Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных сетей и систем

Кафедра Информатики

# ЛАБОРАТОРНАЯ РАБОТА №7
«Метод главных компонент»

БГУИР 1-40 81 04

Магистрант:
гр. 858642
Кукареко А.В.

Проверил:
Стержанов М. В.

Минск, 2019

## ХОД РАБОТЫ

### Задание.

Набор данных ex7data1.mat представляет собой файл формата *.mat (т.е. сохраненного из Matlab). Набор содержит две переменные X1 и X2 - 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.

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?
19. Ответы на вопросы представьте в виде отчета.

**Результат выполнения:**

1. Загрузите данные ex7data1.mat из файла.

```
data1 = scipy.io.loadmat('ex7data1.mat')
X1 = data1['X']
X1.shape
```

```
(300, 2)
```

2. Постройте график загруженного набора данных.

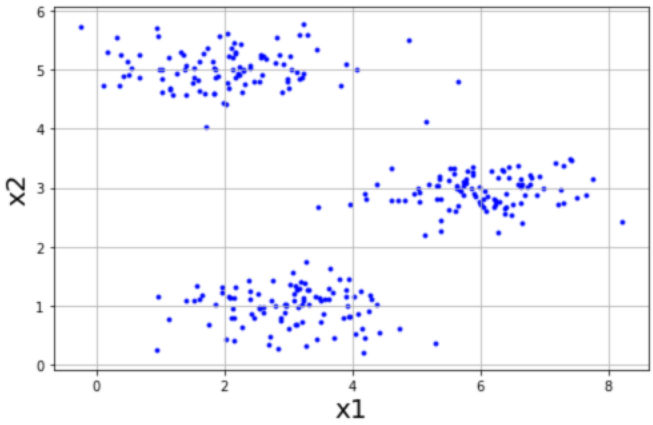


Рисунок 1 – исходные данные файла ex7data1.mat.

3. Реализуйте функцию вычисления матрицы ковариации данных.

Функция вычисления матрицы ковариации данных:

```
def calc_sigma(X):
    m = len(X)

    return ( 1 / m) * np.dot(X.T, X)
```

Результат работы функции:

```
norm_X1, norm_mean, norm_std = normalization(X1)
Sigma = calc_sigma(norm_X1)

print(f'X.shape = {X1.shape}')
print(f'Sigma.shape = {Sigma.shape}')
```

```
X.shape = (300, 2)
Sigma.shape = (2, 2)
```

4. Вычислите координаты собственных векторов для набора данных с помощью сингулярного разложения матрицы ковариации (разрешается использовать библиотечные реализации матричных разложений).

Для вычисления координаты собственных векторов с помощью с помощью сингулярного разложения матрицы ковариации была использована библиотека «scipy.linalg».

```
U, S, _ = linalg.svd(Sigma)
```

```
U.shape
```

```
(2, 2)
```

5. Постройте на графике из пункта 2 собственные векторы матрицы ковариации.

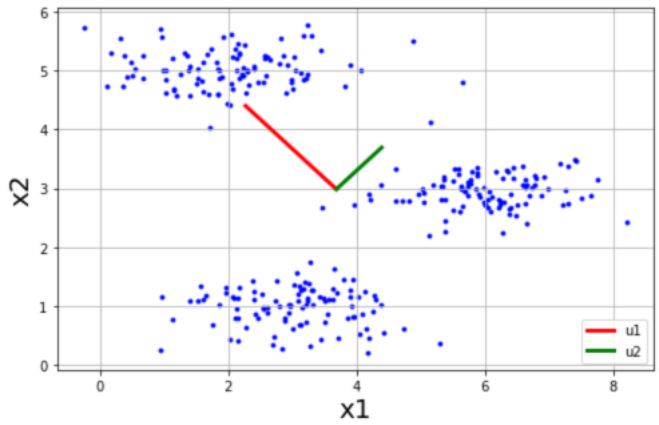


Рисунок 2 – исходные данные файла ex7data1.mat и векторы матрицы ковариации.

6. Реализуйте функцию проекции из пространства большей размерности в пространство меньшей размерности с помощью метода главных компонент.

Функция проекции из пространства большей размерности в пространство меньшей размерности:

```
def run_pca(X_norm, k):
    Sigma = calc_sigma(X_norm)
    U, S, _ = linalg.svd(Sigma)
    U_red = get_k_vectors(U, k)

    return np.dot(X_norm, U_red), U_red, S
```

Вспомогательные функции:

```
def get_k_vectors(U, k):
    return U[:, 0 : k]
```

```
def calc_dispersion(S, K):
    return np.sum(S[0: K])/np.sum(S)
```

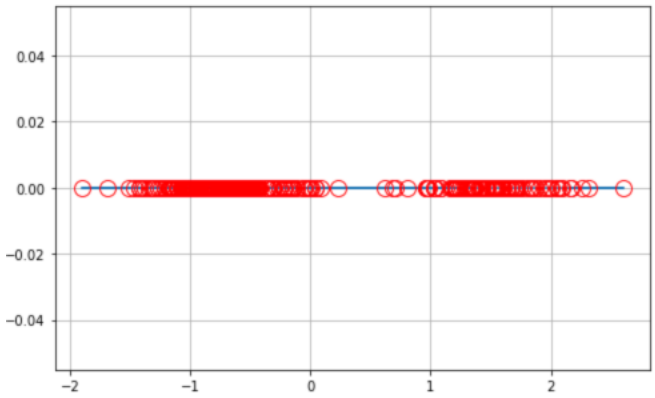


Рисунок 3 – график проекции из пространства большей размерности (2D) в пространство меньшей размерности (1D).

7. Реализуйте функцию вычисления обратного преобразования.

Функция обратного преобразования:

```
def pca_revert(Z, U_red):
    return np.dot(Z, U_red.T)
```

7. Постройте график исходных точек и их проекций на пространство меньшей размерности (с линиями проекций).

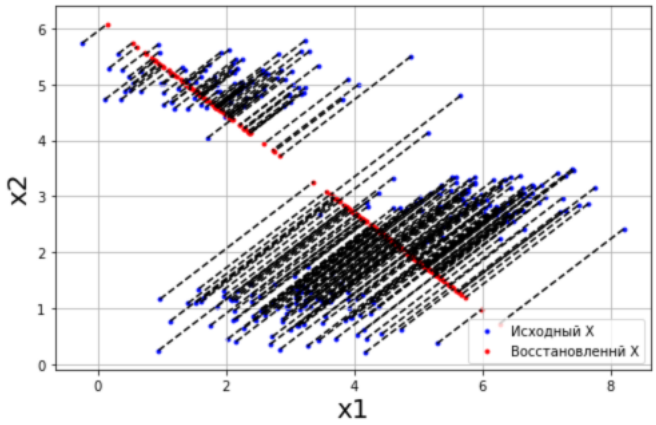


Рисунок 4 – график исходных точек и их проекций на пространство меньшей размерности.

9. Загрузите данные ex7faces.mat из файла.

```
faces_data = scipy.io.loadmat('ex7faces.mat')
faces_X = faces_data['X']
faces_X.shape
```

```
(5000, 1024)
```

10. Визуализируйте 100 случайных изображений из набора данных.

Рисунок 5 – визуализация 100 случайных изображений.

11. С помощью метода главных компонент вычислите собственные векторы.

```
Z_36, U_red_36, S_36 = run_pca(faces_norm_X, 36)
```

12. Визуализируйте 36 главных компонент с наибольшей дисперсией.

Рисунок 6 – визуализация 36 главных компонент с наибольшей дисперсией.

Рисунок 7 – восстановленные данные из 36 главных компонент.

13. Как изменилось качество выбранных изображений?

При 36 главных компонентах «сохранность дисперсии» составляет 83.12%. Если оценить качество картинок визуально, то оно значительно ухудшилось. При этом в визуализации главных компонент видны мягкие очертания лиц.

14. Визуализируйте 100 главных компонент с наибольшей дисперсией.

Рисунок 8 – визуализация 100 главных компонент с наибольшей дисперсией.

Рисунок 9 – восстановленные данные из 100 главных компонент.

15. Как изменилось качество выбранных изображений?

При 100 главных компонентах «сохранность дисперсии» составляет 93.19%. Если оценить качество картинок визуально, гораздо лучше, чем при 36. При этом в визуализации главных компонент имеет более сложные "узоры" чем у 36 главных компонент.

16. Используйте изображение, сжатое в лабораторной работе №6 (Кластеризация).

```
bird_data = scipy.io.loadmat('bird_small.mat')
Xb = bird_data['A'].reshape(-1, 3)
bird_norm_X, bird_norm_mean, bird_norm_std = normalization(Xb)
bird_norm_X.shape
```

```
bird_k = 16
cluster = KMeans(n_clusters=bird_k, random_state=0)
cluster.fit(bird_norm_X)
```

17. С помощью метода главных компонент визуализируйте данное изображение в 3D и 2D.

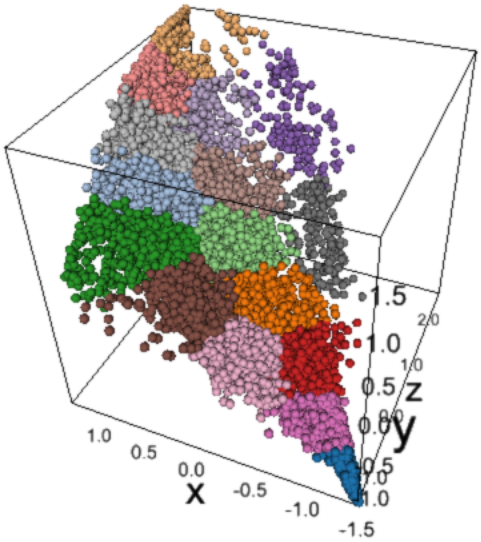


Рисунок 10 – визуализация картинки в 3D пространстве по кластерам.

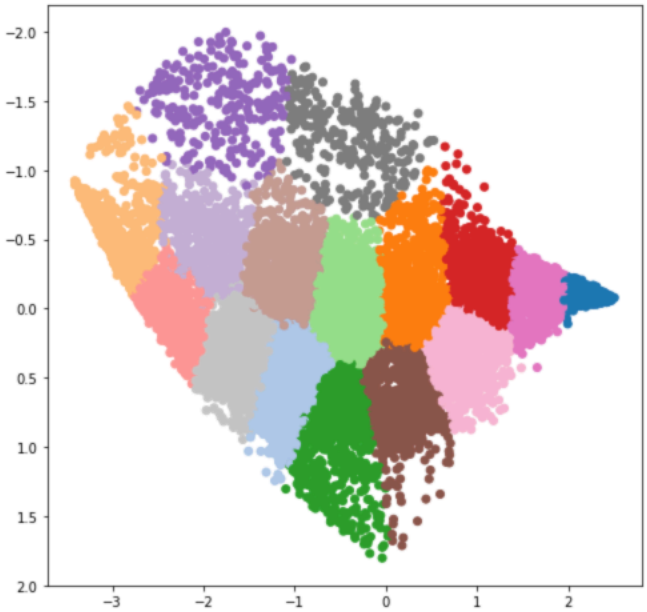


Рисунок 11 – 2D проекция картинки полученная с помощью метода главных компонент.

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

Если сравнить рисунок 10 и рисунок 11, то сходства видны невооруженным глазом. Этот пример показывает, что алгоритм главных компонент работает корректно.

**Вывод.**

В ходе выполнения лабораторной работы я ознакомился с методом главных компонент, который применяется для уменьшения размерности векторов «фич». Так же в данной работе я реализовал алгоритм на практике и убедился в его работоспособности и удобности его применения как на лабораторных данных так и на примере картинки.