

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Кафедра информатики

Лабораторная работа №7
«Метод главных компонент»

Выполнил: Чёрный Родион Павлович
магистрант кафедры информатики
группа №858642

Проверил: доцент кафедры информатики
Стержанов Максим Валерьевич

Минск 2019

Постановка задачи

Набор данных **ex7data1.mat** представляет собой файл формата *.mat (т.е. сохраненного из Matlab). Набор содержит две переменные X_1 и X_2 - координаты точек, для которых необходимо выделить главные компоненты.

Набор данных **ex7faces.mat** представляет собой файл формата *.mat (т.е. сохраненного из Matlab). Набор содержит 5000 изображений 32x32 в оттенках серого. Каждый пиксель представляет собой значение яркости (вещественное число). Каждое изображение сохранено в виде вектора из 1024 элементов. В результате загрузки набора данных должна быть получена матрица 5000x1024.

Задание.

1. Загрузите данные **ex7data1.mat** из файла.
2. Постройте график загруженного набора данных.
3. Реализуйте функцию вычисления матрицы ковариации данных.
4. Вычислите координаты собственных векторов для набора данных с помощью сингулярного разложения матрицы ковариации (разрешается использовать библиотечные реализации матричных разложений).
5. Постройте на графике из пункта 2 собственные векторы матрицы ковариации.
6. Реализуйте функцию проекции из пространства большей размерности в пространство меньшей размерности с помощью метода главных компонент.
7. Реализуйте функцию вычисления обратного преобразования.
8. Постройте график исходных точек и их проекций на пространство меньшей размерности (с линиями проекций).
9. Загрузите данные **ex7faces.mat** из файла.
10. Визуализируйте 100 случайных изображений из набора данных.
11. С помощью метода главных компонент вычислите собственные векторы.
12. Визуализируйте 36 главных компонент с наибольшей дисперсией.
13. Как изменилось качество выбранных изображений?
14. Визуализируйте 100 главных компонент с наибольшей дисперсией.
15. Как изменилось качество выбранных изображений?
16. Используйте изображение, сжатое в лабораторной работе №6 (Кластеризация).
17. С помощью метода главных компонент визуализируйте данное изображение в 3D и 2D.

18. Соответствует ли 2D изображение какой-либо из проекций в 3D?

Описание реализации

1. Загрузим данные из файла `ex7data1.mat` используя функцию `loadmat`:

```
mat = loadmat("ex7data1")  
x = mat['X']
```

2. График загруженного набора данных изображен на рисунке 1.

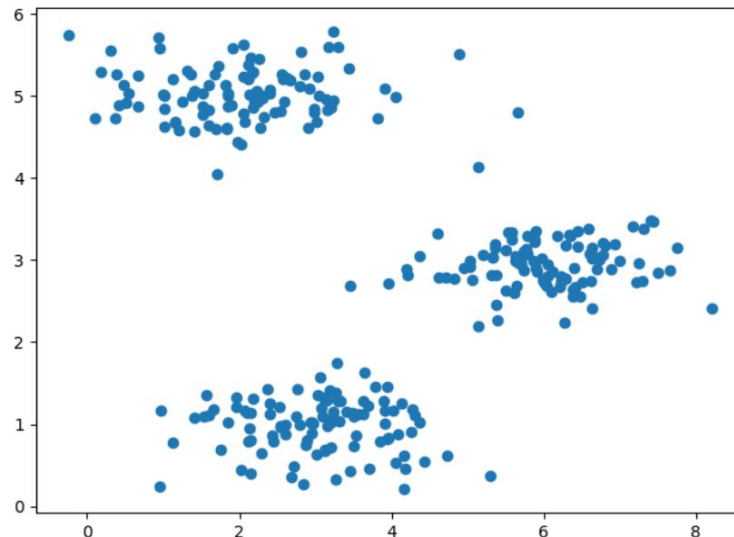


Рисунок 1 - График набора данных из файла `ex7data1.mat`

3. Реализация функции нахождения матрицы ковариации данных приведена ниже:

```
def compute_covariance_matrix(x):  
    m = len(x)  
    return (1 / m) * np.dot(x.T, x)
```

4. Найдем матрицу ковариации для исходного набора данных. Воспользуемся функцией `svd` из пакета `numpy.linalg` для сингулярного разложения матрицы. Получим значения собственных векторов:

```
u1 = [-0.7934064338206201, -0.6086922299258029]
```

```
u2 = [-0.608692229925803, 0.7934064338206199]
```

5. Построим график исходных данных совместно с собственными векторами матрицы ковариации. Результат изображен на рисунке 2.

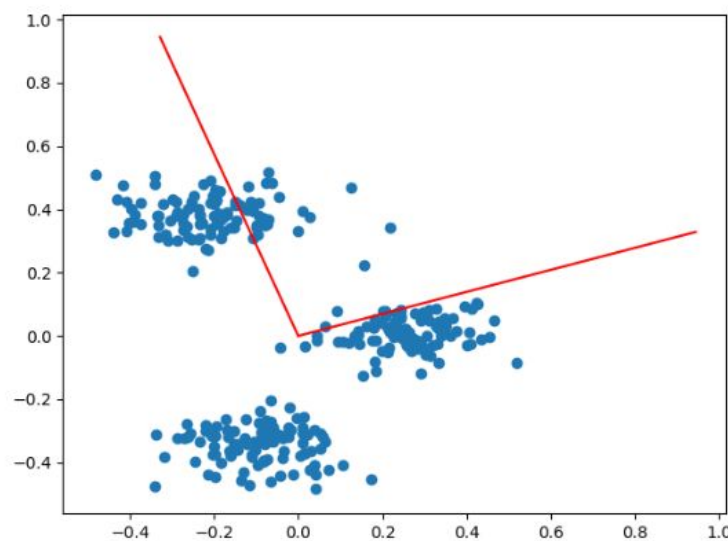


Рисунок 2 - Исходные данные совместно с собственными векторами матрицы ковариации

6. Реализация функции проекции из пространства большей размерности в пространство меньшей (k) размерности с помощью метода главных компонент:

```
def project_data(k, u, x):
    u = u[:, :k]
    z = np.dot(x, u)
    return z
```

7. Реализуем функцию вычисления обратного преобразования:

```
def recover_data(z, u, k):
    return z.dot(u[:, :k].T)
```

8. График исходных точек и их проекций на пространство меньшей размерности (с линиями проекций) изображен на рисунке 3.

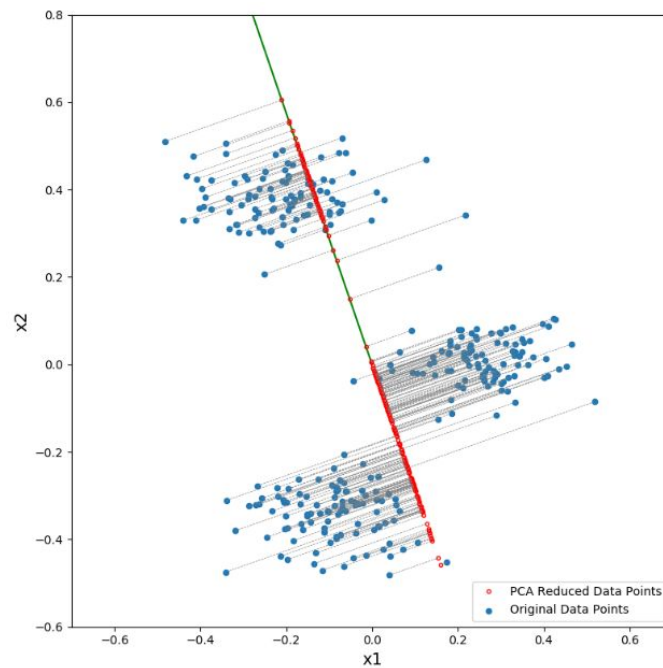


Рисунок 3 - График исходных точек и их проекций на пространство размерности 1

9. Загрузим данные из файла ex7faces.mat:

```
mat = loadmat('ex7faces.mat')
x = mat['X']
```

10. Визуализируем 100 случайных изображений из набора данных. Сетка 100 картинок изображена на рисунке 4.



Рисунок 4 - 100 случайных изображений из ex7faces.mat

11. Вычислим собственные векторы для набора данных.

```
mean = np.mean(x)
x -= mean
sigma = compute_covariance_matrix(x)
u, s, v = np.linalg.svd(sigma)
```

12. Визуализируем 36 главных компонент с наибольшей дисперсией. Сетка изображена на рисунке 5.

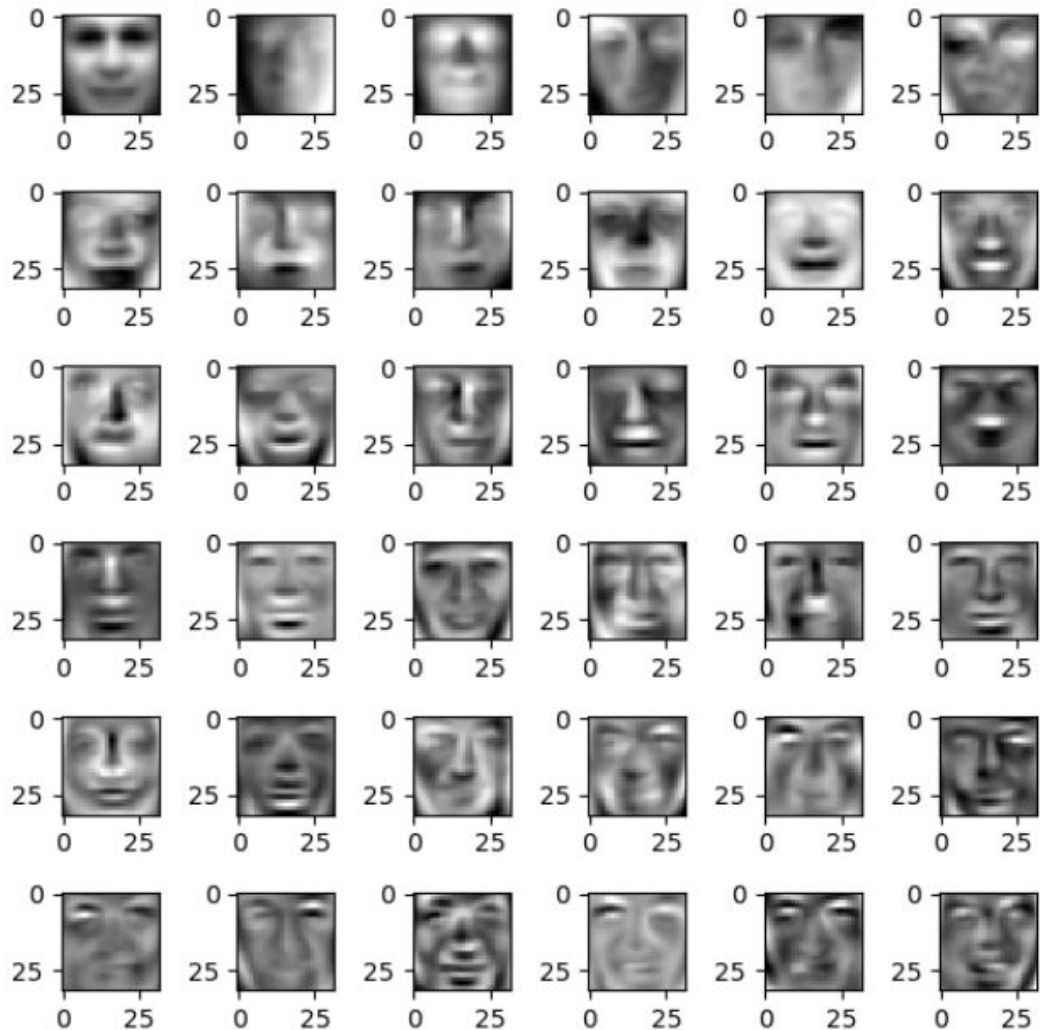


Рисунок 5 - 36 главных компонент с наибольшей дисперсией

13. В изображениях с наибольшей дисперсией можно разглядеть общий план человеческого лица. Но чем меньше дисперсия, тем ярнее проявляются какие-то специфичные черты.

14. Визуализируем 100 главных компонент с наибольшей дисперсией.
Сетка изображена на рисунке 6.

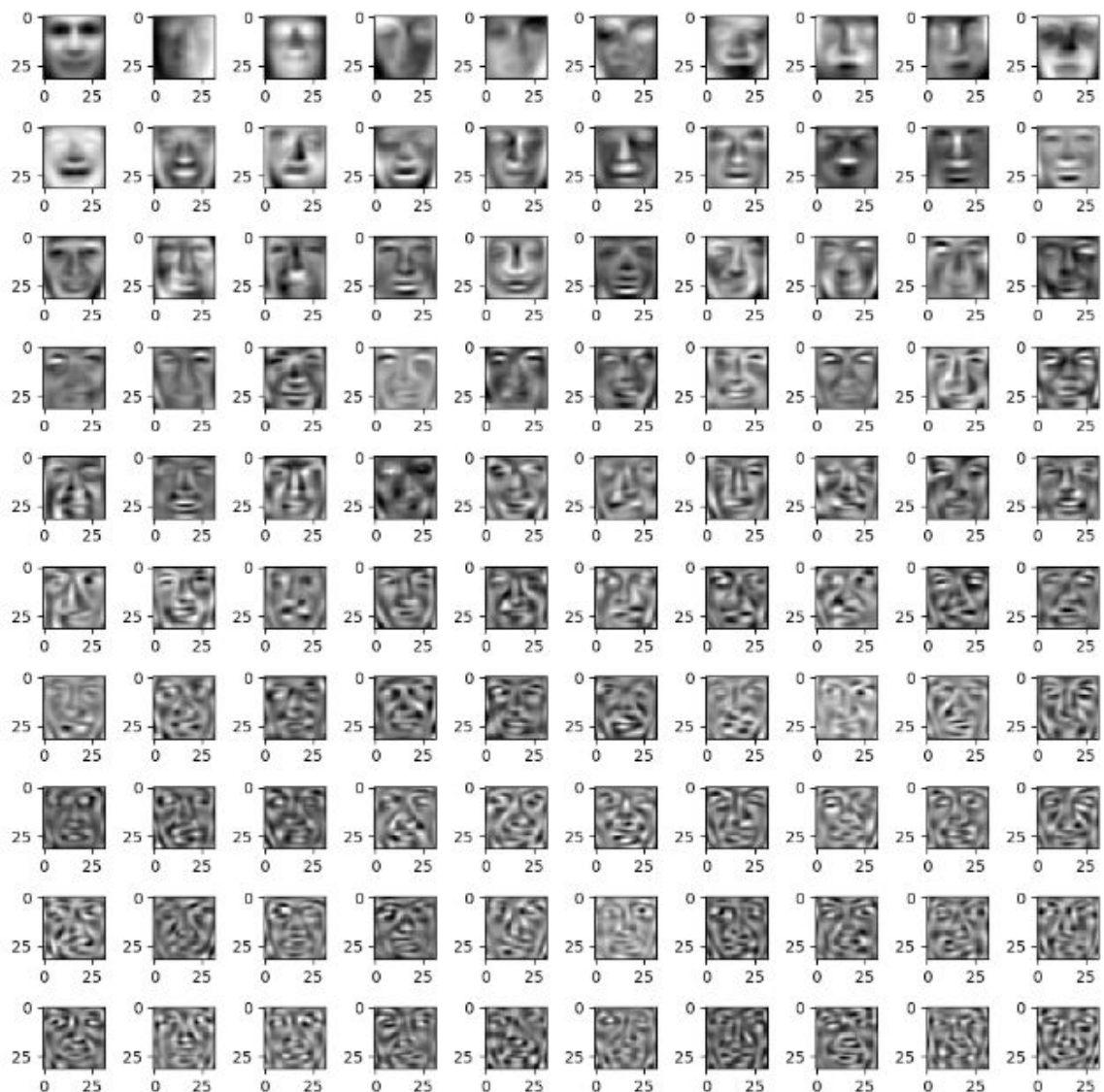


Рисунок 6 - 100 главных компонент с наибольшей дисперсией

15. При выводе большого числа изображений разница между компонентой с большой дисперсией и малой становится более заметна: при малой дисперсии изображения содержат больше специфичных деталей.

16. Загрузим сжатое изображение птицы из прошлой лабораторной работы:

```
A = imageio.imread("bird_16.png")  
X = A.reshape(-1, 3).astype(np.float64)
```

17. Визуализируем точки в трехмерном пространстве цветов на двумерную плоскость. Трехмерный вариант изображен на рисунке 7, а двумерная проекция - на рисунке 8.

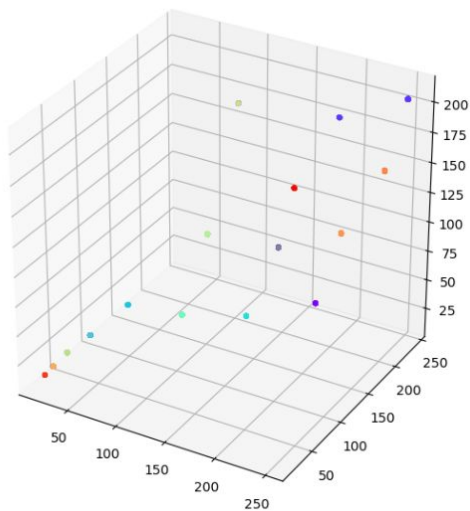


Рисунок 7 - Визуализация пикселей в трехмерном пространстве

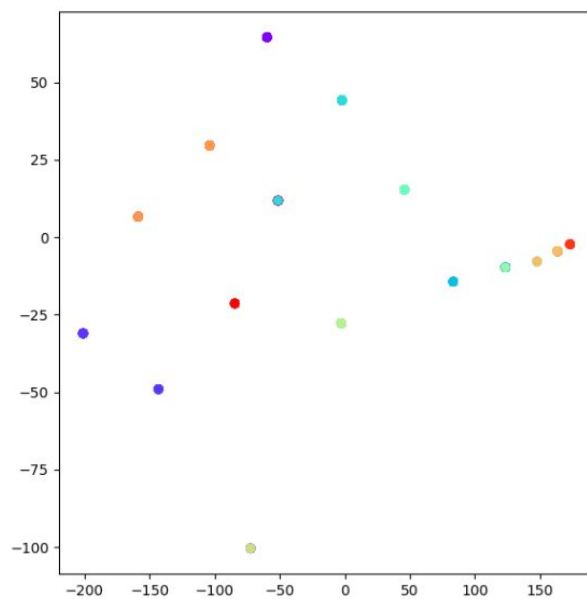


Рисунок 8 - Проекция пикселей на двумерную плоскость