



Основы линейной алгебры в Numpy

Лавприт Сингх-Пальчевская
младший научный сотрудник МГУ им. Ломоносова,
кафедра биоинженерии

Проверка связи



Отправьте «+», если меня видно и слышно

Если у вас нет звука или изображения:

- перезагрузите страницу
- попробуйте зайти заново
- откройте трансляцию в другом браузере (используйте Google Chrome или Microsoft Edge)
- с осторожностью используйте VPN, при подключении через VPN видеопотоки могут тормозить

Многообразие библиотек Python

- **NumPy — работа с большими матрицами и массивами**
- Pandas — обработка и анализ структурированных данных
- SciPy — научные вычисления
- Matplotlib, Seaborn, Plotly — построение графиков и визуализация, статистическая визуализация
- Scikit Learn, TensorFlow, Keras, PyTorch — машинное обучение, анализ больших данных



О чём поговорим сегодня

1. Повторим основы линейной алгебры
2. Познакомимся с библиотекой NumPy и ее основным функционалом
3. Отработаем на практике основные операции над векторами и матрицами с использованием NumPy

Основы линейной алгебры

Линейная алгебра — раздел алгебры, изучающий объекты *линейной природы*:

- скаляр,
- вектор и векторные (или линейные) пространства,
- матрицы, определители и системы линейных уравнений (СЛУ),
- тензоры.

Линейная алгебра — раздел алгебры, изучающий объекты *линейной природы*:

- скаляр,
- вектор и векторные (или линейные) пространства,
- матрицы, определители и системы линейных уравнений (СЛУ),
- тензоры.

Основные **инструменты** линейной алгебры:

- базовые операции с объектами *линейной природы* (сложение, вычитание, произведение и т.п.)
- линейные отображения,
- сопряжение,
- тензорное исчисление.

Линейная алгебра в NumPy

Numpy - это основная библиотека для математических и научных вычислений на Python. Она предоставляет высокопроизводительный модуль для работы с многомерными массивами и инструменты для работы с этими массивами: сложные математические функции, генераторы случайных чисел, процедуры линейной алгебры, преобразования Фурье и многое другое.



Линейная алгебра в NumPy



`a = np.array([1, 2, 3])` Создание вектора $a = (1, 2, 3)$

`b = np.array([[1, 2, 3], [4, 5, 6]])` Создание матрицы b :

<code>print(a.ndim)</code>	число измерений массива
<code>print(a.shape)</code>	размеры массива, его форма
<code>print(a.size)</code>	количество элементов массива

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

Линейная алгебра в NumPy

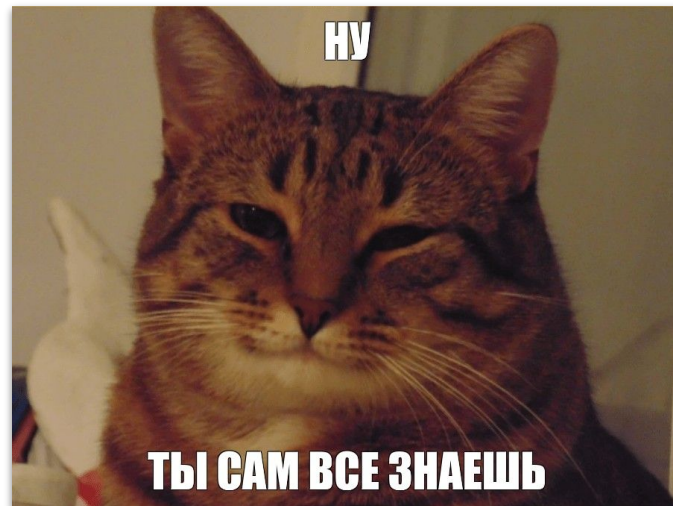
```
a = np.zeros((2,2))    # Create an array of all zeros
print(a)               # Prints "[[ 0.  0.]
                        #           [ 0.  0.]]"

b = np.ones((1,2))     # Create an array of all ones
print(b)               # Prints "[[ 1.  1.]]"

c = np.full((2,2), 7)  # Create a constant array
print(c)               # Prints "[[ 7.  7.]
                        #           [ 7.  7.]]"

d = np.eye(2)          # Create a 2x2 identity matrix
print(d)               # Prints "[[ 1.  0.]
                        #           [ 0.  1.]]"

e = np.random.random((2,2)) # Create an array filled with random values
print(e)               # Might print "[[ 0.91940167  0.08143941]
                        #           [ 0.68744134  0.87236687]]"
```



Индексация массивов:

```
# Create the following rank 2 array with shape (3, 4)
# [[ 1  2  3  4]
#  [ 5  6  7  8]
#  [ 9 10 11 12]]
a = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])

# Use slicing to pull out the subarray consisting of the first 2 rows
# and columns 1 and 2; b is the following array of shape (2, 2):
# [[2 3]
#  [6 7]
#  [10 11]]
b = a[:, 1:3]
```

Базовые математические операции:

```
x = np.array([[1,2],[3,4]], dtype=np.float64)
y = np.array([[5,6],[7,8]], dtype=np.float64)
```

```
# Elementwise sum; both produce the array
```

```
# [[ 6.0  8.0]
```

```
# [10.0 12.0]]
```

```
print(x + y)
```

```
print(np.add(x, y))
```

```
# Elementwise difference; both produce the array
```

```
# [[-4.0 -4.0]
```

```
# [-4.0 -4.0]]
```

```
print(x - y)
```

```
print(np.subtract(x, y))
```

```
# Elementwise product; both produce the array
```

```
# [[ 5.0 12.0]
```

```
# [21.0 32.0]]
```

```
print(x * y)
```

```
print(np.multiply(x, y))
```

```
# Elementwise division; both produce the array
```

```
# [[ 0.2          0.33333333]
```

```
# [ 0.42857143  0.5          ]]
```

```
print(x / y)
```

```
print(np.divide(x, y))
```

```
# Elementwise square root; produces the array
```

```
# [[ 1.          1.41421356]
```

```
# [ 1.73205081  2.          ]]
```

```
print(np.sqrt(x))
```

Линейная алгебра в NumPy

Базовые математические операции:

```
x = np.array([[1,2],[3,4]], dtype=np.float64)
y = np.array([[5,6],[7,8]], dtype=np.float64)
```

```
# Elementwise sum; both produce the array
```

```
# [[ 6.0  8.0]
```

```
# [10.0 12.0]]
```

```
print(x + y)
```

```
print(np.add(x, y))
```

```
# Elementwise difference; both produce the array
```

```
# [[-4.0 -4.0]
```

```
# [-4.0 -4.0]]
```

```
print(x - y)
```

```
print(np.subtract(x, y))
```

Для матричного умножения
используйте метод `np.dot()`

```
# Elementwise product; both produce the array
```

```
# [[ 5.0 12.0]
```

```
# [21.0 32.0]]
```

```
print(x * y)
```

```
print(np.multiply(x, y))
```

```
# Elementwise division; both produce the array
```

```
# [[ 0.2          0.33333333]
```

```
# [ 0.42857143  0.5          ]]
```

```
print(x / y)
```

```
print(np.divide(x, y))
```

```
# Elementwise square root; produces the array
```

```
# [[ 1.          1.41421356]
```

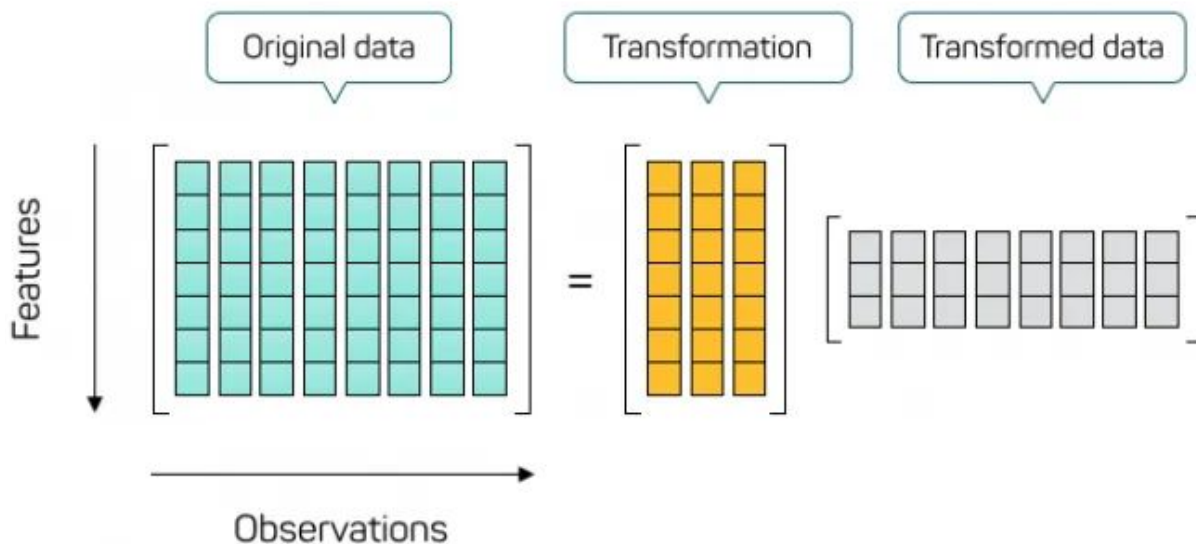
```
# [ 1.73205081  2.          ]]
```

```
print(np.sqrt(x))
```

Матричные разложения и задача снижения размерности методом главных компонент

Разложение матрицы

Разложение матрицы — представление матрицы A в виде произведения матриц, обладающих некоторыми определенными свойствами (например, ортогональностью, симметричностью, диагональностью).



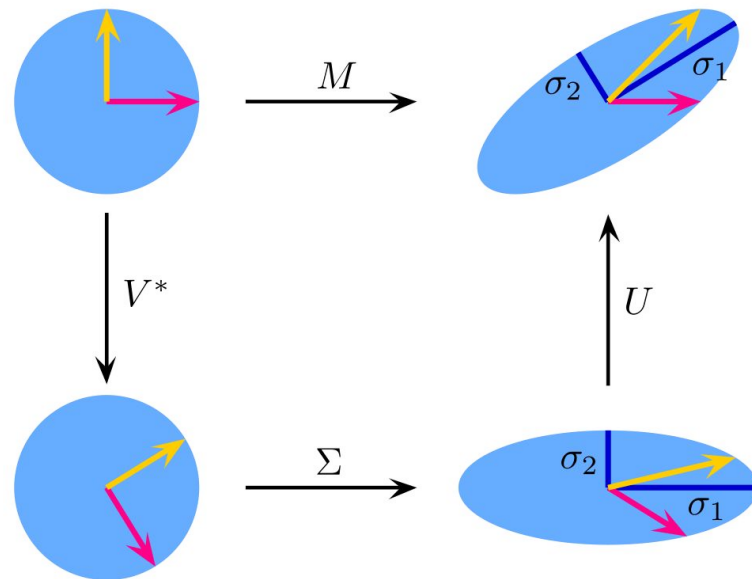
Сингулярное разложение матрицы

Сингулярным разложением матрицы M порядка $m \times n$ является разложение следующего вида

$$\begin{matrix} & n \\ m & \boxed{A} \end{matrix} = \begin{matrix} & m \\ m & \boxed{U} \end{matrix} \begin{matrix} & n \\ m & \boxed{\Sigma} \end{matrix} \begin{matrix} & n \\ \boxed{V^T} & n \end{matrix}$$

Геометрический смысл сингулярного разложения матрицы

Компоненты сингулярного разложения наглядно показывают геометрические изменения при отображении линейным оператором множества векторов из векторного пространства в себя или в векторное пространство другой размерности.



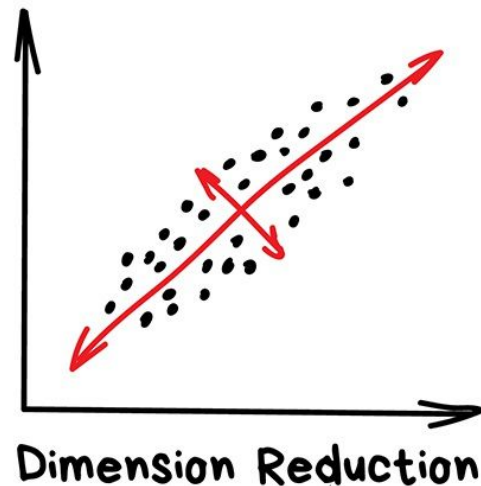
$$M = U \cdot \Sigma \cdot V^*$$

Задача снижения размерности

Снижение размерности - преобразование данных, состоящее в уменьшении числа признаков путем получения главных признаков: отбор признаков и выделение признаков.

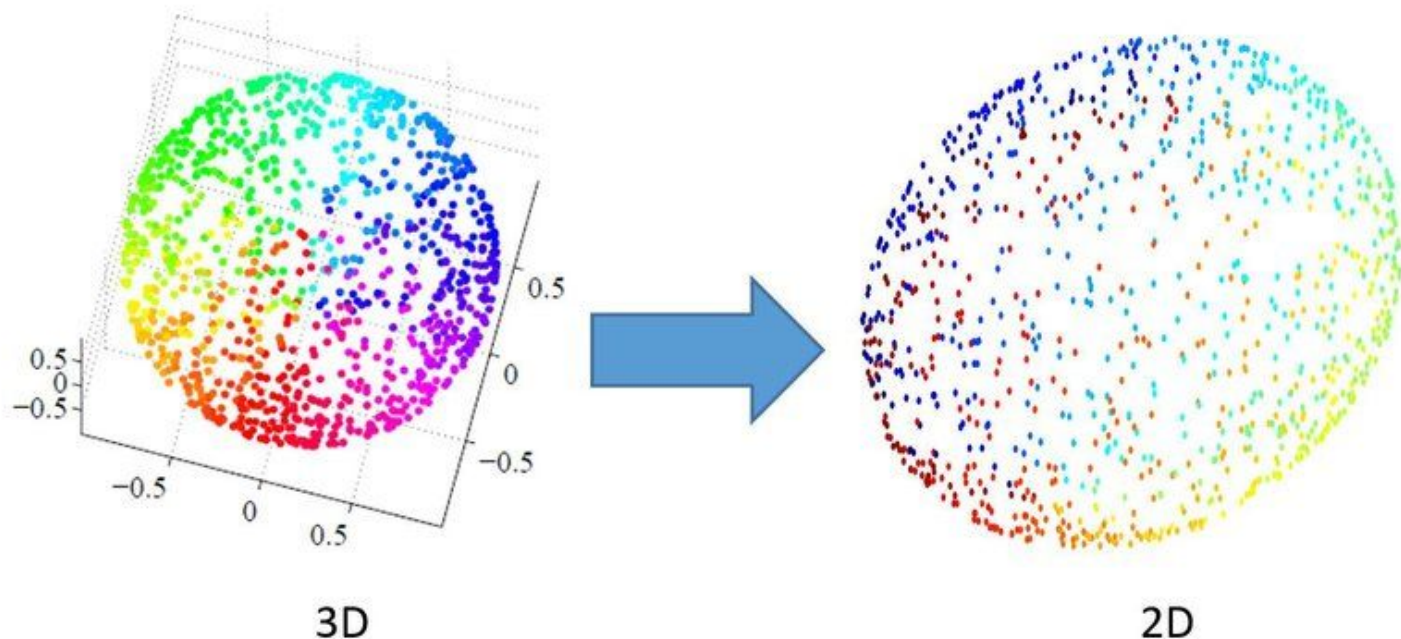
Сегодня используют для:

- Рекомендательные системы
- Красивые визуализации
- Определение тематики и поиска похожих документов
- Анализ фейковых изображений
- Риск-менеджмент



Практический смысл снижения размерности

Практическая польза **методов снижения размерности** в том, что мы можем объединить несколько признаков в один и получить абстракцию.



Метод главных компонент

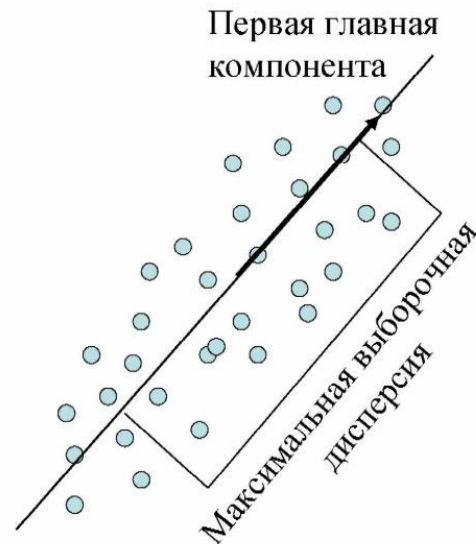
Одним из основных методов снижения размерности является **метод главных компонент (principal component analysis, PCA)**.

Первая главная компонента - направление наибольшей дисперсии.

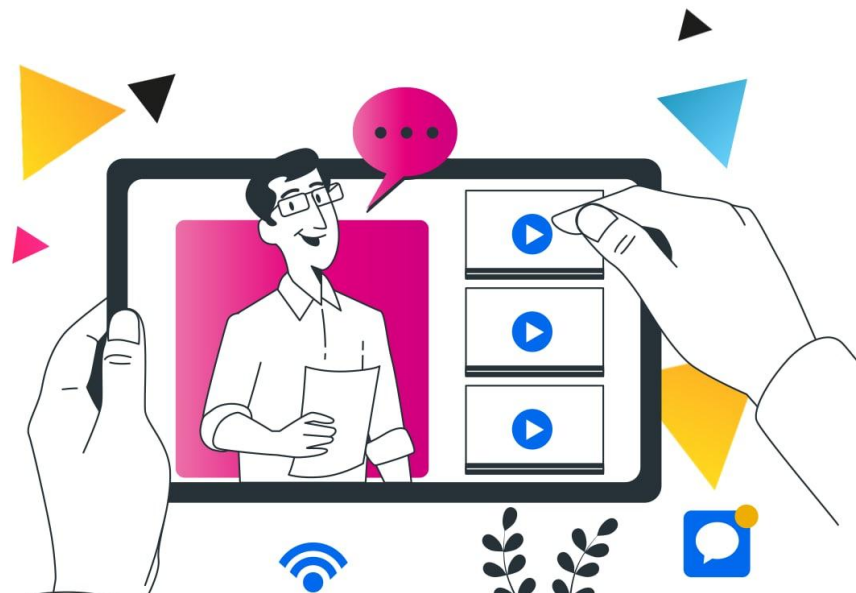
Вторая главная компонента - направление наибольшей дисперсии, при условии ортогональности по отношению к первой компоненте.

...

K-главная компонента - направление наибольшей дисперсии, при условии ортогональности по отношению к всем предыдущим компоненте.



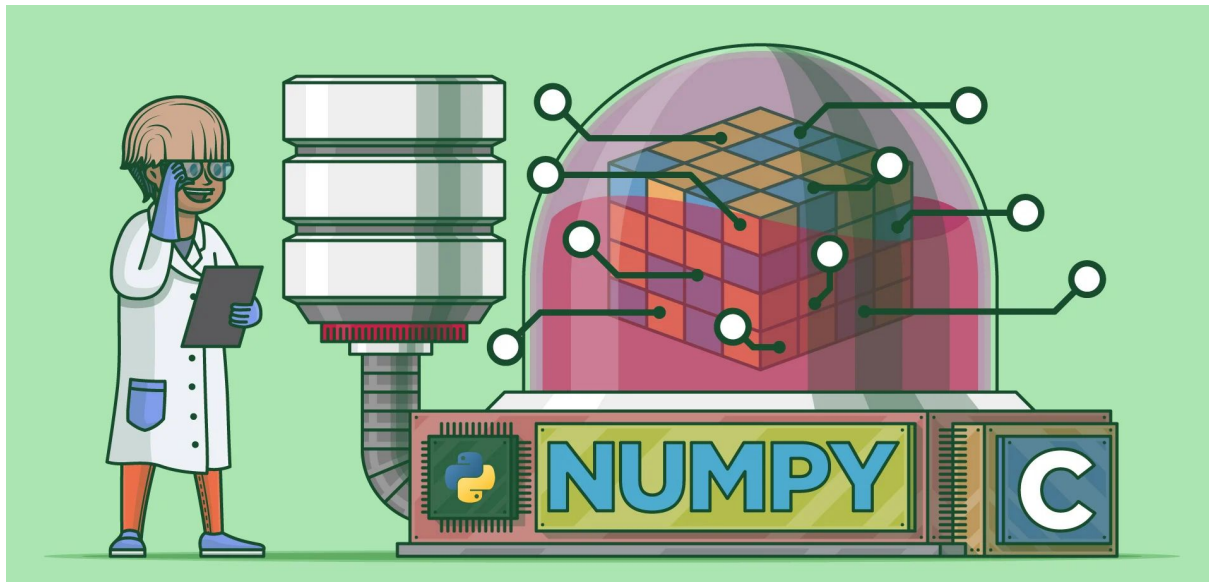
Ваши вопросы?



Практика

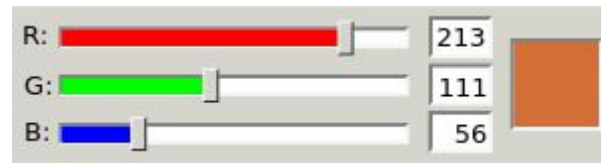
Задача 1

Простые упражнения на освоение основного функционала библиотеки NumPy. Задачи представлены в [ноутбуке](#).



Задача 2

Цвета часто представляются в компьютерах в виде трехмерных векторов: Red, Green и Blue (RGB). Другими словами, цвет выражается как триплет RGB (r, g, b) , каждый компонент которого может варьироваться от нуля до определенного максимального значения.



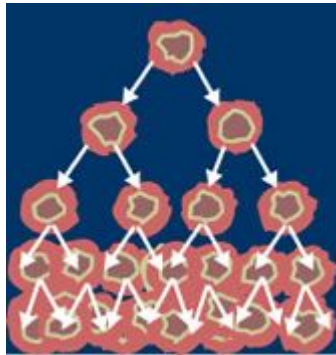
Если все компоненты равны нулю, результат черный. Если, напротив, все на максимуме, результатом будет самый яркий представляемый белый цвет.

В [этом ноутбуке](#) вы будете практиковать основные векторные операции с цветными векторами из [цветового обзора xkcd](#).

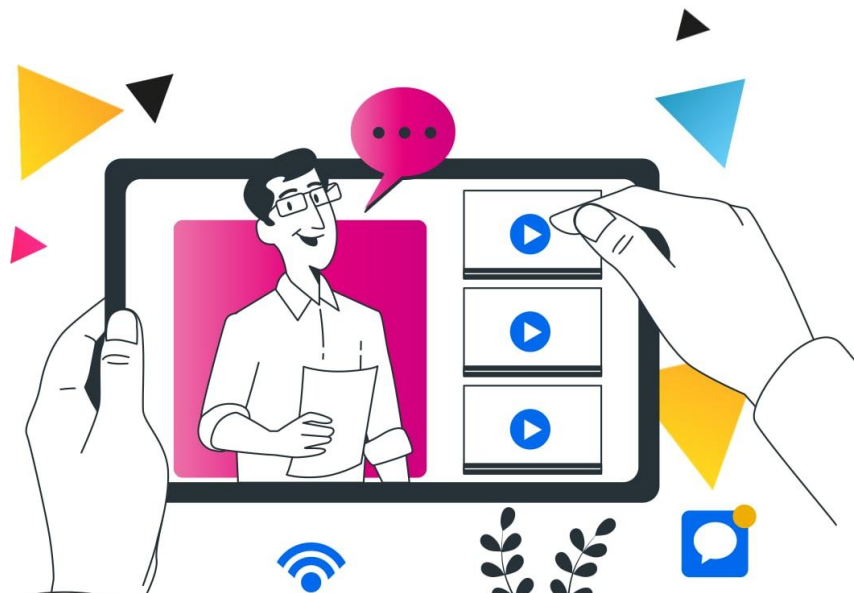
Задача 3

Для диагностики рака по цифрованному изображению тонкоигольной аспирации (ТАБ) массы молочной железы были рассчитаны характеристики ядер клеток, присутствующих на изображении.

Задача на снижение размерности методом главных компонент.



Ваши вопросы?



Итоги занятия

1. Повторили основы линейной алгебры
2. Познакомились с библиотекой NumPy и ее основным функционалом
3. Познакомились с применением линейной алгебры в NumPy
4. Научились осуществлять операции над векторами и матрицами с использованием NumPy

Дополнительные материалы по теме занятия



1. <https://pyprog.pro/>
2. https://colab.research.google.com/drive/1WB6KMyOI9BtN0ncjQFtVw4S6E7tFw56?usp=share_link

До встречи!

