Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра информатики

Отчет по лабораторной работе:

**Лабораторная работа №7 “ Метод главных компонент”**

Выполнил: Карп Александр Игоревич

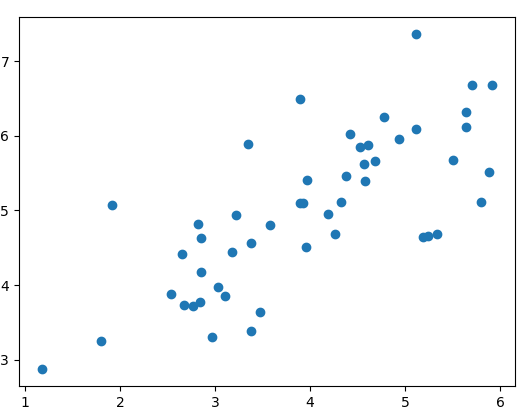
магистрант кафедры информатики

группа №858641

Минск 2019

1. Загрузите данные ex7data1.mat из файла.

#1  
data = sio.loadmat('ex7data1.mat')  
X = data.get('X')  
plt.scatter(X [:,0], X [:,1])  
plt.show()

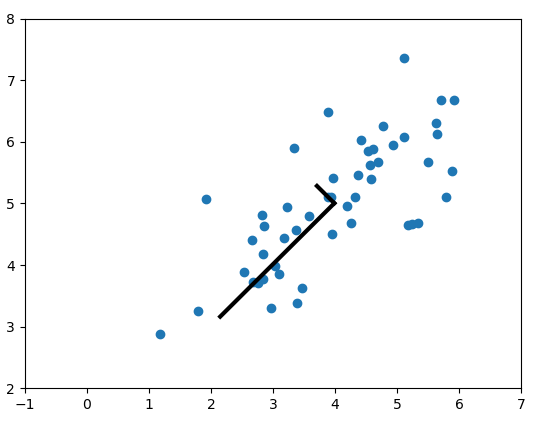
1. Постройте график загруженного набора данных
2. Реализуйте функцию вычисления матрицы ковариации данных.

def cov(X):  
 return X.T.dot(X) / X.shape[0]

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

def pca(X):  
 U, S, V = np.linalg.svd(cov(X))  
 return U, S, V

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



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

def projectData(X, U, K):  
 U\_reduced = U[:, :K]  
 Z= np.dot(X, U\_reduced)  
 return Z

Проекция первого элемента в одномерное пространство: 1.496

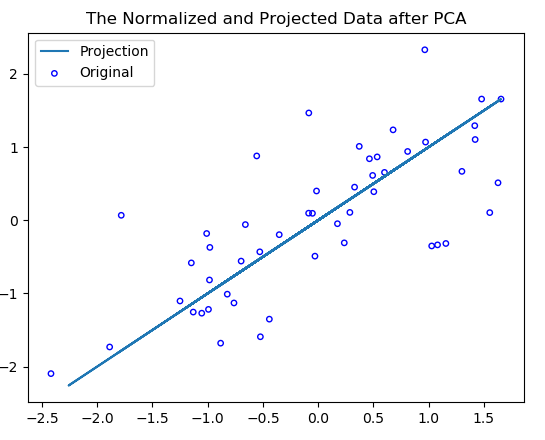
1. Реализуйте функцию вычисления обратного преобразования

def recoverData(Z, U, K):  
 U\_reduced = U[:, :K]  
 X\_rec =np.dot( Z , U\_reduced.T)  
 return X\_rec

Обратное преобразование первого элемента:

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

plt.scatter(X\_norm[:,0],X\_norm[:,1],marker="o",label="Original",facecolors="none",edgecolors="b",s=15)  
plt.plot(X\_rec[:,0],X\_rec[:,1], label="Projection")  
plt.title("The Normalized and Projected Data after PCA")  
plt.legend()  
plt.show()



1. Загрузите данные ex7faces.mat из файла.

data = sio.loadmat('ex7faces.mat')  
X = data.get('X')

1. Визуализируйте 100 случайных изображений из набора данных.

fig, ax = plt.subplots(nrows=10,ncols=10,figsize=(8,8))  
indexes = np.random.randint(0, 5000, 100)  
axis = 0  
for i in indexes:  
 for j in range(10):  
 ax[int(axis/10),j].imshow(X[i+j,:].reshape(32,32,order="F"),cmap="gray")  
 ax[int(axis/10),j].axis("off")  
 axis +=1



1. С помощью метода главных компонент вычислите собственные векторы.

U =pca(X\_norm)[0]

1. Визуализируйте 36 главных компонент с наибольшей дисперсией.
2. U\_reduced = U[:,:36].T  
   fig2, ax2 = plt.subplots(6,6,figsize=(8,8))  
   for i in range(0,36,6):  
    for j in range(6):  
    ax2[int(i/6),j].imshow(U\_reduced[i+j,:].reshape(32,32,order="F"),cmap="gray")  
    ax2[int(i/6),j].axis("off")



13. Как изменилось качество выбранных изображений?

Видно, что качество изображений ухудшилось. С уменьшением дисперсии можно заметить, что детализация картинок растет.

1. Визуализируйте 100 главных компонент с наибольшей дисперсией.

U\_reduced = U[:,:100].T  
fig2, ax2 = plt.subplots(10,10,figsize=(8,8))  
for i in range(0,100,10):  
 for j in range(10):  
 ax2[int(i/10),j].imshow(U\_reduced[i+j,:].reshape(32,32,order="F"),cmap="gray")  
 ax2[int(i/10),j].axis("off")  
plt.show()



15. Как изменилось качество выбранных изображений?

Можно заметить, что с уменьшением дисперсии детализация изображений увеличивается

**Выводы**

Метод главных компонент — один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации.

Недостаток метода заключается в том, что метод главных компонент

является очень чувствительным к входным данным.