/2 & -1/2 \end{pmatrix}$$

$$A^{-1} \cdot A = A \cdot A^{-1} =$$

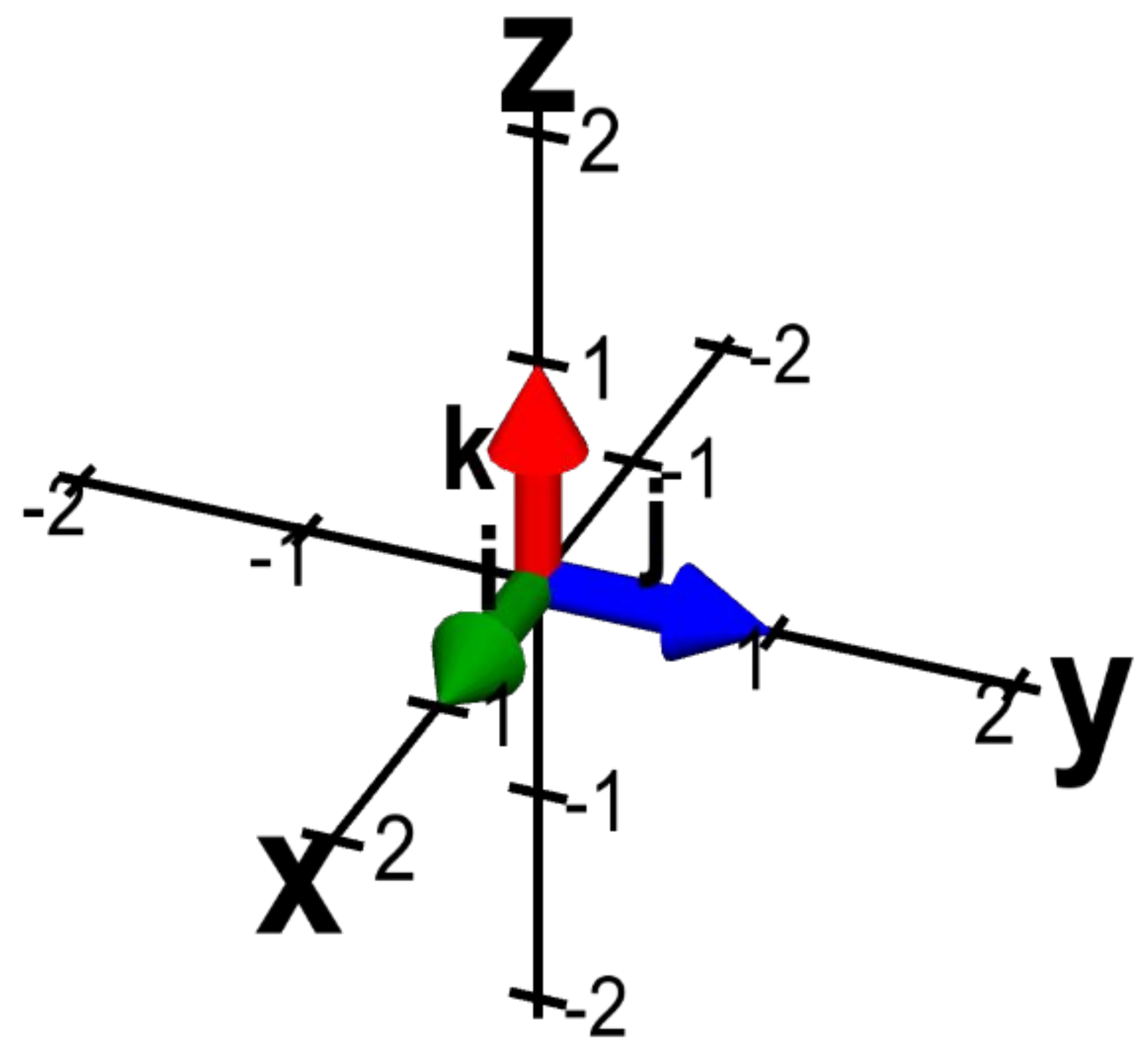
В общем случае, находить обратную матрицу очень сложно и это на практике часто делают приближенно.

Базис линейного пространства

Во всяком линейном пространстве есть несколько главных векторов,
— так называемый базис. Остальные вектора выражаются через базисные.

Количество базисных векторов определяет размерность пространства.

Пример: В случае R^n базисных векторов ровно n. Они задают координатные оси.



Базис линейного пространства

Базис — такое минимальное по включению множество векторов, что:

-Ни один из них не выражается линейной комбинацией остальных;

-Каждый элемент пространства можно единственным образом представить линейной комбинацией конечного набора векторов из этого множества.

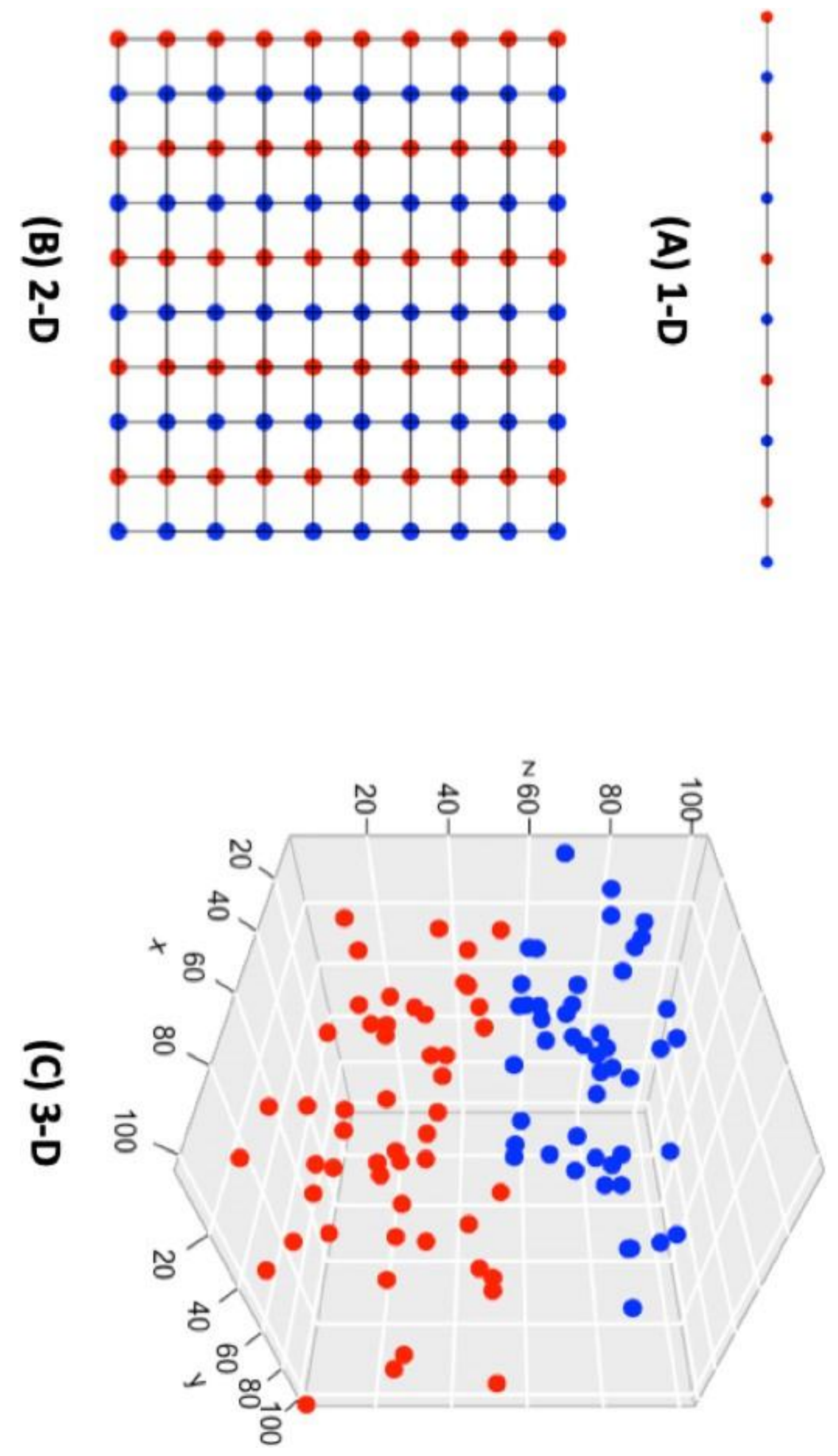
Доказать, что всякое линейное пространство имеет базис, можно при помощи леммы Цорна.

Размерность линейного пространства

Размерность — одна из главных характеристик линейного пространства.

Размерность в точности совпадает с числом базисных векторов.

В машинном обучении есть даже одноимённое [проклятие](#):)



Важные понятия линейной алгебры

<u>Определение:</u> *(детерминант/определитель) Численная характеристика матрицы, которая которая в некотором смысле описывает сжатие или растяжение пространства при преобразовании этой матрицей.*

$$det \begin{pmatrix} -1 & -1 \\ 1 & 3 \end{pmatrix} = (-1) \cdot 3 - (-1) \cdot 1 = -2$$

Если связать получившееся значение с геометрическим смыслом, то если рассмотреть то, что мы матрицей преобразуем единичный квадрат, то:

- Знак минус будет говорить, что наш квадрат будет иметь противоположную ориентацию
- Значение 2 будет говорить о том, что после преобразования наш квадрат будет иметь площадь, в два раза превосходящую площадь оригинального квадрата.

Мы опустим алгоритм подсчета определителя для матриц большего порядка.

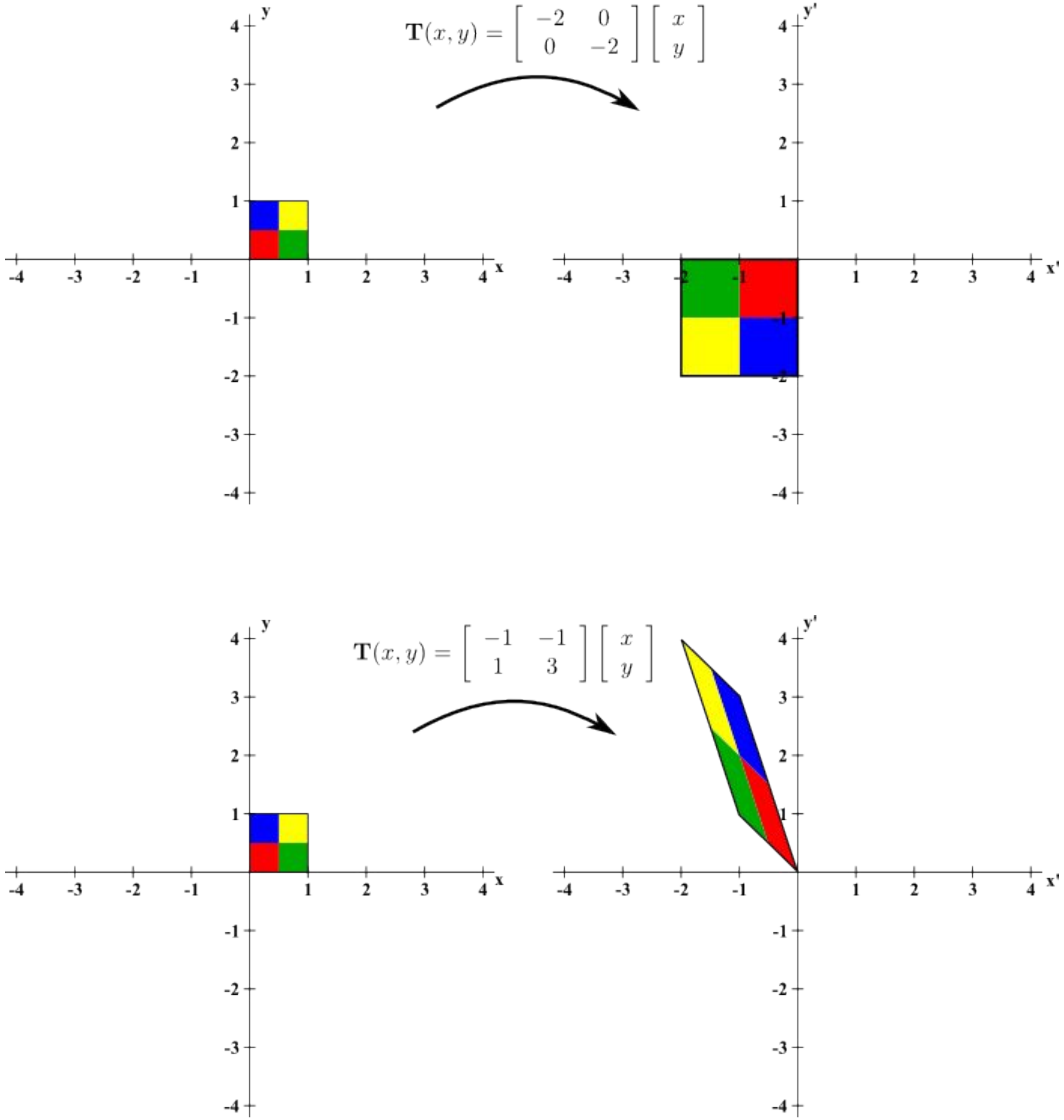
Что касается матриц:

Каждая матрица кодирует линейное преобразование линейных пространств в некоторой системе координат: одно пространство отображается на подпространство другого.

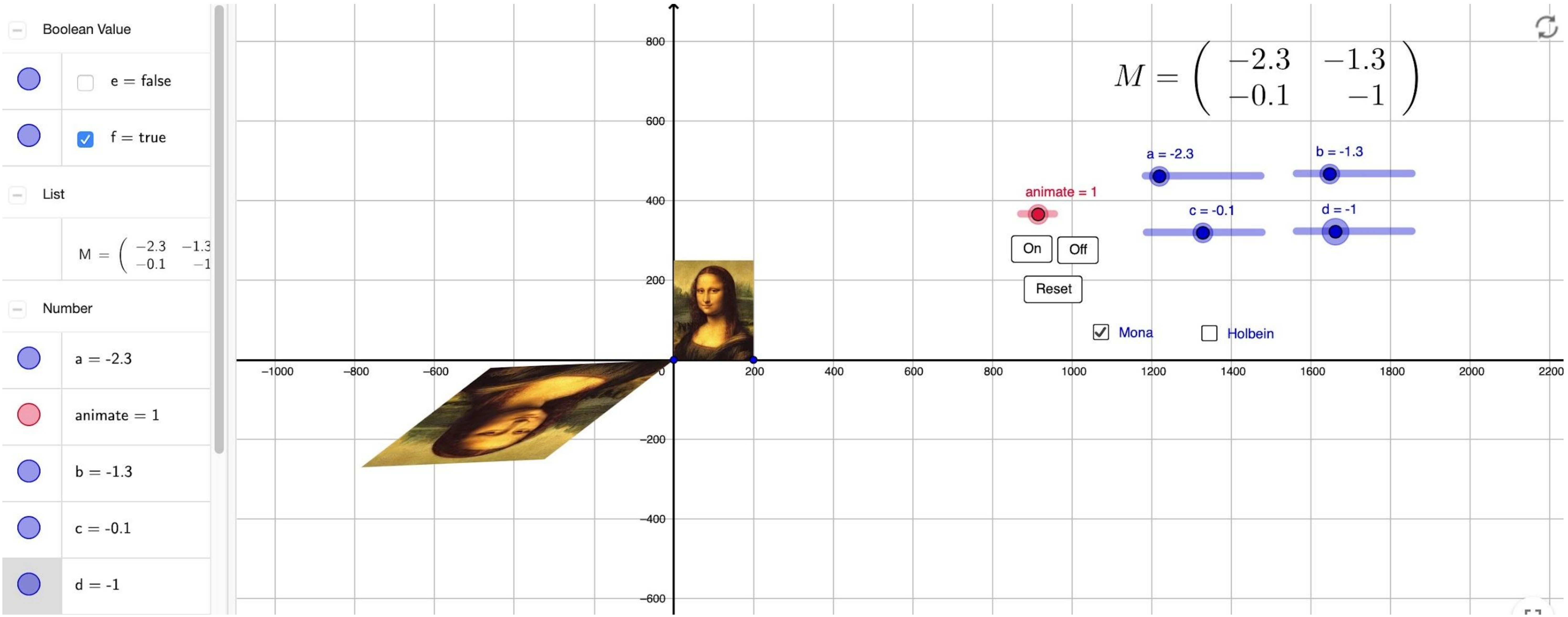
Все эти преобразования можно разложить на комбинацию проекций, вращений, отражений и масштабирования вдоль координатных осей. Других преобразований нет.

Одному преобразованию можно сопоставить бесконечное количество матриц! Столько же, сколько есть систем координат. Сама по себе матрица это просто таблица чисел!

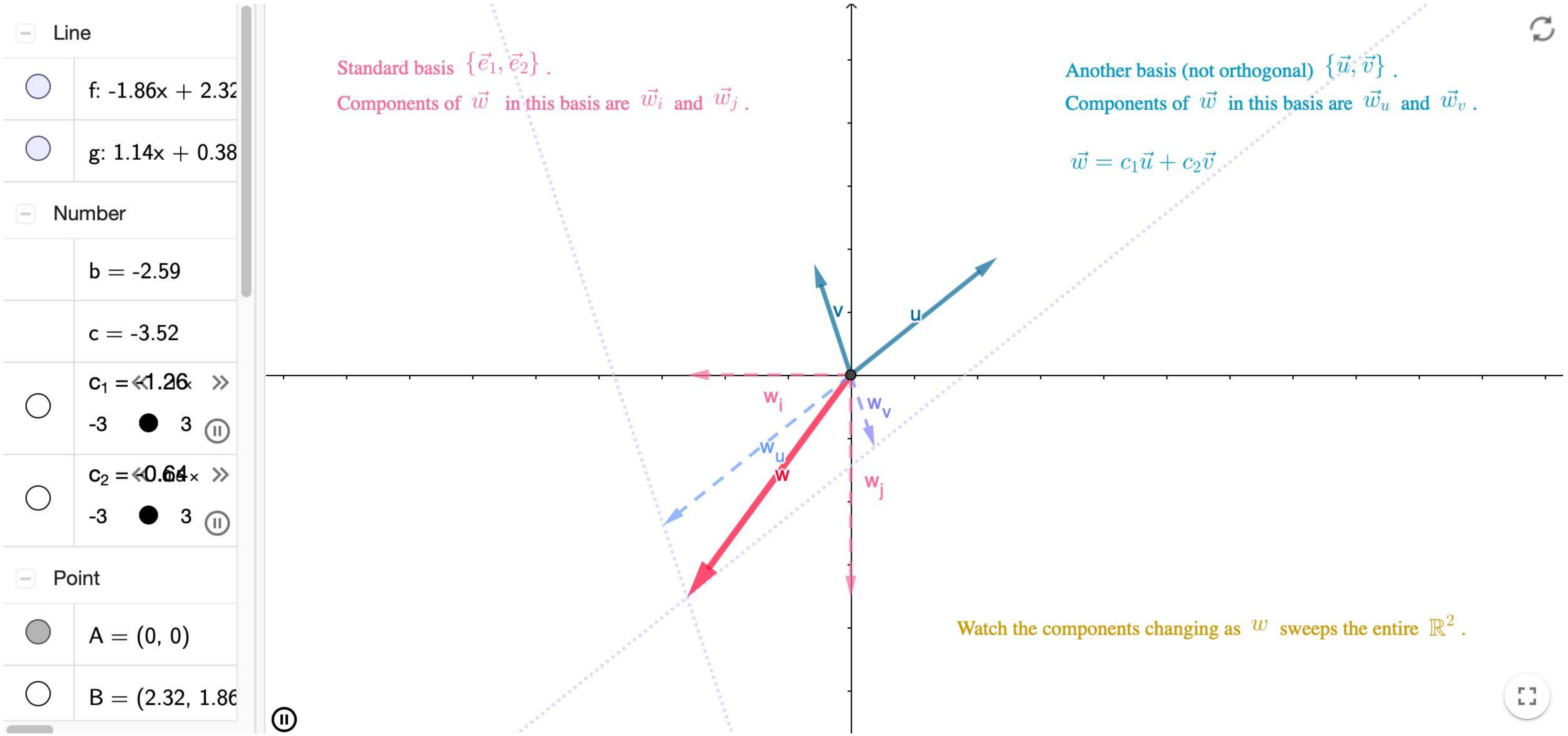
$$\begin{pmatrix} -1 & -1 \\ 1 & 3 \end{pmatrix} \cdot \begin{pmatrix} \ \end{pmatrix} =$$



"Потрогать" это определение можно на GeoGebra



Наблюдать за тем, как замена базиса влияет на координаты векторов, тоже можно на [GeoGebra]



Собственные направления

У линейного преобразования $A : V \to V$ есть некоторое количество собственных направлений.

Вдоль них оно не вращает, не отражает, а масштабирует.

Это пример т.н. инвариантных подпространств: они не изменяются под действием A.

Формулами это задаётся так:

$$Av = \lambda v$$

- λ – собственное значение
- v – собственный вектор

Например:

$$A = \begin{pmatrix} 1 & 2 \\ -1 & 4 \end{pmatrix}$$

$$v = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

Понятие собственных направлений очень полезно, поскольку позволяет находить “хорошие” направления нашего преобразования и, кроме того, если собственных векторов окажется столько, сколько базисных, то мы сможем “перейти” в базис собственных векторов, в котором наша матрица A будет иметь диагональный вид, что упрощает многие выкладки.

У собственных направлений, например, может быть следующий физический смысл: В теории колебаний собственные числа это частоты колебаний системы при свободном движении, а собственные вектора это их «траектории».

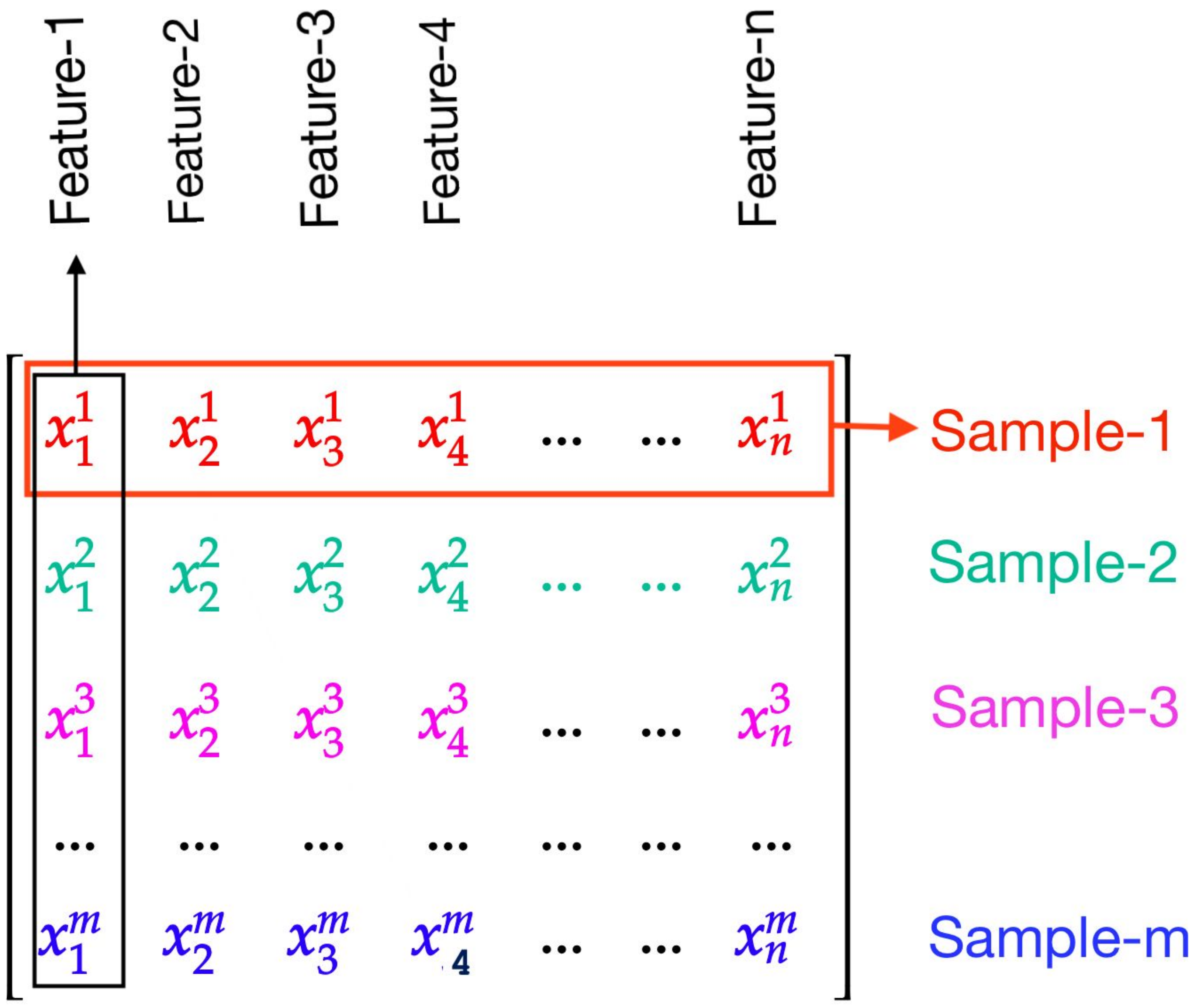
Матричные разложения

SVD и его связь с **PCA** (метод главных векторов)

Матрицы признаков прямоугольные вещественные матрицы размера *m x n*, **где:**

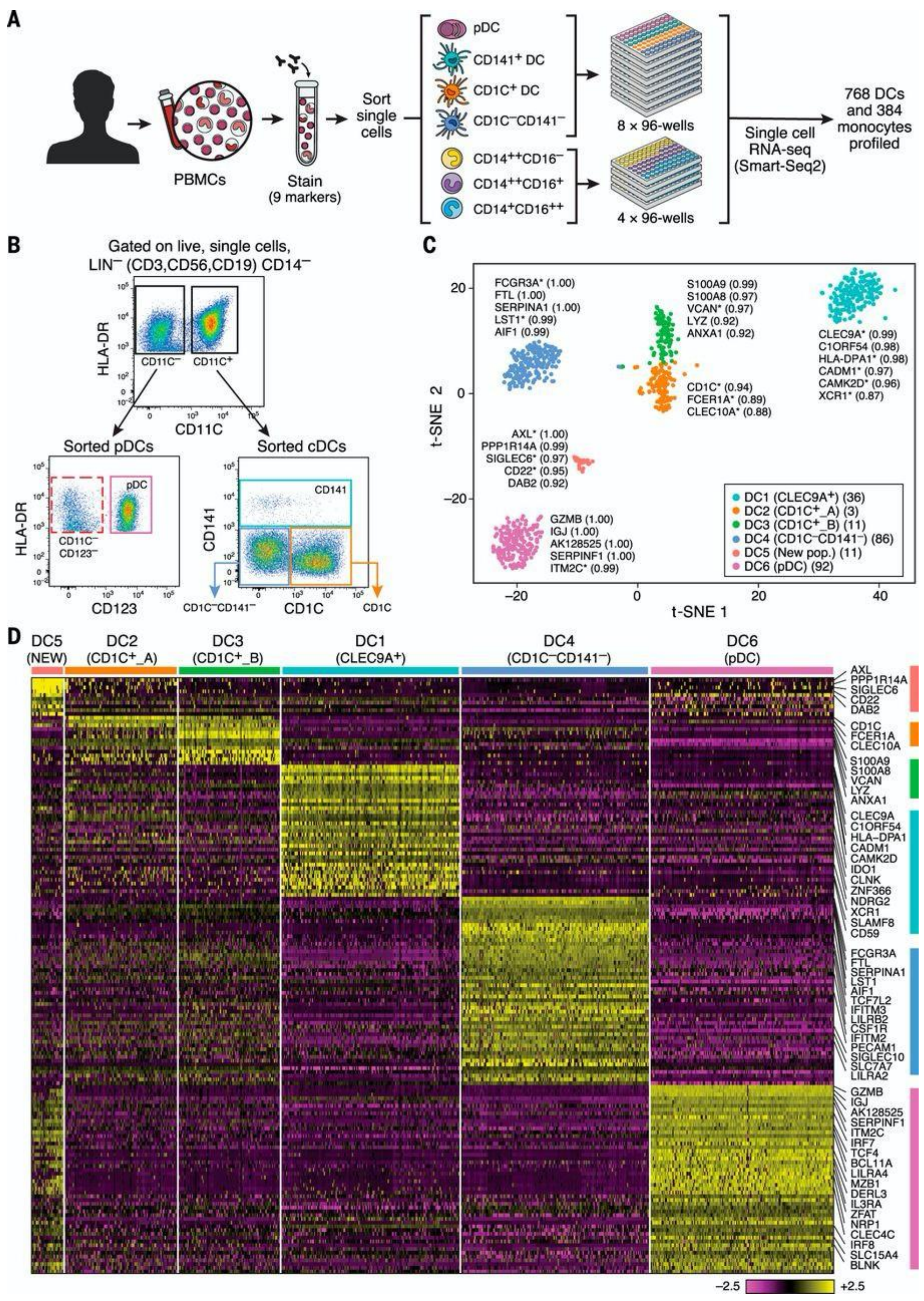
m – размер выборки

n – количество признаков (и размер признакового пространства)



Линейная алгебра — инструмент, который позволяет вам работать с данными **огромной размерности**.

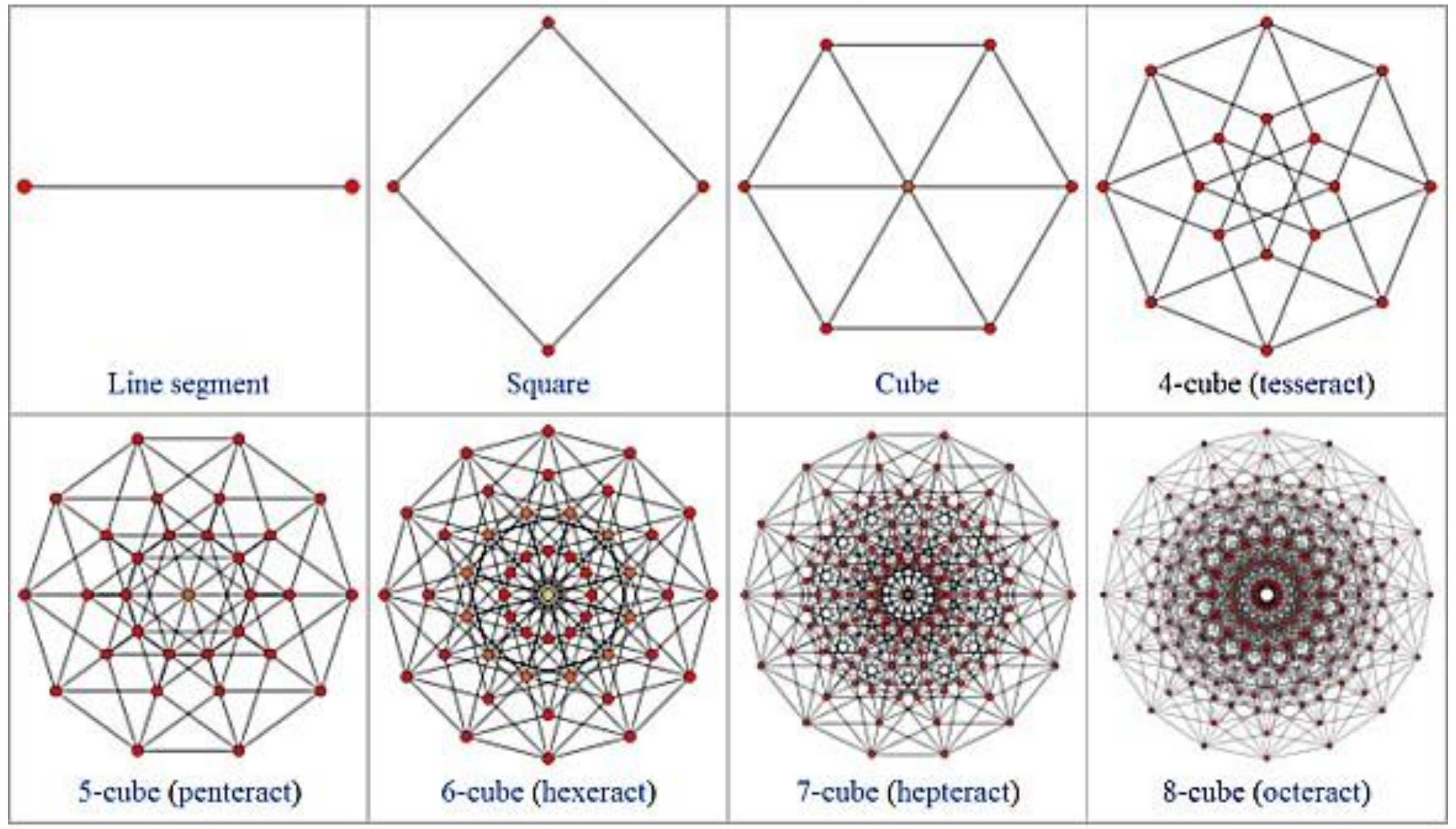
Пример: типичная клетка опухоли в данных секвенирования это 14000 чисел, а образец обычно содержит от ста до миллиона клеток.



Тем не менее, наш мозг не в состоянии воспринимать данные высокой размерности.

Всё, что выше размерности 7 (длина, ширина, высота, три цвета, время) — за гранью нашего восприятия.

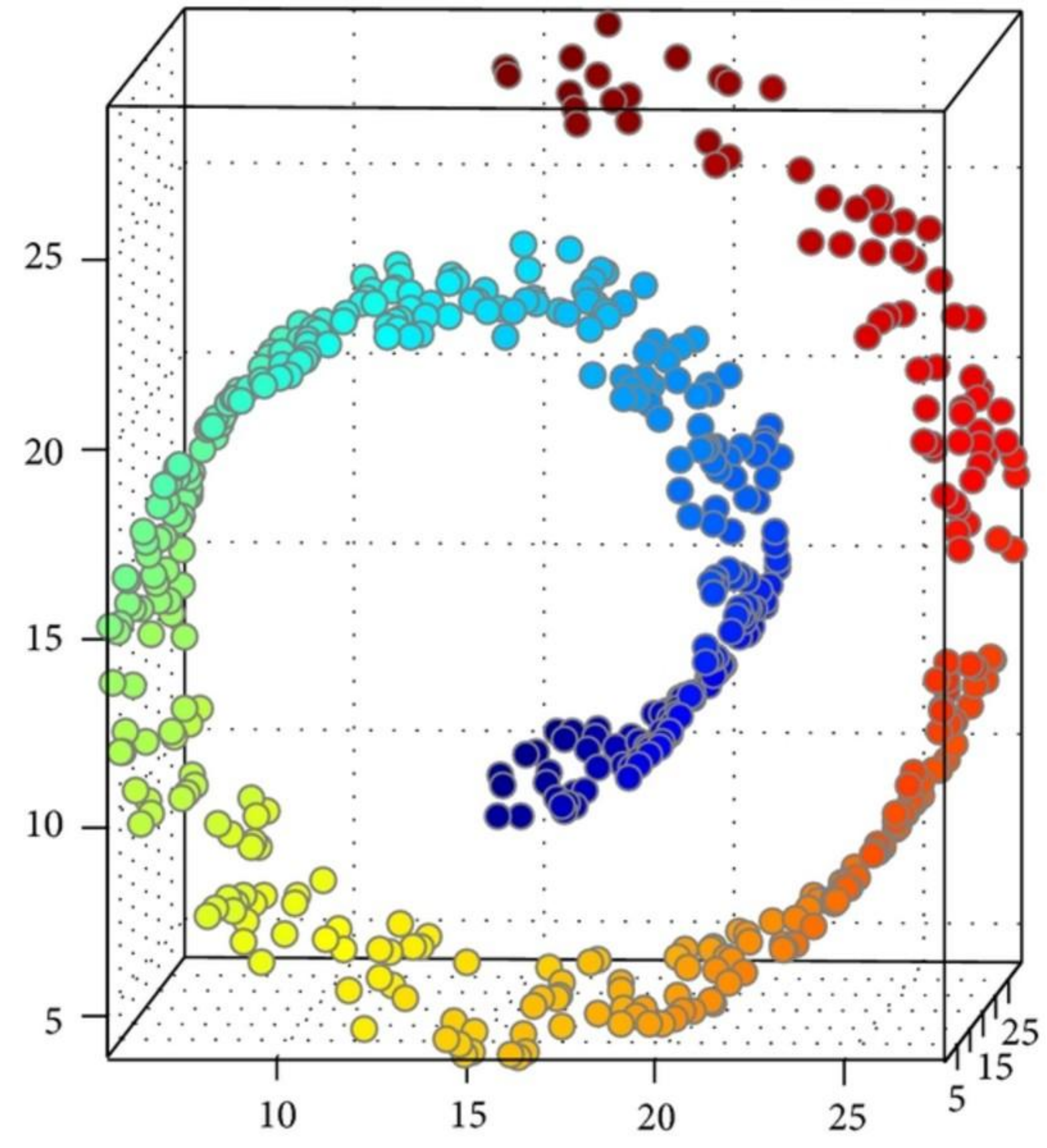
Даже простейшие геометрические объекты в духе куба становится практически невозможно изобразить без потери информации.



Укладка многомерных кубов (гиперкубов) на плоскости

К счастью, реальные данные обычно лежат на т.н. маломерных многообразиях.

Они **не** распределены по многомерному пространству равномерно, а лежат на поверхностях существенно меньшей размерности.

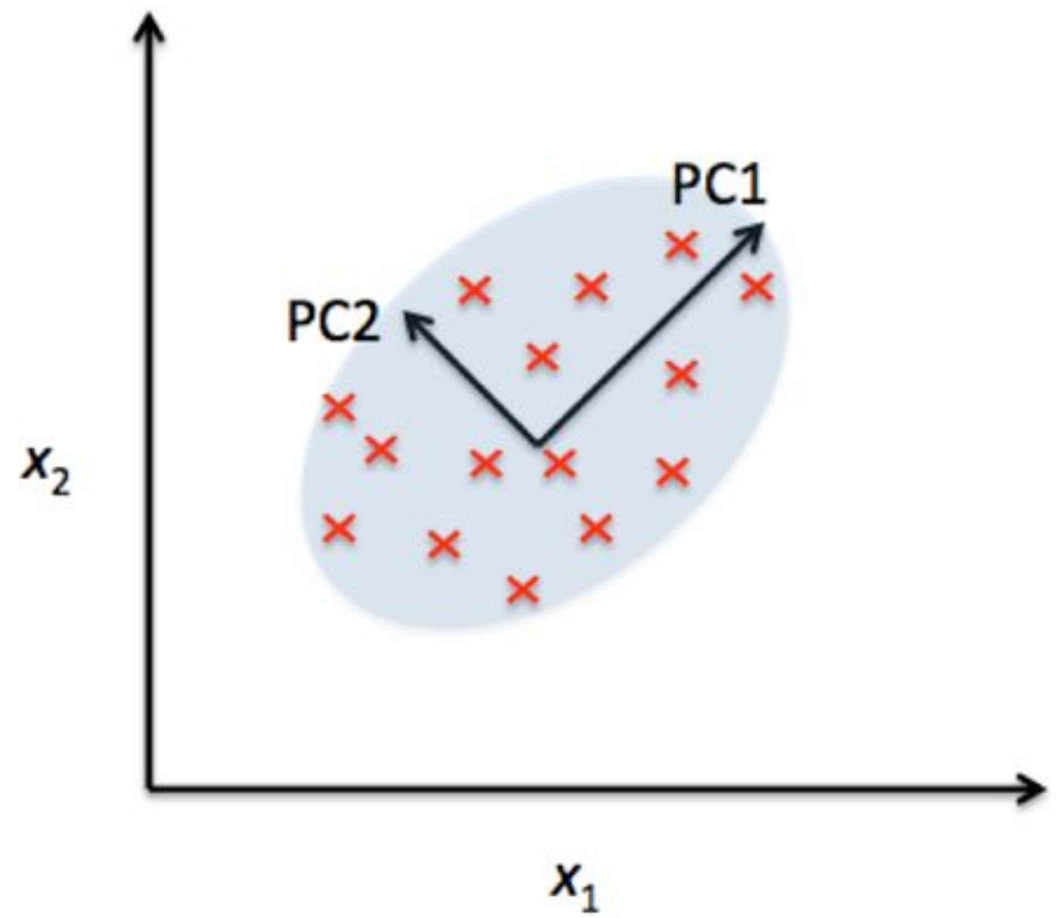


Формально это трёхмерные данные, но по факту они лежат на двумерной полосе

Определением тонкой геометрической структуры данных занимается раздел ML под названием manifold learning.

Но часто понизить размерность, скажем, с 14000 до 30-100 без существенной потери информации можно и при помощи более простых методов.

Например, при помощи PCA — метода главных компонент, который ищет оптимальное приближение размерности для вашей матрицы данных.



Метод главных компонент (PCA)

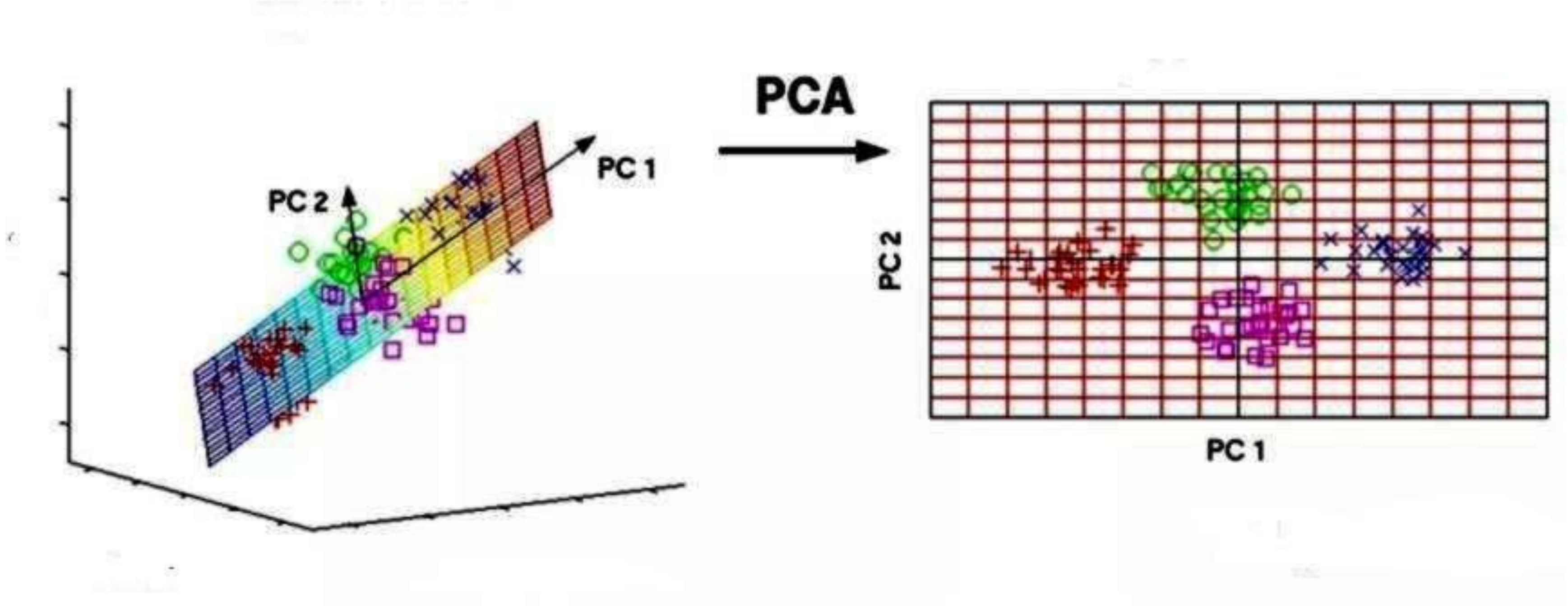
Алгоритм работы PCA можно описать следующими шагами:

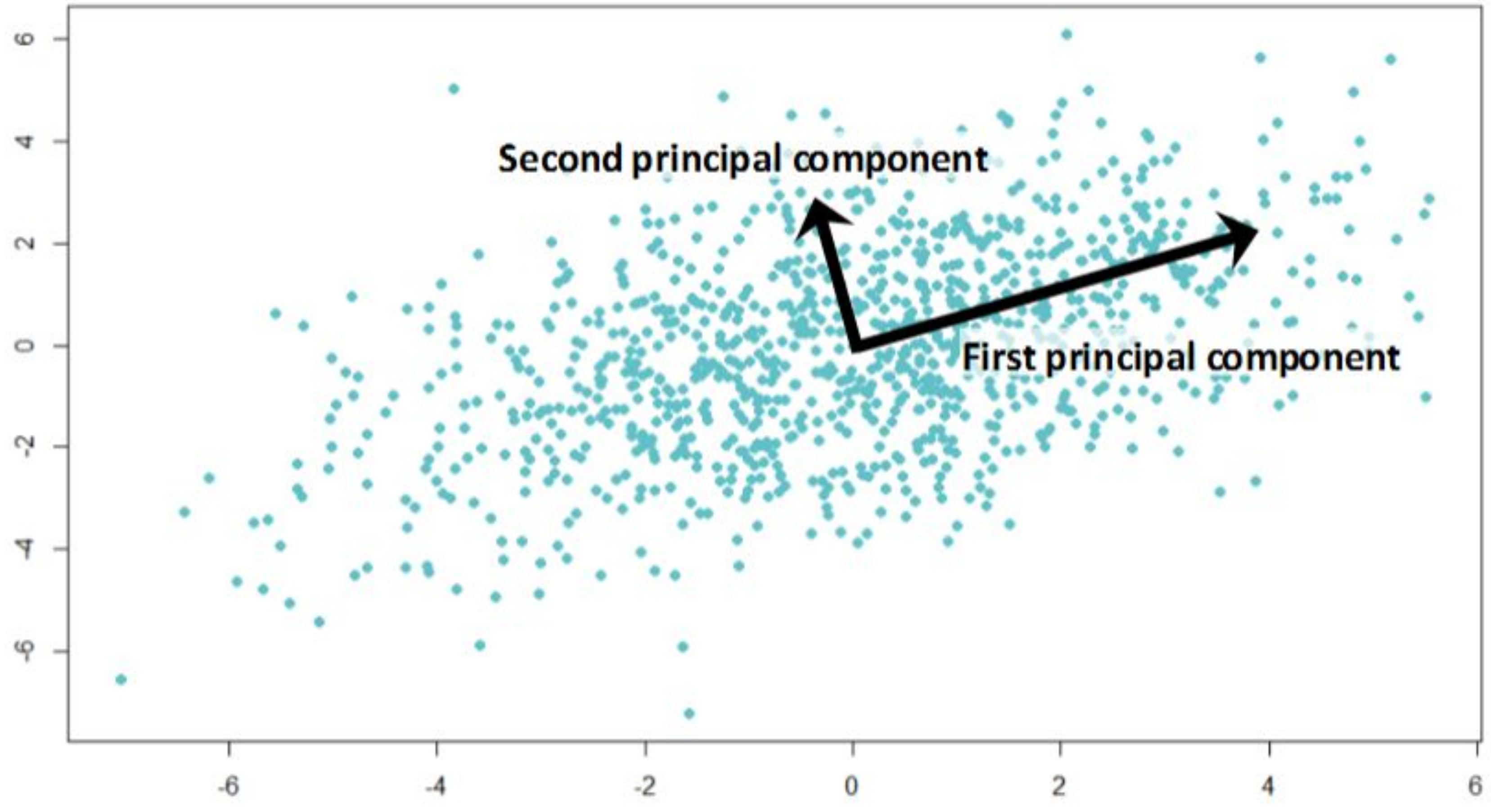
1. Найти такое направление в пространстве, вдоль которого дисперсия данных максимальна.
2. Среди оставшихся направлений, ортогональных предыдущим, найти направление вдоль которого дисперсия максимальна.
3. Повторять шаг 2 до тех пор, пока это возможно.

Получившиеся направления и называют главными компонентами.

Мы могли бы находить такие направления полным перебором, но сколько бы это заняло времени хотя бы в 10-мерном пространстве?)

Поэтому мы, как настоящие математики, воспользуемся доступным мат. аппаратом, чтобы решить эту задачу!





Важные понятия линейной алгебры

<u>Определение:</u> *(Ортогональная матрица) Матрица, для которой обратная матрица совпадает с транспонированной.*

$$A \cdot A^T = A \cdot A^{-1} = I$$

Например: $\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$

<u>Определение:</u> *(Сингулярные направления и значения)*

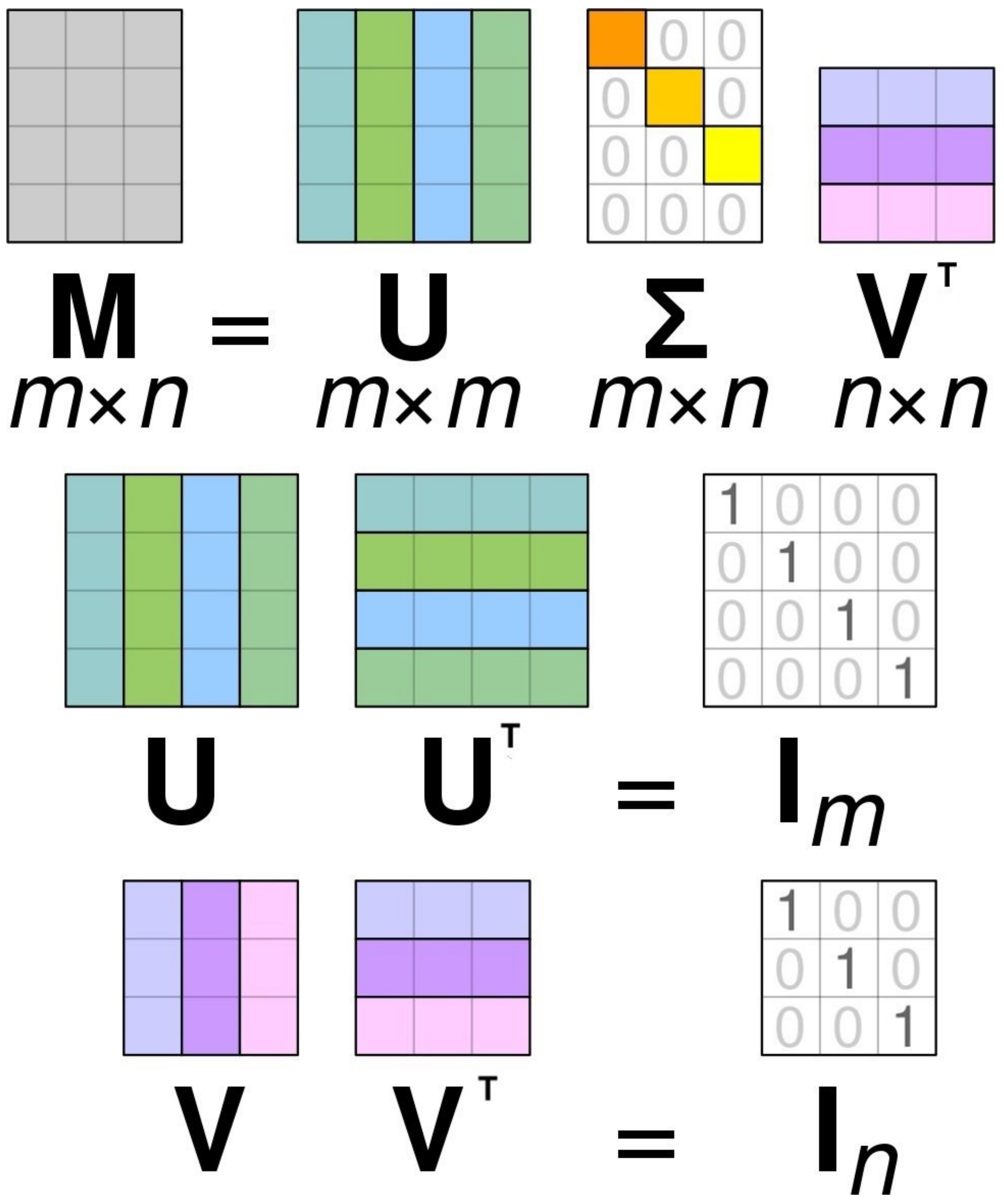
$$Mv = \sigma u$$
$$M^T u = \sigma v$$

здесь v – правый сингулярный вектор, u – левый сингулярный вектор, σ – сингулярное значение.
Сингулярное значение это корень из собственного значения матрицы $M^T M$ (что мы можем легко проверить прямо здесь)

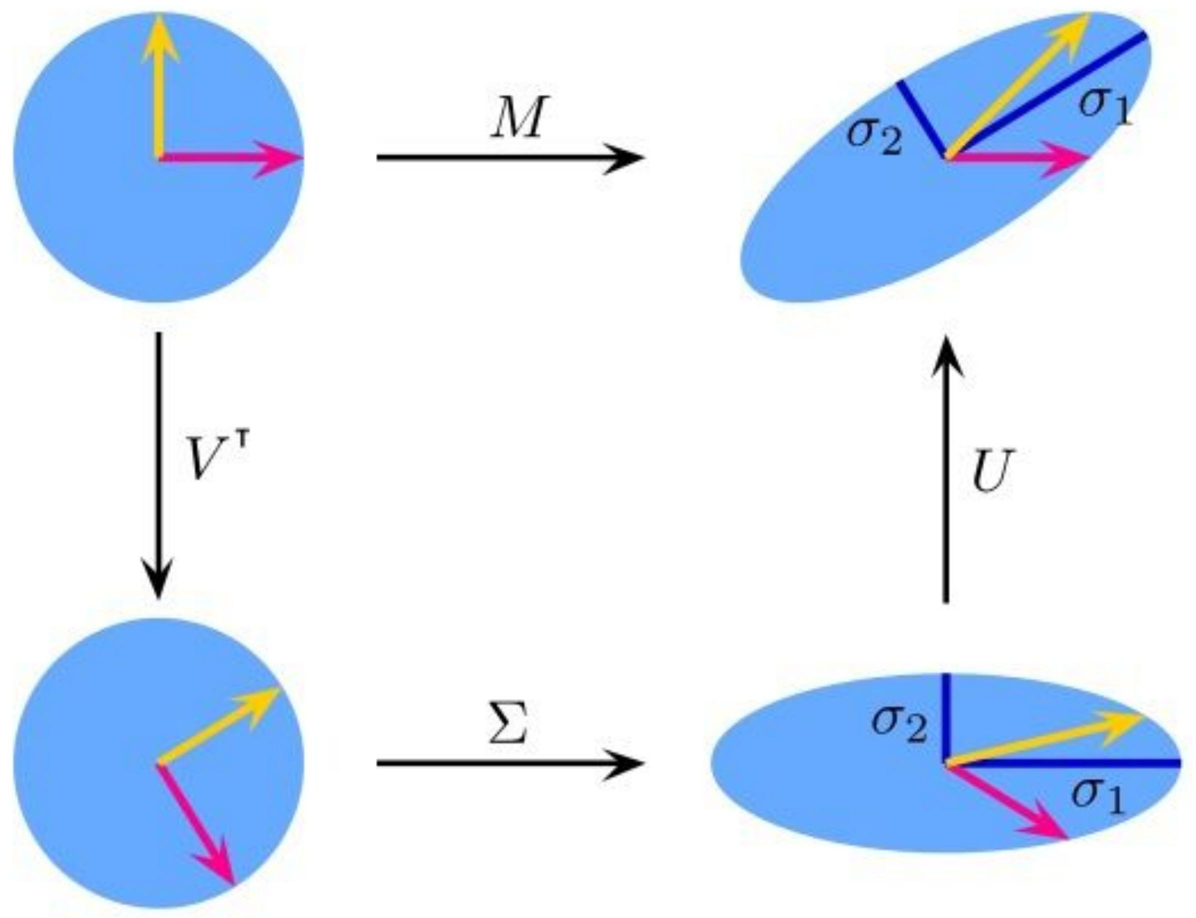
Сингулярные вектора и значения это в некотором смысле аналоги собственных векторов и чисел для неквадратных матриц. Причем если матрица квадратная, то квадрат сингулярного значения равен модулю соответствующего собственного значения (если такое существует).

https://ru.frwiki.wiki/wiki/D%C3%A9composition_en_valeurs_singuli%C3%A8res

Сингулярное (SVD) разложение матрицы

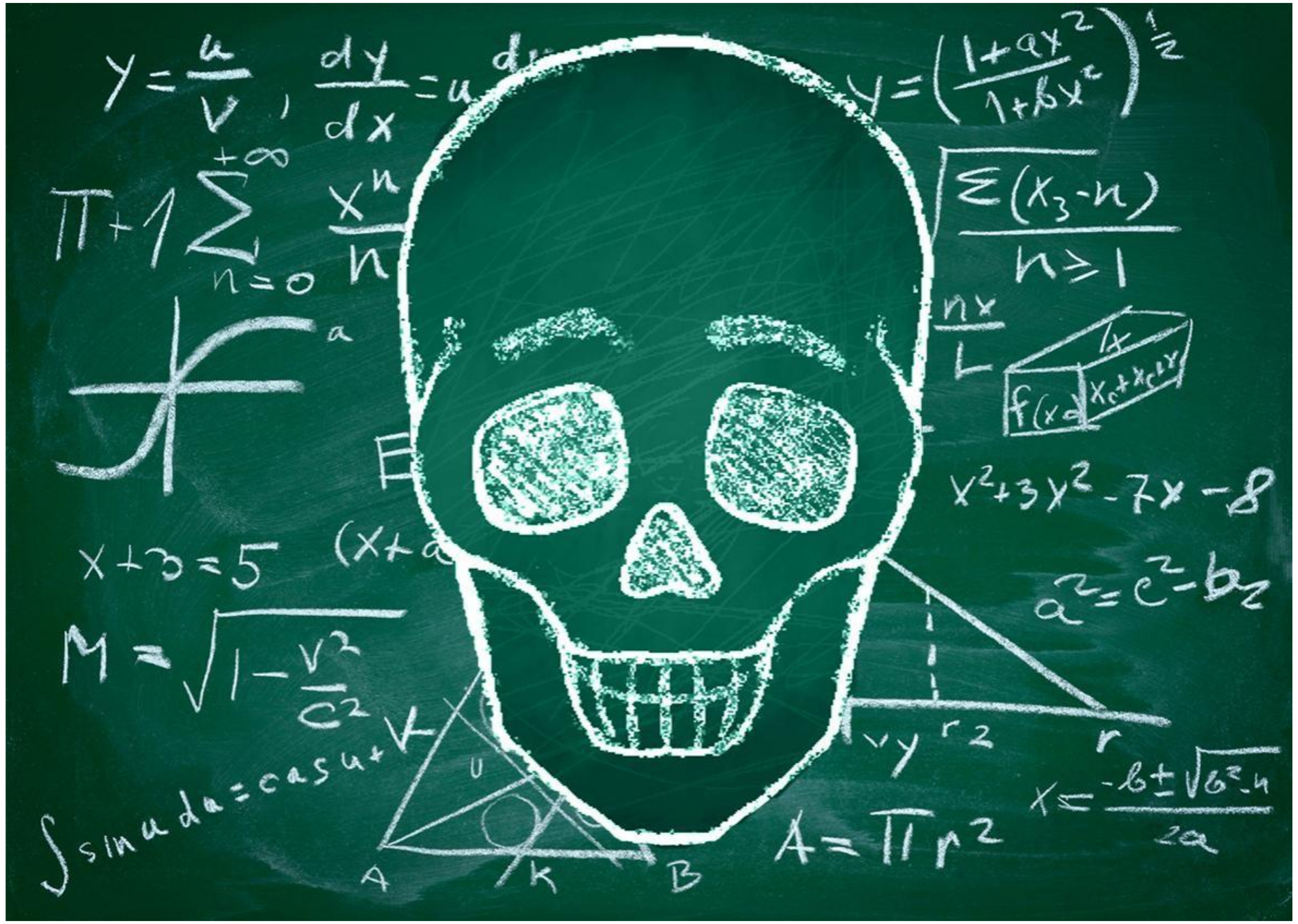


- $M \in \mathbb{R}^{m\times n}$ — Исходная матрица;
- $U \in \mathbb{R}^{m\times m}$ — Матрица левых сингулярных векторов;
- $\Sigma \in \mathbb{R}^{m\times n}$ — Матрица, на главной диагонали которой находятся т.н. сингулярные числа, (в порядке невозрастания);
- $V \in \mathbb{R}^{n\times n}$ — Матрица правых сингулярных векторов.



$$M = U \cdot \Sigma \cdot V^{\intercal}$$

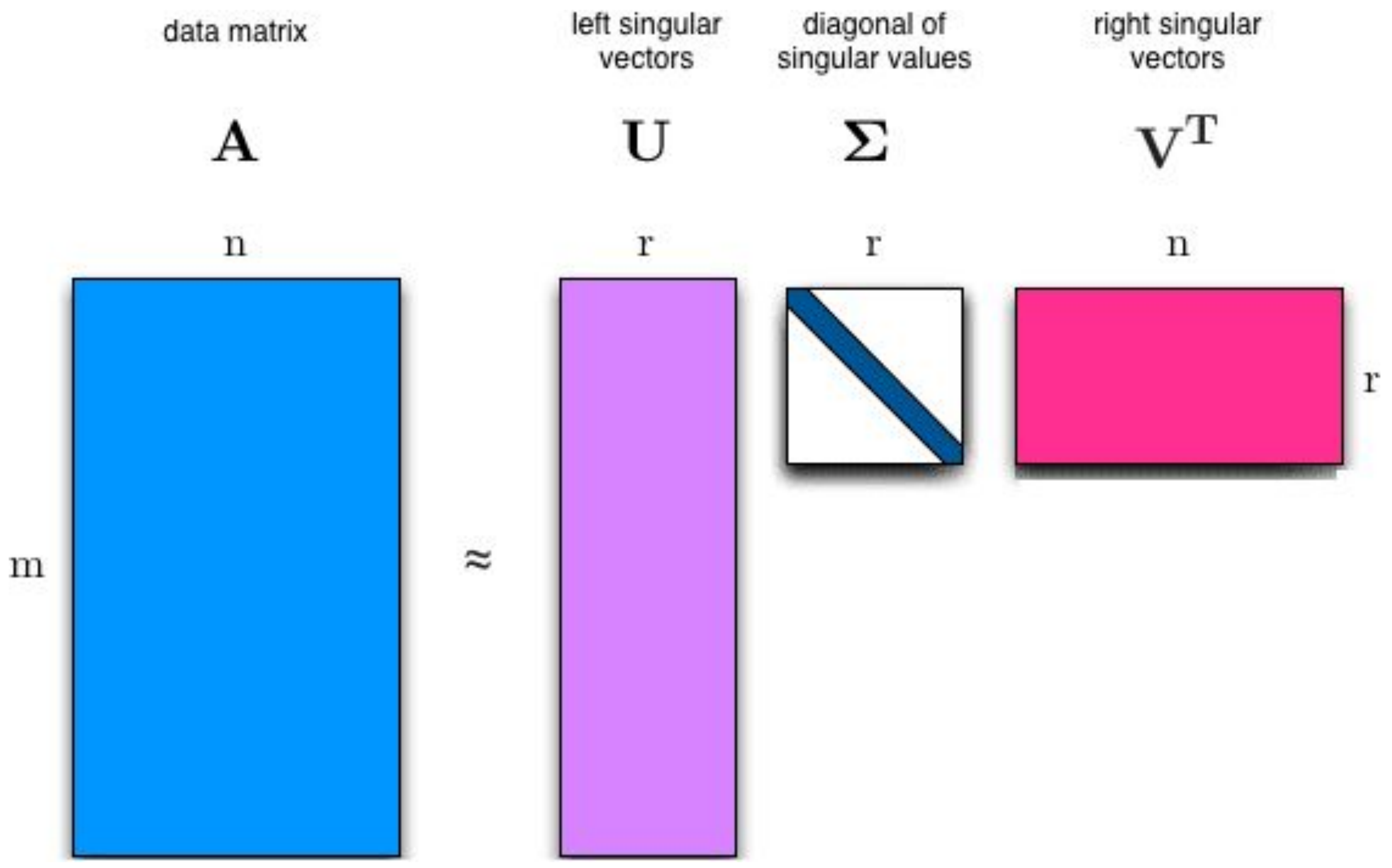
Выглядит жутковато…



Усечённое SVD-разложение ранга r

матрицы $M^{m \times n} \in \mathbb{R}$

— обычное SVD-разложение, где оставили только самых больших сингулярных значений вместе с соответствующими сингулярными векторами.



Можно доказать, что усечённое SVD-разложение ранга r это **оптимальная по Фробениусовой норме аппроксимация ранга r**.

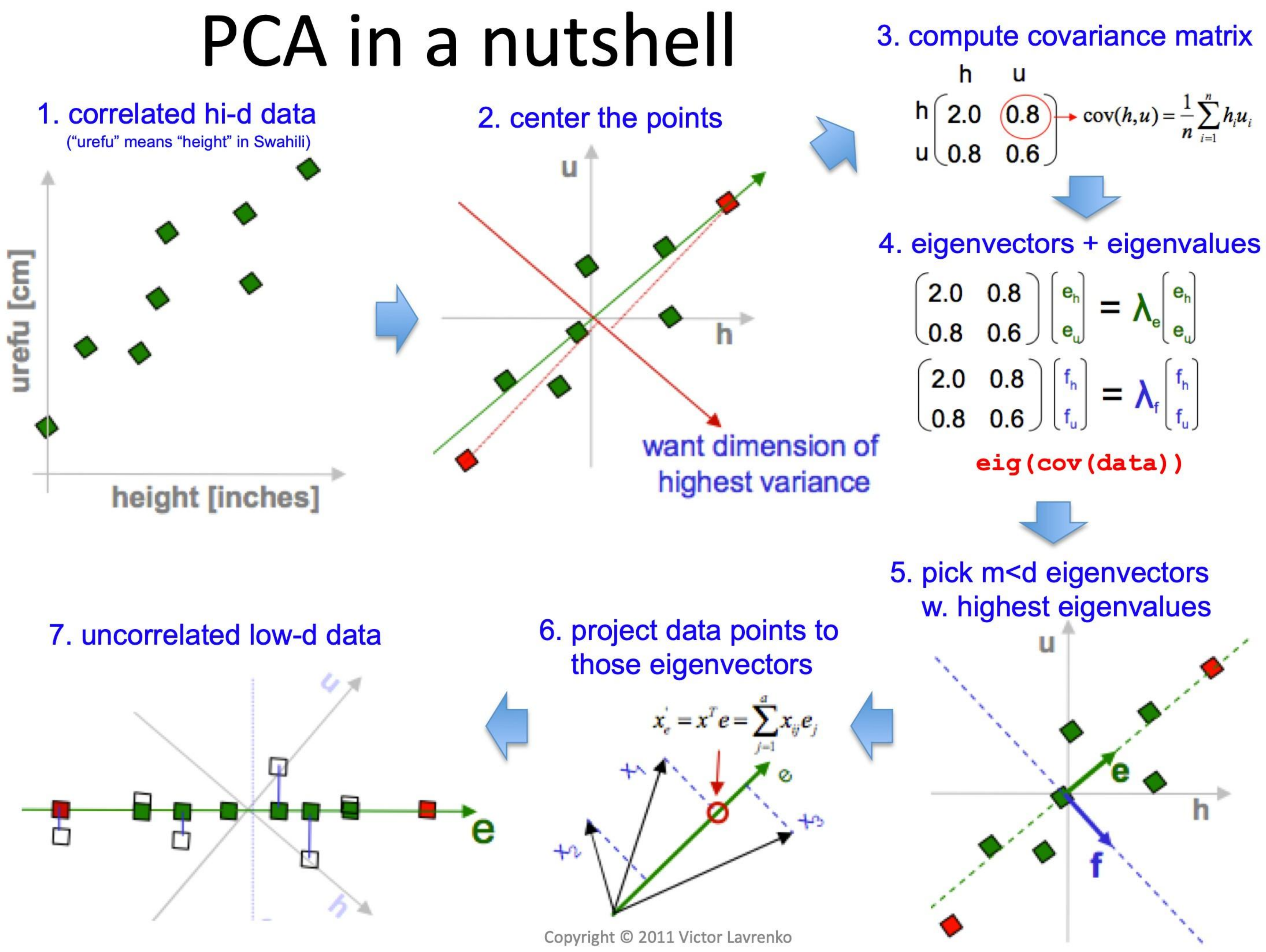
Поздравляю, благодаря этому факту мы можем сформулировать алгоритм метод главных компонент. :)

Норма Фробениуса, или **евклидова норма** (для евклидового пространства) представляет собой частный случай p-нормы для $p = 2$: $\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$.

Связь SVD и PCA

PCA преобразование можно было делать при помощи нахождения собственных векторов (как на алгоритме справа), но к сожалению почти всегда наши датасете имеют разное число строк и столбцов и поэтому мы вынуждены использовать **SVD**.

Но есть и хорошая новость! Для того, чтобы получить **PCA** преобразование, достаточно всего лишь посчитать $U\Sigma$, то есть *произведение первых двух матриц в разложении **SVD***.

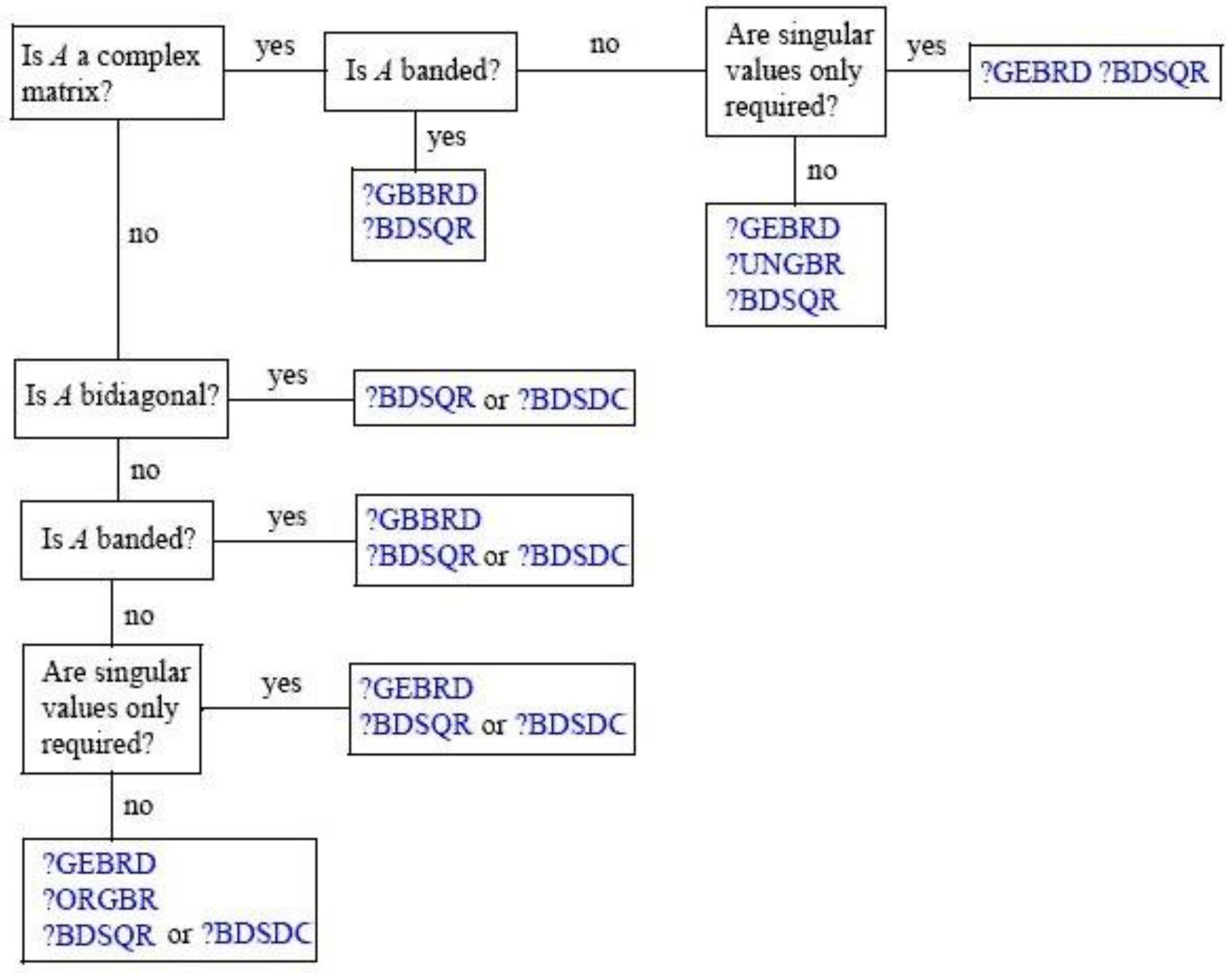


https://towardsdatascience.com/pca-and-svd-explained-with-numpy-5d13b0d2a4d8

http://www.inf.ed.ac.uk/teaching/courses/iaml/2011/slides/pca.pdf

Конкурентное преимущество SVD - разложения в том, что для его вычисления есть эффективные алгоритмы.

В том числе для огромных разреженных матриц, которые часто встречаются на практике (в рекомендательных системах, биоинформатике etc).



Упрощённая схема вычисления SVD-разложения. На каждый случай — свой алгоритм!

И это, ещё, без учёта разреженности матрицы.

Но не **PCA** единым хорош **SVD**!

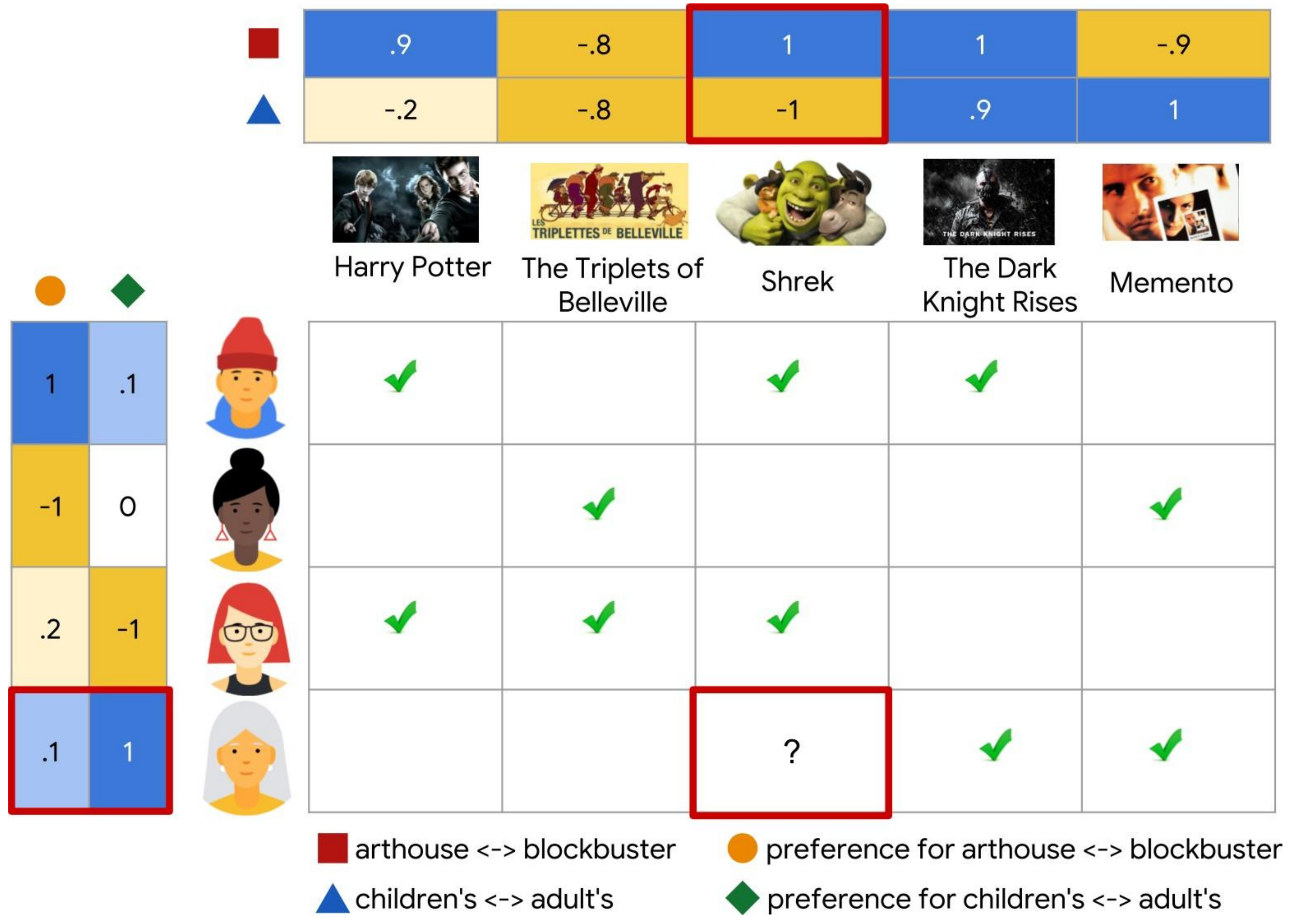
Приложения SVD

Коллаборативная фильтрация и многое другое

В задачах рекомендации важную роль играет матрица взаимодействий пользователя с контентом.

Представьте Netflix:

- ➢ Миллионы пользователей;
- ➢ Тысячи фильмов и сериалов;
- ➢ Каждый пользователь, в среднем, смотрит < 100 из них.



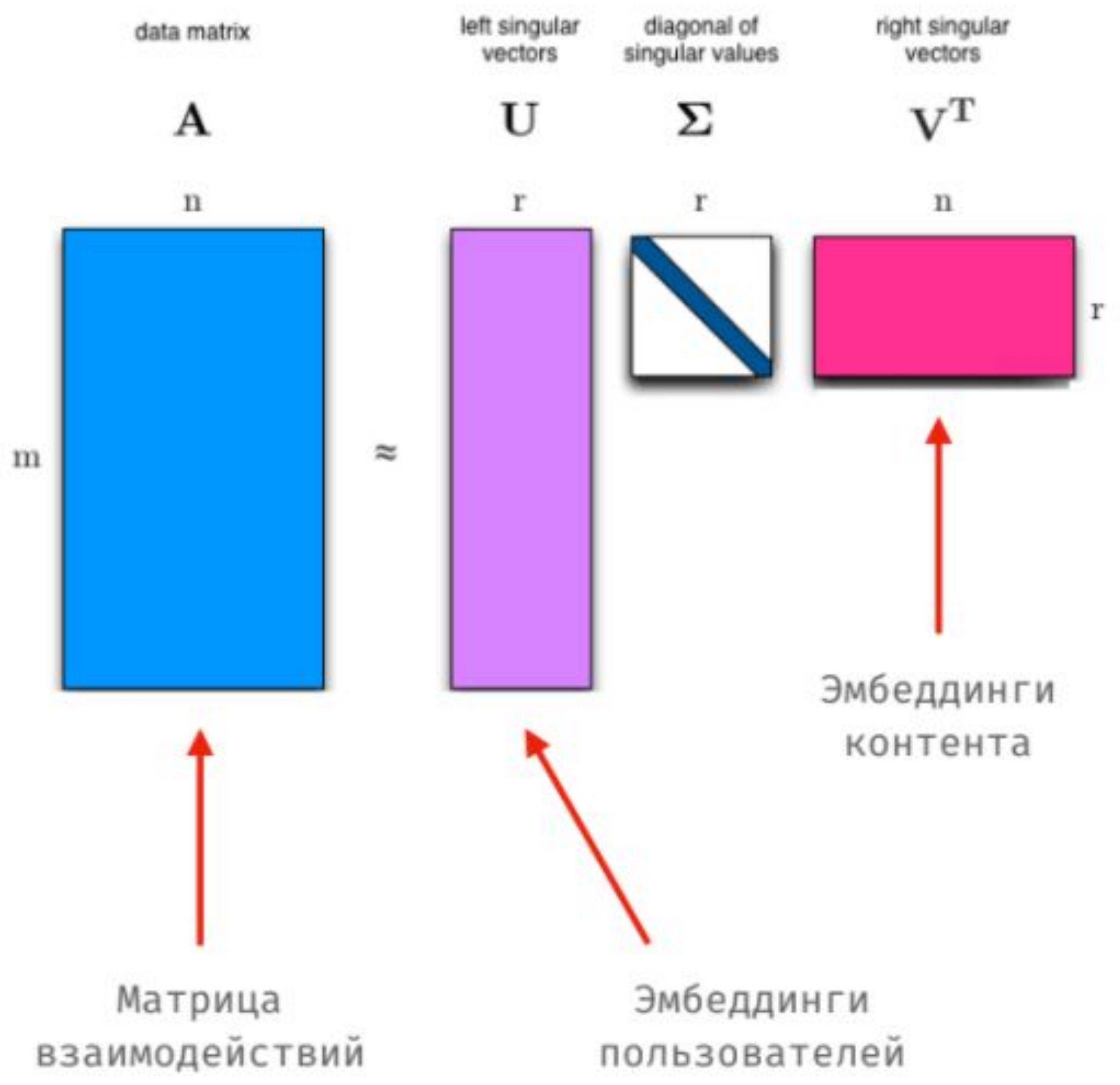
https://developers.google.com/machine-learning/recommendation/collaborative/basics

Гигантская разреженная матрица. На пересечении (пользователь, столбец) — оценка или любая другая полезная информация (“посмотрел ли до конца” и другие прокси-метрики)

Усеченное SVD-разложение ранга r матрицы взаимодействий позволяет получить эмбеддинги размерности r как для пользователей, так и для контента!

Более того: чем больше скалярное произведение эмбеддингов пользователя и контента, тем выше шанс, что пользователю понравится контент!

Это позволяет построить простейшую рекомендательную систему!



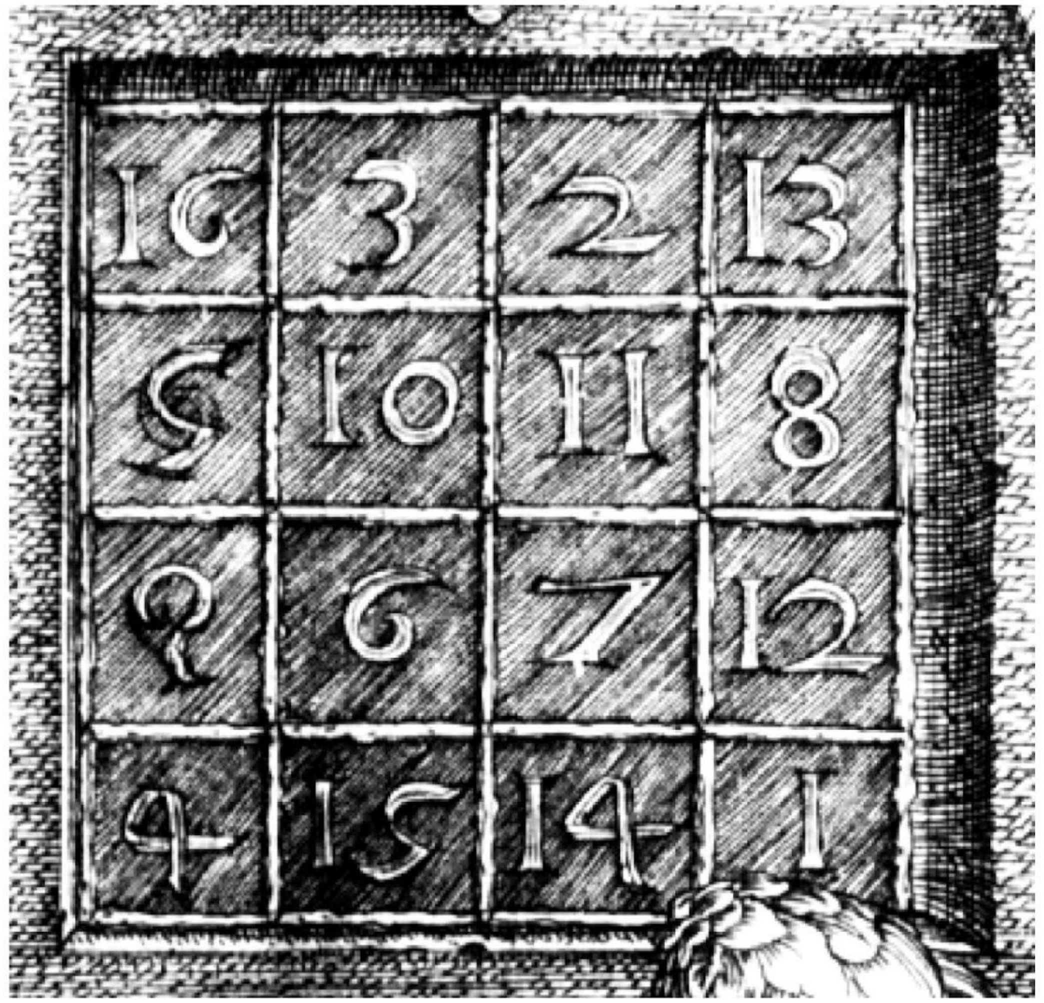
Рекомендательные системы на основе коллаборативной фильтрации — основа рекомендательных систем в Яндексе — в рекламе, Дзене и т.д.

Разумеется, там используются более продвинутые матричные факторизации. Но идея та же.

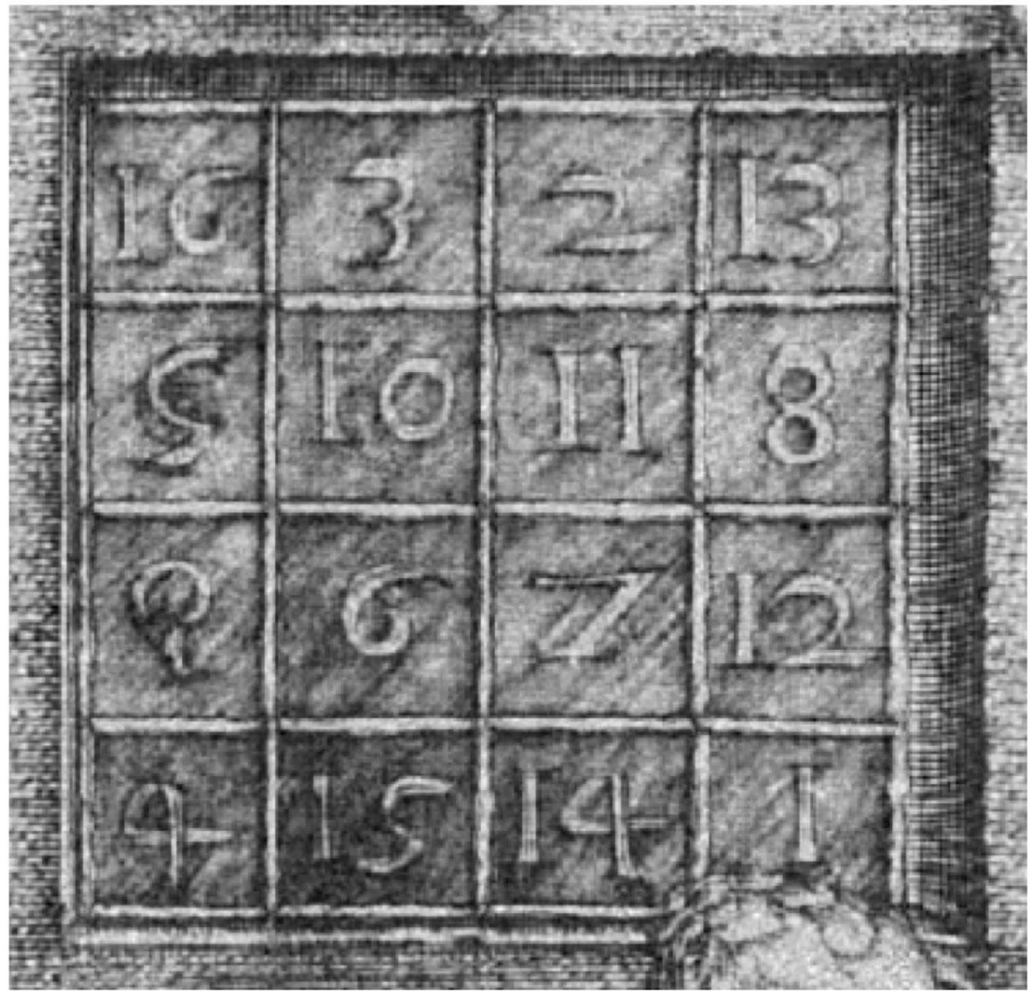
Статья про коллаборативную фильтрацию в Яндекс. Дзене: https://habr.com/ru/company/yandex/blog/490140/

Кроме того, SVD - разложение можно использовать как механизм сжатия изображений и видео…

Detail from Durer's Melancolia, dated 1514., 359x371 image



EOF reconstruction with 50 modes



http://www.columbia.edu/itc/applied/e3101/SVD_applications.pdf