

# Продвинутая линейная алгебра

Содержание:

Что такое матрица?

Какие бывают матрицы? Типы матриц  
Собственные векторы и собственные значения  
Матричные разложения

Что такое матрица?

$$M = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \end{bmatrix}$$

Матрица — математический объект, записываемый в виде прямоугольной таблицы.

Диагональная матрица — квадратная матрица, все элементы которой, кроме диагональных, равны 0.

$$M = \begin{bmatrix} A_{1,1} & 0 & 0 \\ 0 & A_{2,2} & 0 \\ 0 & 0 & A_{3,3} \end{bmatrix}$$

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

Единичная матрица — квадратная матрица, все диагональные элементы которой равны 1, остальные элементы равны 0.

## Примеры матриц

Матрица - прямоугольная таблица чисел:

$$\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix}$$

Матрица размера  $m \times n$

Матрица размера  $2 \times 2$

$$\begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix}$$

Матрица размера  $2 \times 3$

$$\begin{pmatrix} 0 & 1 & 5 \\ -1 & 2 & -2 \end{pmatrix}$$

Матрица размера  $3 \times 1$

$$\begin{pmatrix} 3 \\ -1 \\ 8 \end{pmatrix}$$

## Типы матриц

### Типы матриц

По внешнему виду:

- квадратные
- прямоугольные
- матрицы-строки, матрицы-столбцы
- единичные
- диагональные
- симметричные
- ортогональные

По внешнему виду матрицы различаются во-первых формой.

Они бывают **квадратные** или **прямоугольные** или **вообще матрицы-строки и матрицы-столбцы**, которые на самом деле вектора. Квадратные часто возникали когда у нас были преобразования одного и того же пространства. А прямоугольные — это просто набор векторов, упакованные для удобства или это отображение пространства одной размерности в пространство другой размерности.

Бывает **единичная матрица** с единицами на диагонали, которая действует как тождественное отображение. Также бывают **диагональные матрицы** — это матрицы, у которых на главной диагонали стоят какие-то числа, а все остальное- это нули.

**«симметричные матрицы»** - это матрицы, которые симметричны относительно главной диагонали, то есть если отразить нашу матрицу относительно нее, то получатся ровно те же числа.

**Ортогональные матрицы** задают ортогональные преобразования пространства, то есть в размерности 2 это повороты, в большей размерности — это как бы аналоги поворотов.

По смыслу матрицы бывают как наборы векторов, линейные преобразования и матрицы, похожие на ковариации случайной величины или на попарное скалярное произведение векторов.

$$A_{m \times n} := \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

1.  $m=n \Rightarrow$  квадратная, иначе прямоугольная

$$A_{n \times n} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nj} & \dots & a_{nn} \end{pmatrix}$$

2.  $m=1 \Rightarrow$  матрица-строка

$$A = (a_1 \ a_2 \ \dots \ a_n)$$

3.  $n=1 \Rightarrow$  матрица-столбец

$$A = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_m \end{pmatrix}$$

4. Нулевая матрица, если все элементы = 0

$$O = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

5. Диагональная (единичная – частный случай диагональной матрицы) – все элементы, кроме элементов главной диагонали равны 0

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

6. Треугольная (нижнетреугольная - матрица у которой все элементы над главной диагональю равны 0, верхнетреугольная - матрица у которой все элементы под главной диагональю равны 0)

7. Транспонированная матрица

$$A_{m \times n} \rightarrow A_{n \times m}^T. \text{ Пример: } A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \rightarrow A^T = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix}$$

8. Симметричная матрица - квадратная матрица, все элементы которой удовлетворяют условию  $a_{ij} = a_{ji}$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \dots & a_{3n} \\ \vdots & & \ddots & & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \dots & a_{nn} \end{pmatrix};$$

9. Ортогональная матрица

$$AA^T = A^T A = E$$

Если матрица  $Q$  ортогональна, а  $Q'$  — транспонированная к ней матрица, то

$$Q' \cdot Q = \begin{pmatrix} q_{11} & q_{21} & \dots & q_{n1} \\ q_{12} & q_{22} & \dots & q_{n2} \\ \vdots & \ddots & \ddots & \vdots \\ q_{1n} & q_{2n} & \dots & q_{nn} \end{pmatrix} \cdot \begin{pmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ q_{n1} & q_{n2} & \dots & q_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}.$$

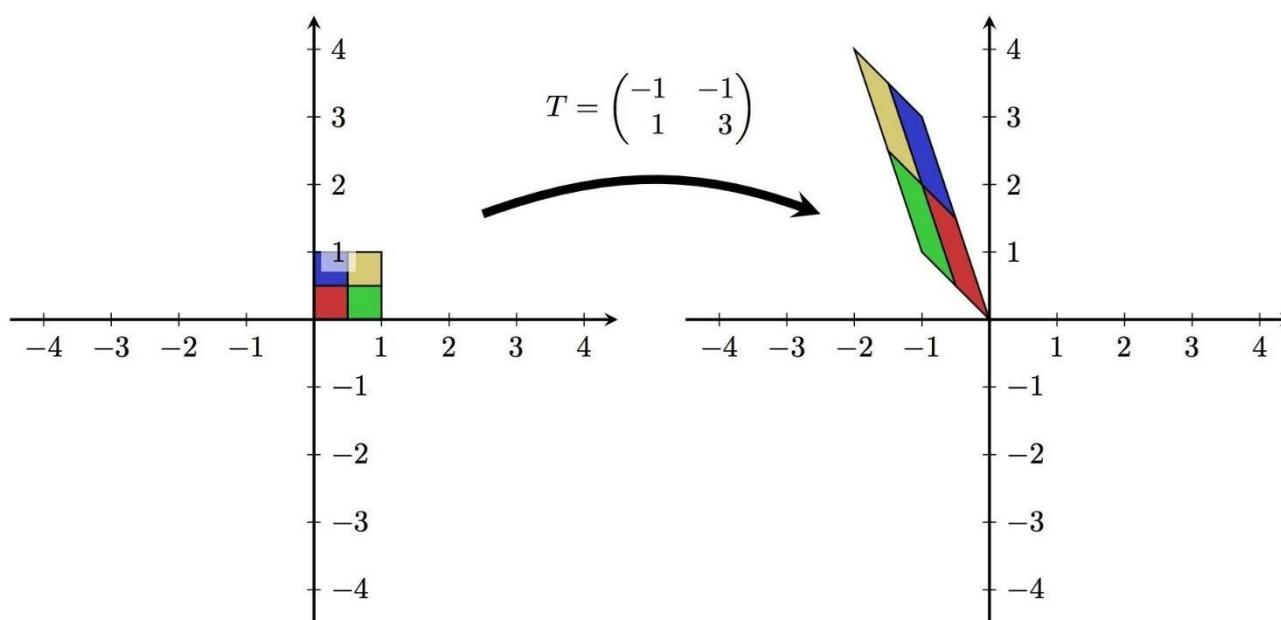
Итак, если матрица  $Q$  ортогональна, то

$$Q' \cdot Q = E. \quad (3)$$

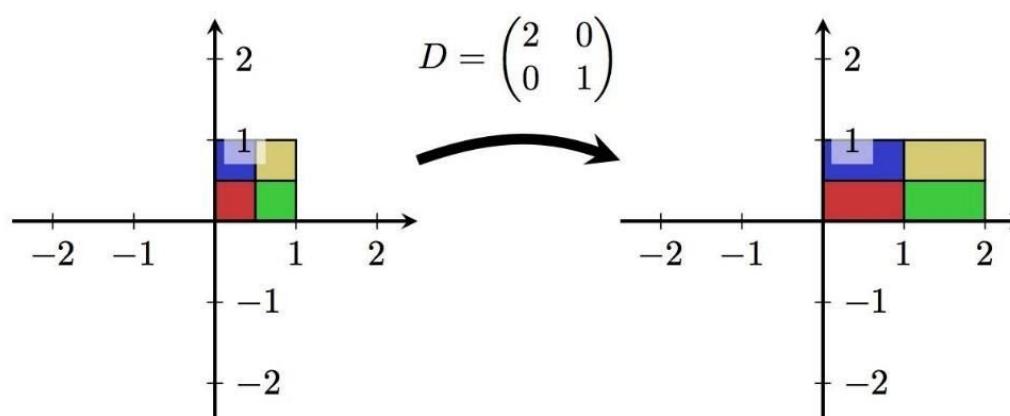
При умножении вектора на матрицу происходит преобразование пространства. При умножении вектора  $n$  на квадратную матрицу размерности  $(n, n)$ , вектор преобразуется в том же пространстве. При умножении вектора размерности  $n$  на прямоугольную матрицу размерности  $(m, n)$  вектор преобразуется в пространство  $m$  с размерностью  $m$ .

**Примеры преобразования (для простоты взяты квадратные матрицы):**

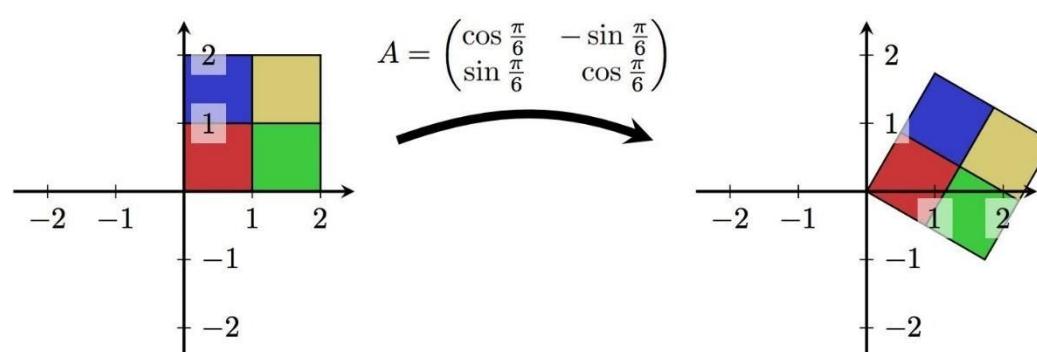
Преобразование с помощью произвольной матрицы.



Преобразование с помощью диагональной матрицы (при преобразовании с помощью диагональной матрицы происходит растяжение относительно разных осей).



Преобразование с помощью ортогональной матрицы (при преобразовании с помощью ортогональной матрицы происходит поворот на заданный угол).

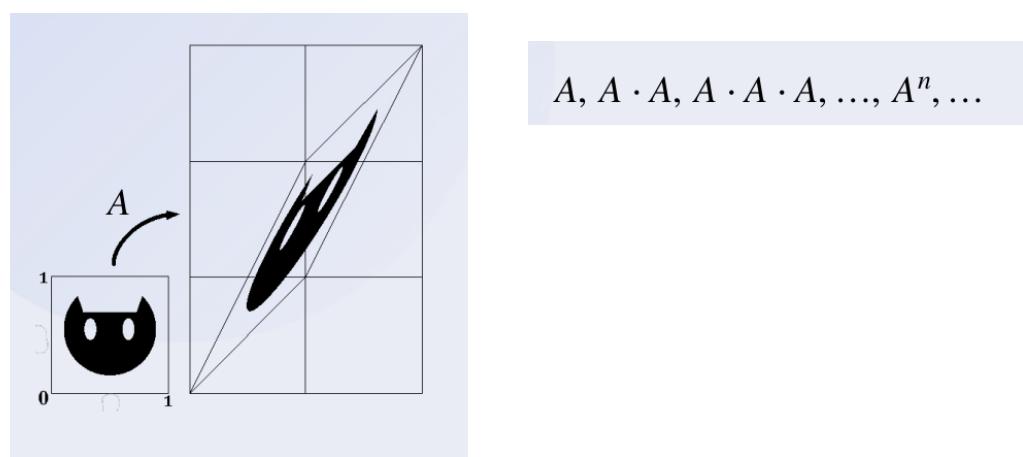


## Собственные векторы и собственные значения

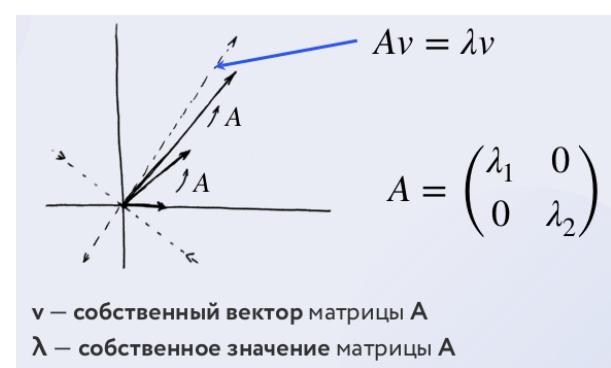
Пример. Отображение двумерной плоскости в себя, заданное матрицей  $\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$ . Оно действует вот так:



На рисунке изображено что происходит с образом единичного квадрата. 0 остается на месте, точка a1,0 отправляется в другую точку, точка a0,1 отправляется в другой угол и наконец вершина квадрата отправляется в дальний угол параллелограмма. Все остальные точки, как мы помним, как бы наклеены на плоскость и отправляются туда, куда их заставляет двигаться отображение, заданное образами этих точек. Из картинки мы уже имеем некоторую интуицию как это отображение работает. Посмотрим на то, как действует вот такая последовательность отображений, когда мы применяем сначала эту матрицу один раз, потом применяем ее к себе, к тому, чтобы получилось еще раз, потом еще раз, еще раз. По сути это то же самое, что мы действуем последовательностью отображений, то есть сначала матрицей A, потом ее квадратом, потом ее кубом, потом в n-ой степени и так далее.



Будет происходить примерно такое, что каждый вектор, допустим  $(1, 0)$  — это исходный вектор, он отправляется матрицей A в другой вектор, следующее применение матрицы A отправляет его в следующий вектор, и они продолжают растягиваться и растягиваться и по направлению приближаться к некоторому асимптотическому направлению.



То есть пунктирное направление — это предел направления векторов, когда мы применяем матрицу A очень много раз. Мы можем предположить, что если бы мы взяли вектор, который направлен ровно вдоль этого направления, то он бы матрице A не поворачивался никак, а в точности растягивался какое-то число раз вдоль этого направления. То есть раз мы вдоль этого направления удлиняемся, то наверно он растягивался бы в какое-то больше единицы число раз. То есть из нашего как бы наблюдения за поведением отображения A мы предполагаем, что есть такой вектор, который описывает это направление (вектор v), для которого действие матрицы A в точности эквивалентно умножению на некоторое число  $\lambda$ . Если такое выполнено, то v называется собственным вектором матрицы A, а  $\lambda$  называется собственным значением матрицы A. На самом деле у этой матрицы есть два собственных вектора и два собственных значения. И второй пунктирный вектор перпендикулярен первому, направлен вдоль этого направления, и вдоль него наша матрица сжимает все. То есть вдоль одного направления мы растягиваемся, вдоль другого мы сжимаемся. Но это совсем не те же самые направления, что в исходной оси координат. Однако если мы рассмотрим что происходит в базисе из собственных векторов, то есть мы перейдем в систему координат, где пунктирные векторы становятся новыми осями x и y, тогда в этой новой системе координат наша матрица будет устроена вот так:

$$A = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Это будет диагональная матрица с числами  $\lambda_1$  и  $\lambda_2$  на диагонали, где  $\lambda_1$  и  $\lambda_2$  - это собственные числа, и она будет действовать в точности как умножение на одну координату, и умножение на другую координату. То есть мы видим, что как только мы нашли два собственных вектора у нашей матрицы, мы можем ее переписать в гораздо более простом виде и получить гораздо более интуитивно понятное представление о том, как она действует на наше векторное пространство. Это то, что еще называется «жорданово-нормальная форма», когда мы приводим матрицу к виду, максимально близкому к диагональному

### Как найти собственные значения.

Собственные значения мы найдем из следующего соображения. Если мы просто предположим, что собственное значение есть, то тогда мы знаем, что для некоторого ненулевого вектора  $v$  выполнено вот такое условие:  $Av = \lambda v$ . Мы можем  $\lambda v$  перенести в левую часть, получается такое уравнение:  $Av - \lambda v = 0$ . Дальше мы можем вектор  $v$  вынести за скобку, расписав вместо просто числа  $\lambda$ , заменив его  $\lambda$  умножить на единичную матрицу  $I$ . Мы знаем, что единичная матрица, умножаясь на вектор, ничего с ним не делает, оставляет его ровно таким же, поэтому  $\lambda$  умножить на  $I$  умножить на  $v$ , то же самое, что  $\lambda v$ , соответственно мы приходим к такому уравнению:  $(A - \lambda \cdot I)v = 0$ . В скобках у нас записана некоторая матрица, причем мы знаем, что раз мы ее применили к ненулевому вектору  $v$  и получили 0, то значит матрица необратима. Потому что если бы мы применили обратную матрицу, то мы должны были бы из нуля восстановить обратно вектор  $v$ , но из нуля можно получить только ноль, то есть такого быть не может.

$$Av - \lambda v = 0 \Leftrightarrow (A - \lambda \cdot I)v = 0$$

Значит сама матрица  $A - \lambda I$  вырождена, необратима, мы знаем из предыдущей лекции, что это означает, что ее определитель равен нулю, т.е.  $\det(A - \lambda I) = 0$ . Такой определитель  $A - \lambda I$  называется характеристическим многочленом матрицы  $A$ , когда мы рассматриваем его как многочлен степени  $n$  относительно переменной  $\lambda$ . То есть мы считаем, что  $\lambda$  — это некоторая переменная, которую мы хотим найти, то есть мы ищем собственное значение и с самого начала выписываем характеристический многочлен матрицы  $A$ , он имеет степень  $n$ , если  $n$  — это размерность матрицы, то есть если матрица была изначально скажем  $2 \times 2$ , то это квадратное выражение, а  $\det(A - \lambda I) = 0$  — характеристическое уравнение, то есть из матрицы  $2 \times 2$  мы получаем просто квадратное уравнение, которое мы умеем решать со школы. И просто по построению, по всему тому, что мы делали, мы корни характеристического уравнения — это и есть собственные значения матрицы  $A$ . То есть нам достаточно решить характеристическое уравнение, чтобы найти все возможные числа  $\lambda$ , для которых вот такое условие:  $Av = \lambda v$  может быть вообще выполнено.

У симметричных матриц есть одно очень полезное свойство, что их характеристические уравнения всегда имеют решение и даже более того, ровно  $n$  решений и соответственно ровно  $n$  собственных значений. Некоторые из них могут в принципе совпадать, но это уже не так важно.

### Как найти собственные векторы.

Когда мы нашли собственное значение, как найти собственный вектор, соответствующий этому собственному значению. Мы снова вернемся к нашему исходному уравнению:

$$Av = \lambda v, \text{ которое мы опять же можем в таком виде переписать: } (A - \lambda I)v = 0.$$

Дальше на самом деле эта штука  $(A - \lambda I)v$  — это система линейных уравнений, записанная в матричном виде. Дальше мы находим вектор  $v$  как решение этой системы линейных уравнений. Однако поскольку матрица  $A - \lambda I$  по построению вырожденная, раз  $\lambda$  — это собственное значение, то эта матрица  $A - \lambda I$  обязательно вырожденная, у неё нет обратного, поэтому мы не можем найти обратную матрицу и придется решать систему линейных уравнений вручную. Просто как-то исключая неизвестные или еще как мы умеем это делать более примитивными способами. Так же поскольку матрица в левой части этой системы уравнений вырожденная, она (система) имеет не единственное решение обычно, а бесконечно много решений, как правило это одно и то же решение, определенное с точностью до умножения на число, то есть как говорят с точностью до пропорциональности. Таким образом и собственный вектор определен с точностью до пропорциональности. Это как то, что мы видели, что на самом деле вектор задает именно направление, а не есть какой-то как бы эффективный вектор. То есть все векторы этого направления нам годятся в качестве собственного.

Разберем это на примере. Возьмем вот такую матрицу:

$$A = \begin{pmatrix} -1 & -6 \\ 2 & 6 \end{pmatrix}$$

Дальше мы записываем определитель матрицы  $A - \lambda I$ , где  $I$  — это единичная матрица. Вспоминая что такое единичная матрица, мы увидим, что  $A - \lambda I$  — это просто добавить вот такие  $-\lambda$  на диагонали. То есть у нас была та же самая матрица, просто мы дописываем к каждому числу на диагонали такое дополнительное слагаемое  $-\lambda$ , берем от всего этого определитель и приравниваем нулю:

$$A = \begin{pmatrix} -1 & -6 \\ 2 & 6 \end{pmatrix} \rightarrow \begin{vmatrix} -1 - \lambda & -6 \\ 2 & 6 - \lambda \end{vmatrix} = 0$$

$$\lambda^2 - 5\lambda + 6 = 0$$

Получается вот такое квадратное уравнение:

Заметим, что на самом деле для матриц  $2 \times 2$  коэффициенты этого уравнения можно вычислить прямо из исходной матрицы, а именно 6 — это определитель исходной матрицы, то есть вспомним, определитель это -6, -12, получается +6, в точности то, что мы видим в уравнении. А то число, которое записано при  $\lambda$  — это минус сумма диагональных чисел, тоже обращаем

внимание:  $6 \cdot -1$  это  $5$ , здесь у нас как раз  $-5$ , а при  $\lambda^2$  всегда ровно единица, тоже можно это увидеть из того, что тут:  $-\lambda$ ,  $-\lambda$  умножается друг на друга, получается в точности  $\lambda^2$ . То есть для матриц  $2 \times 2$  есть такой простой способ не выписывать второй шаг, а прямо немедленно выписать все характеристическое уравнение.

$$A = \begin{pmatrix} -1 & -6 \\ 2 & 6 \end{pmatrix} \rightarrow \begin{vmatrix} -1 - \lambda & -6 \\ 2 & 6 - \lambda \end{vmatrix} = 0 \rightarrow \lambda^2 - 5\lambda + 6 = 0$$

Получаем такое квадратное уравнение, и мы можем его решить, считая его дискриминант как нас учили в школе, или мы можем просто угадать корни. Мы видим, что числа  $\lambda_1 = 2$  и  $\lambda_2 = 3$  — это корни этого уравнения. В этом можно убедиться, подставив. Можно воспользоваться теоремой Виета, которая говорит нам, что произведение корней, равное  $6$ , а сумма корней должна быть равна  $5$ . Дальше мы уже нашли собственные значения нашей матрицы, это  $2$  и  $3$ , и теперь мы найдем собственные векторы, соответствующие этим значениям. Как мы это сделаем? Мы для каждой  $\lambda$  запишем что же такое  $A - \lambda I$  и выпишем соответствующую систему линейных уравнений:  $A - 2I$  — это будет:  $-3, -6, 2, 4$ .

$$\lambda_1 = 2$$

$$\begin{cases} -3x - 6y = 0 \\ 2x + 4y = 0 \end{cases}$$

$$v_1 = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

Это матрица левой части этой системы. И решение этой системы — это вектор  $(x, y)$  и  $(2, -1)$ . Любой вектор, отличающийся от этого умножением на число, тоже подходит.

И для  $\lambda_2 = 3$  все работает ровно так же. Матрица левой части это  $A - 3I$ , и вот такой вектор у нас получается, когда мы решаем  $(3, -2)$ .

$$\lambda_2 = 3$$

$$\begin{cases} -4x - 6y = 0 \\ 2x + 3y = 0 \end{cases}$$

$$v_2 = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

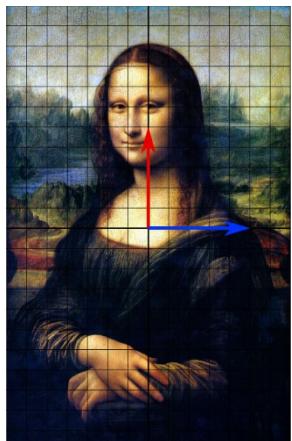
### Свойства собственных векторов.

Собственные векторы — направления, в которых матрица лишь растягивает или сжимает векторы, но не поворачивает.

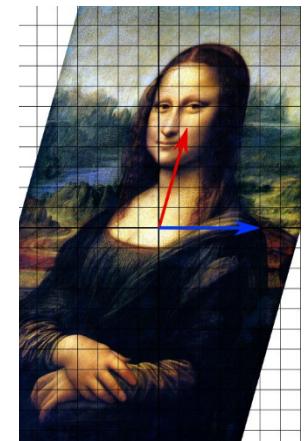
Второе полезное свойство — это если мы нашли  $n$  собственных векторов, то записав матрицу в базисе из этих векторов, мы получаем ее в диагональном виде. Что довольно удобно. Обычно когда у нас большая матрица, то не очень понятно как оно действует, умножаясь на вектора, а когда у нас матрица диагональная, это значит, что мы просто каждую отдельную координату умножаем на соответствующее число.

И собственные векторы, соответствующие наибольшим значениям, показывают вдоль каких направлений наша матрица больше всего растягивает наше пространство.

Что как раз используется при понижении размерности данных.



Представим изображение  $X$  в виде набора чисел и применим к нему некое преобразование  $A$ , после которого получили следующий результат: красный вектор изменил свое положение в пространстве, синий вектор не изменил свое начальное положение.



Вектор  $X$  (ненулевой вектор) будет называться собственным, если при преобразовании  $A$  получится коллинеарный вектор — тот же вектор, умноженный на некоторое скалярное значение.

$$AX = \lambda X$$

где  $X$  — собственный вектор (ненулевой!)

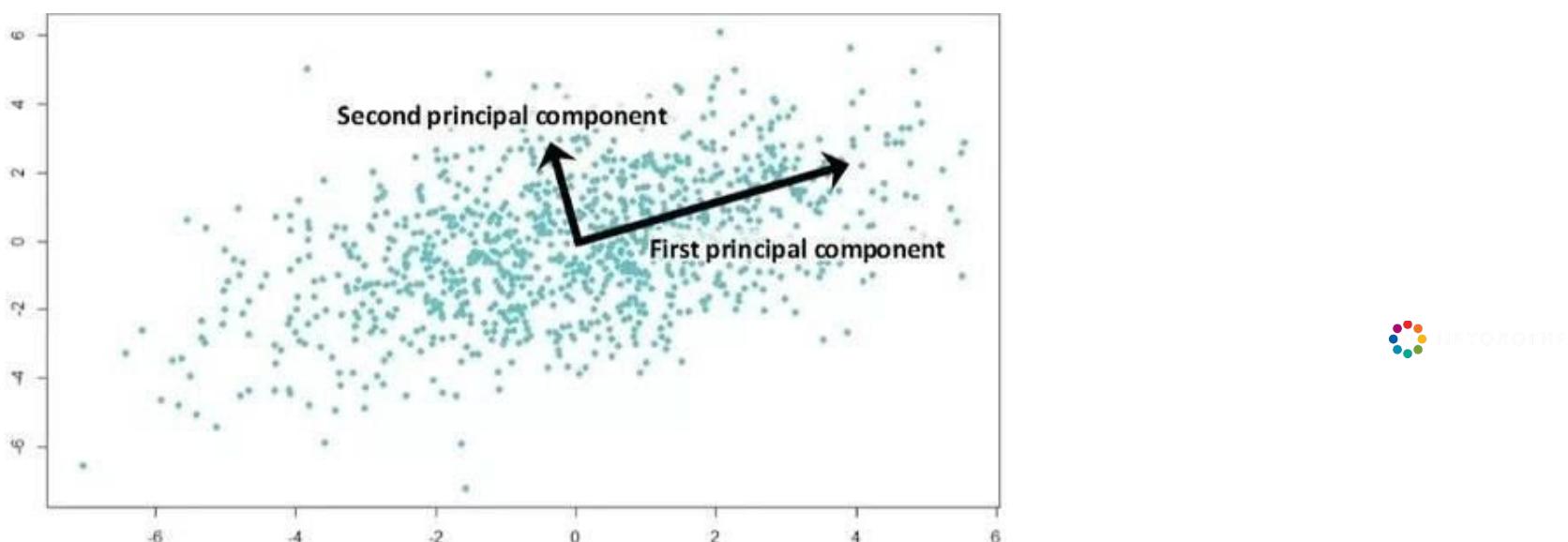
$\lambda$  — собственное значение

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

В примере выше может быть не более двух собственных векторов и собственных значений.

Собственные векторы и собственные значения: применение

1. Собственные векторы - направления, в которых матрица преобразования лишь растягивает или сжимает векторы, но не поворачивает.
2. Собственные вектора показывают направления наибольшего изменения.
3. Возникают при уменьшении размера матрицы (например, при преобразовании трехмерного пространства в двухмерное, при проектировании трехмерного пространства на собственные вектора произойдут наименьшие потери информации)



Собственные векторы и собственные значения (поиск)

$$A - \lambda E$$

характеристическая матрица матрицы A

$$|A - \lambda E|$$

характеристический многочлен матрицы A

$$|A - \lambda E| = 0$$

характеристическое уравнение матрицы A

A – матрица,  $\lambda$  – скаляр, E – единичная матрица

Теорема. Собственными числами матрицы A являются корни характеристического уравнения матрицы A и только они.

Для примера решим характеристическое уравнение матрицы A.

$$A = \begin{pmatrix} -1 & -6 \\ 2 & 6 \end{pmatrix} \quad A - \lambda E \quad \rightarrow \quad \begin{vmatrix} 1-\lambda & -6 \\ 2 & 6-\lambda \end{vmatrix} = 0$$

Считаем определитель характеристического уравнения матрицы A  $(-1-\lambda)(6-\lambda) - (-6) \cdot 2 = 0$

$$\lambda^2 - 5\lambda + 6 = 0$$

Решаем квадратное уравнение относительно  $\lambda$

$$\lambda_1 = \frac{5-1}{2} = 2$$

$$\lambda_2 = \frac{5+1}{2} = 3$$

Подставляем найденные значения в характеристическое уравнение матрицы A:

$\lambda_1 = \frac{5-1}{2} = 2$  подставляем в  $\begin{vmatrix} 1-\lambda & -6 \\ 2 & 6-\lambda \end{vmatrix}$  находим коэффициенты преобразования. Подставляем в  $A\bar{X}-\lambda X=0$  ( $A\bar{X}=\lambda X$ ), где  $X$  вектор с координатами  $(x,y)$ , получаем преобразование по каждой компоненте из которого находим соотношение координат собственного вектора.

$$\begin{cases} -3x - 6y = 0 \\ 2x + 4y = 0 \end{cases} \Rightarrow x = -2y$$

Соответственно для  $\lambda_2 = \frac{5+1}{2} = 3$  получим следующее соотношение координат вектора:

$$\begin{cases} -4x - 6y = 0 \\ 2x + 3y = 0 \end{cases} \quad x = -\frac{3}{2}y$$

$$\begin{pmatrix} -1 & -6 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \quad \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

**Ответ:** собственные числа:  $\lambda_1 = 2, \lambda_2 = 3$ , собственные векторы:  $\bar{u}_1 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \bar{u}_2 = \begin{pmatrix} 3 \\ -2 \end{pmatrix}$

т. е.  $= 2$

$$\begin{pmatrix} -1 & -6 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} 3 \\ -2 \end{pmatrix} = 3 \begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

Также можно заметить, что если квадратная матрица  $A$  имеет размерность 2, то характеристический многочлен матрицы  $A$  получается во 2 степени, что дает не более двух корней. Если брать матрицу с размерностью 3, получим многочлен третьей степени и не более трех корней и т.д.

## Матричные разложения

### Principal Component Analysis (PCA).

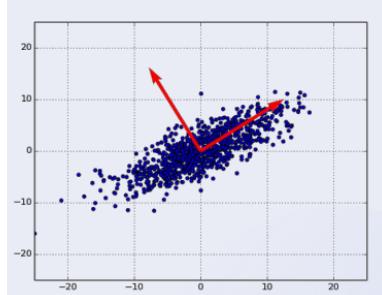
Техника Principal Component Analysis или обычно сокращается до PCA. Это метод понижения размерности данных. Он очень часто используется, буквально повсеместно. Его очень полезно знать. Что такое понижение размерности данных, зачем это нужно?

Представим себе, что наши данные — это набор векторов в  $N$ -мерном пространстве, где  $N$  — это довольно большое число. Мы можем хотеть, чтобы размерность стала поменьше, потому что  $N$  измерений — это много данных, они занимают много места в памяти и сложно на них что-то обучать, слишком много может быть признаков. Также понижение размерности данных используется, когда мы хотим уменьшить размер цифровых изображений, чтобы они занимали меньше места, при этом мы конечно хотим особенно не потерять в качестве, то есть чтобы вся важная информация, которая в этом наборе данных есть, она сохранилась, мы ее как бы минимально потеряли.

Другое полезное использование — когда у нас есть набор признаков для машинного обучения, мы их откуда-то взяли, насыптизовали, их получилось очень много, мы хотели бы, чтобы у нас осталось только какое-то меньшее количество только самых важных. Как правило, редко получается так, что какие-то признаки можно просто изначально полностью выкинуть, это было бы слишком легко. То есть то, что мы ищем — это проекция нашего  $N$ -мерного пространства в какое-то пространство меньшей размерности, которое сохраняет нам большую часть вариативности наших данных, то есть той информации, которая там содержится и при этом  $N$ , то есть то, куда мы проецируем, размерность этого поменьше. Как это делается? Сначала мы вычисляем матрицу попарных скалярных произведений наших векторов. Если у нас было  $N$  векторов, то у нас получается матрица  $N \times N$ , квадратная матрица, причем она симметрична, потому что мы помним, что

скалярное произведение — это симметричная операция, то есть если мы умножаем  $i$ -ый вектор на  $j$ -ый вектор, это то же самое, что мы умножаем  $j$ -ый вектор на  $i$ -ый вектор. Получается симметричная матрица, и такое у нее есть очень полезное дополнительное свойство, которое присуще именно матрицам скалярных произведений наборов векторов, что все ее собственные значения положительны. Как просто симметричная матрица она имеет полный набор собственных значений и собственных векторов и кроме того, все собственные значения еще и положительны.

Что такое геометрически собственные вектора и собственные значения матрицы скалярных произведений? Это если мы представляем наши точки данных, наши векторы как вот такое облако точек в  $N$ -мерном пространстве, исходное большое пространство, тогда мы можем представить, что мы как бы описали некоторый эллипсоид вокруг этого облака. То есть облако было более-менее сферическое, то облако получится сфера, а если оно какое-то вытянутое, то у нас будет эллиптическая поверхность, которая описана вокруг нашего облака точек. И тогда собственные векторы матрицы попарно скалярных произведений — это в точности оси этого эллипсоида.



А собственные значения этой матрицы отвечают за то, насколько эллипсоид, описанный вокруг точек, вытянут вдоль оси 1 или оси 2. У нас  $N$  осей и  $N$  собственных значений. Соответственно мы видим здесь, что наше исходное облако точек, наши исходные данные, оно больше вытянуто по оси 2, чем по оси 1. То есть если бы мы хотели понизить размерность этого облака и как можно меньше информации при этом потерять, то мы бы спроектировали его на прямую, проходящую через большую ось. И при этом мы теряем минимум информации, которая отвечает за изменчивость, за разброс вдоль направления меньшей оси. В точности так и работает РСА. Мы решаем какую размерность данных мы хотим в итоге получить, мы выбираем  $n$  самых больших собственных значений нашей матрицы попарно скалярных произведений, берем пространство, порожденное соответствующими собственными векторами, и проецируем на него. Таким образом мы получаем  $n$ -мерное представление данных, сохраняющее максимум информации, при том, что мы поставили цель отбросить столько измерений, сколько мы отбросили. Такая штука — это частный случай общей концепции матричных разложений, когда мы пытаемся исходный набор векторов или исходную матрицу как-то упростить, записав ее меньшим количеством чисел, меньшими матрицами, при этом сохранив большую часть информации.

**Разложение матрицы** — это представление ее в виде произведения некоторых других матриц, которые обладают какими-то нужными нам свойствами. Это или небольшая размерность, или еще что-то. На примере РСА мы уже видели так называемый спектральное разложение, когда мы представляем симметричную матрицу  $A$  в виде произведения ортогональной и диагональной матрицы, причем диагональная матрица составлена из собственных значений матрицы  $A$ .

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

То есть на самом деле это по сути то же самое, что мы берем и нашу матрицу как бы поворачиваем с помощью так называемого ортогонального преобразования  $S$ . Можно сказать еще, что  $S$  описывает базис собственных значений в матрице  $A$ .

### Спектральное разложение

Пример: спектральное разложение  $X$

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

$X$  — симметричная,  $S$  — ортогональная,  $D$  — диагональная из собственных значений  $X$ .

Преобразование с помощью спектрального анализа используется редко. Так как матрица  $X$  обязательно должна быть симметричной.

Часто встречаются (например, в теории вероятности) квадратичные формы вида:

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

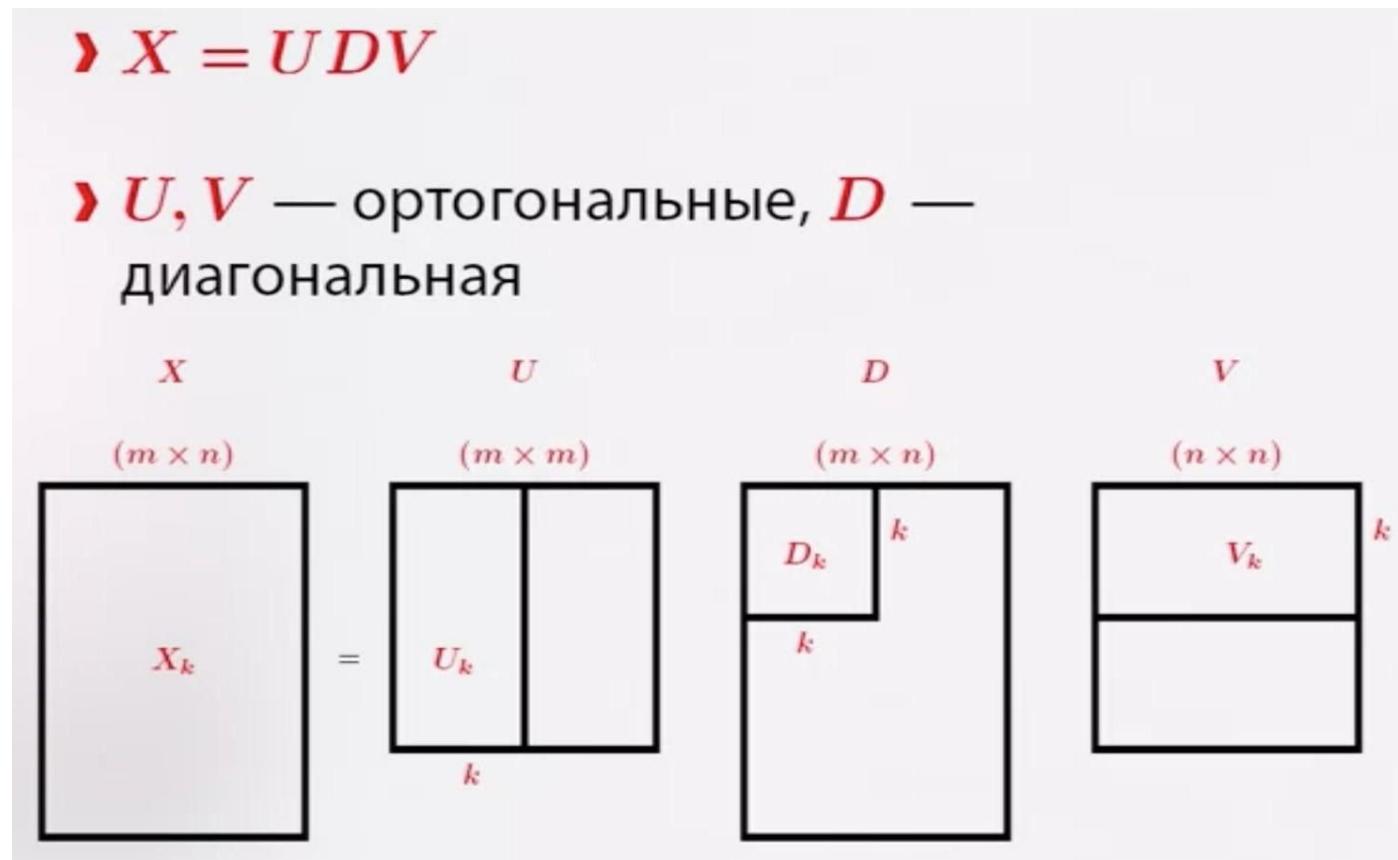
с помощью спектрального разложения приводим к более простому виду:

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

Т. е. с помощью спектрального разложения мы привели квадратичную форму с большим количеством аргументов к более простому в виде суммы произведений собственных значений матрицы на квадрат произведения ортогональной матрицы на матрицу  $Y$ .

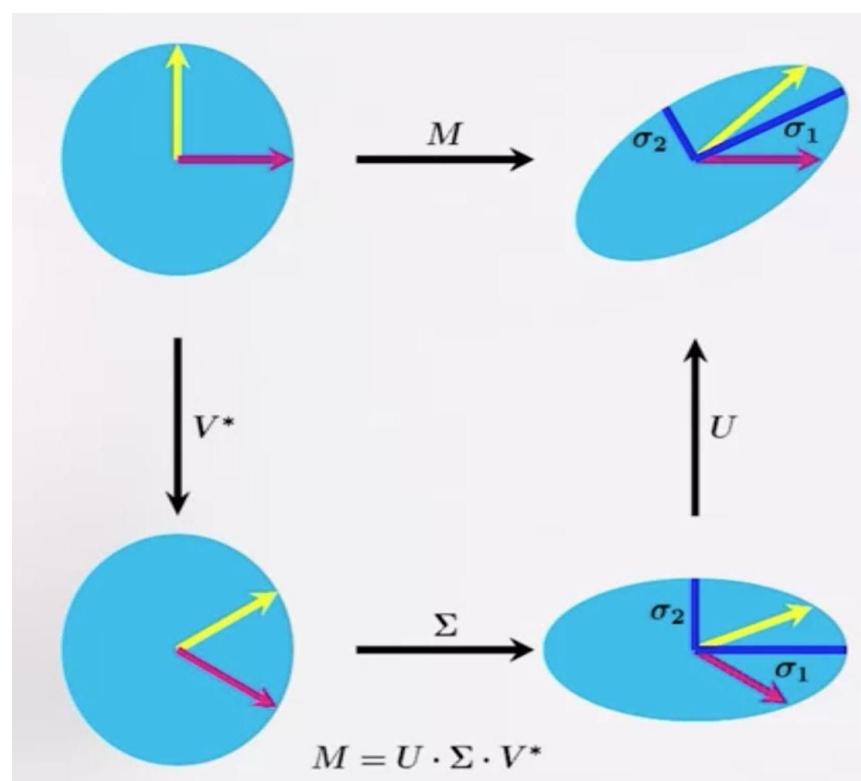
### Сингулярное разложение матрицы

В случае произвольной матрицы используется сингулярное разложение матрицы.



Для уменьшения хранящихся данных можно вычислять не матрицу  $X$ , а приближенную по значениям:  $X_k=U_kD_kV_k$

Сингулярное разложение представляет линейное преобразование в виде композиции: вращения (с помощью ортогональной матрицы), растяжения по осям (с помощью диагональной матрицы), вращения (с помощью второй ортогональной матрицы).



**Ранг матрицы** — это в каком-то смысле мера «сложности» нашей матрицы, то есть если матрица задает какое-то отображение, то ранг — это размерность образа отображения, то насколько сильно матрица уменьшает исходное пространство. Обозначается обычно как  $\text{rk } A$  (сокращение от английского rank), иногда полностью пишут  $\text{rank}$ .

Еще можно ранг трактовать как максимальное количество линейно независимых столбцов или строк матрицы. То есть у нас есть какая-то матрица, у нее есть строки и столбцы, и в хорошем случае они линейно независимы, но в плохом все в целом они линейно зависимы, но там можно выбрать какую-то линейно независимую подсистему и ранг — это в точности есть ее максимальный размер. Простое наблюдение состоит в том, что если  $A$  — это прямоугольная матрица размера  $m \times n$ , то ее ранг не превосходит минимума из ее размерностей, просто потому, что больше строк нет (и столбцов). То есть мы в любом случае ограничены меньшим из этих двух чисел, и даже если строк например больше, чем столбцов, то все равно там не может быть больше линейно независимых строк.

Также можно сказать, что ранг — это максимальный размер подматрицы с ненулевым определителем. Или максимальный размер ненулевой, невырожденной подматрицы, который мы можем найти в нашей матрице.

У ранга есть такое свойство, что когда мы матрицу умножаем, то ранг того, что получится, он не превосходит минимума из двух рангов, то есть когда мы две матрицы перемножили, или, что то же самое, взяли композиции отображений, соответствующих этим матрицам, то сложность того, что получилось, она не превосходит сложности самого простого из сомножителей.

Тут нам становится понятно, что раз ранг — это мера сложности матрицы, то неплохо бы научиться приближать сложные матрицы с какими-то матрицами попроще. Это то, что называется приближение матрицей меньшего ранга.

### Приближение матрицей меньшего ранга

1. Ранг - максимальное количество линейно независимых столбцов или строк.

Пусть дано  $p$  столбцов  $A_1, A_2, \dots, A_p$ . Если существуют такие числа  $c_1, c_2, \dots, c_p$ , что:

1. не все эти числа равны нулю,
2. линейная комбинация  $c_1A_1 + c_2A_2 + \dots + c_pA_p = \sum_{m=1}^p c_m A_m$  равна нулевому столбцу (т.е. столбцу, все элементы которого нули), то говорят, что столбцы  $A_1, A_2, \dots, A_p$  линейно зависимы. Если для данного набора столбцов таких чисел  $c_1, c_2, \dots, c_n$  не существует, столбцы называются линейно независимыми.

2. Ранг - максимальный размер подматрицы с ненулевым определителем.

Важно помнить, что в случае если речь идет о прямоугольной матрице, т.е. размерности  $m \times n$ , то ранг матрицы будет не больше чем минимальное значение из  $m, n$ .

**rank( $X$ )  $\leq \min(m, n)$ , если  $X$  - матрица  $m \times n$**

В случает, когда матрицу  $X$  можно представить в виде произведения:  $X = AB$ , где  $A$  размера  $(m \times k)$ ,  $B$  -размера  $(k \times n)$  и  $k < m, k < n$ . Ранг матрицы будет не больше значения  $k$ .

**rank( $X$ )  $\leq k$**

Приближение матрицей меньшего ранга позволяет уменьшить объем сохраняемых данных, при этом ее значение максимально приближено к исходной матрице.

### Что значит приблизить

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

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

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

$X'$  это матрица меньшего ранга. Параметр  $k$  по сути степень сжатия. Например, если есть  $m$  объектов и  $n$  признаков к ним, то можно сократить количество признаков до  $k$ .

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

Норма Фробениуса (отклонение):

Итоговая задача выглядит как нахождение таких матриц  $U, V$ , при которых  $X'$  максимально приближена к матрице  $X$ :

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

[Матричные разложения \(пример применения\)](#)

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

$\mathbf{X}$  – матрица размерность ( $m \times n$ ), где  $m$  – строчек (объектов),  $n$  – столбцов (признаков столбцов).

У размерности ( $m \times k$ ) с помощью которой восстанавливаются исходные признаки  $\mathbf{X}$  с максимально возможной точностью.

- › Пусть  $\mathbf{X}$  – 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

$\mathbf{X}$  – матрица размерность ( $m \times n$ ), где  $m$  – строчек (кол-во пользователей),  $n$  – столбцов (фильмы).

$x_{ij}$  – это оценка пользователя  $i$  фильму  $j$ .

Стоит задача спрогнозировать оценки пользователей, что еще не поставили свои оценки.

Так как в матрице  $\mathbf{X}$  имеются пропуски, которые мы должны заполнить, то для нахождения значений матриц  $U, V$  берем только известные данные.

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

По сути значения матриц  $U, V$  – это оценка фильма с учетом интересов конкретного пользователя и других пользователей, и учетом качества самого фильма по предыдущим оценкам других пользователей, а так же оценкам, которые поставил конкретный пользователь для других фильмов.

#### Матричные разложения (связь SVD и низкорангового приближения)

$$\hat{\mathbf{X}} = \underset{\operatorname{rg} \hat{\mathbf{X}} \leq k}{\operatorname{argmin}} \| \mathbf{X} - \hat{\mathbf{X}} \|$$

где  $\hat{\mathbf{X}}$  – приближенная к  $\mathbf{X}$  матрица ранга не больше чем  $k$

$$\mathbf{X} = U \cdot D \cdot V^T$$

сингулярное разложение матрицы

$$\begin{array}{cccc}
 X & U & D & V \\
 (m \times n) & (m \times m) & (m \times n) & (n \times n) \\
 X_k & = & U_k & D_k V_k^T \\
 & & k & k
 \end{array}$$

$$X_k = U_k D_k V_k^T$$

Если матрицу  $X$  представить в виде сингулярного разложения и сделать усечение, то полученное  $X_k$ , будет наилучшим приближением матрицы  $X$  матрицей ранга  $\leq k$  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

Матрица оценок пользователей фильмов  $X$ , раскладываем с помощью сингулярного разложения и делаем SVD. Берем матрицу профилей пользователей как матрицу  $U_k D_k$ , а матрицу  $V_k$  как матрицу профилей фильмов.

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

