PCA

GOAL

The goal of this practice is to understand the PCA algorithm and get some intuition about what is it doing and when to use it. You will have to do the computations by hand to learn the maths behind it and you will also see how to apply it in a faster and more efficient way using the Sklearn library.

EXERCISES

1. Suppose that we have a set of 2D samples: [0, 0], [1, 1], [2, 3], [3, 2] i [4, 4]. Answer the following questions doing all the computations manually.

   a) Draw the data and the eigenvectors that you think that the covariance matrix of the data will have.

   b) Without doing any computation, draw in the figure from the previous exercise the projection of the points in the component that explains more variance. Show also for every point the information loss after the projection.

   c) Compute the covariance matrix of the data and find the eigenvalues and eigenvectors. (for now we are going to keep both components). You can use the numpy function numpy.linalg.eig() (find it in the ipython notebook “P4.ipynb”).

   d) Project the data over the basis obtained with PCA. Discuss the relation existing between the covariance of the projected data and the eigenvalues computed in the previous exercise.

   e) Choose the component that maximizes the variance of the data. Project and re-project the data using the chosen component. Draw the results.

   f) Project and re-project the point x = [2, 4] using only the component with maximum variance. Draw the results and compute the reprojection error.

   Solution

   a) and b)
   See picture in the next page.

   c)

   We start computing the covariance matrix:

$$X' = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 3 & 2 & 4 \end{pmatrix}, \qquad \mu = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \rightarrow X = X_{raw} - \mu = \begin{pmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 1 & 0 & 2 \end{pmatrix},$$

$$\Sigma = \frac{1}{N} X X^T = \begin{pmatrix} 2 & 1.8 \\ 1.8 & 2 \end{pmatrix}$$

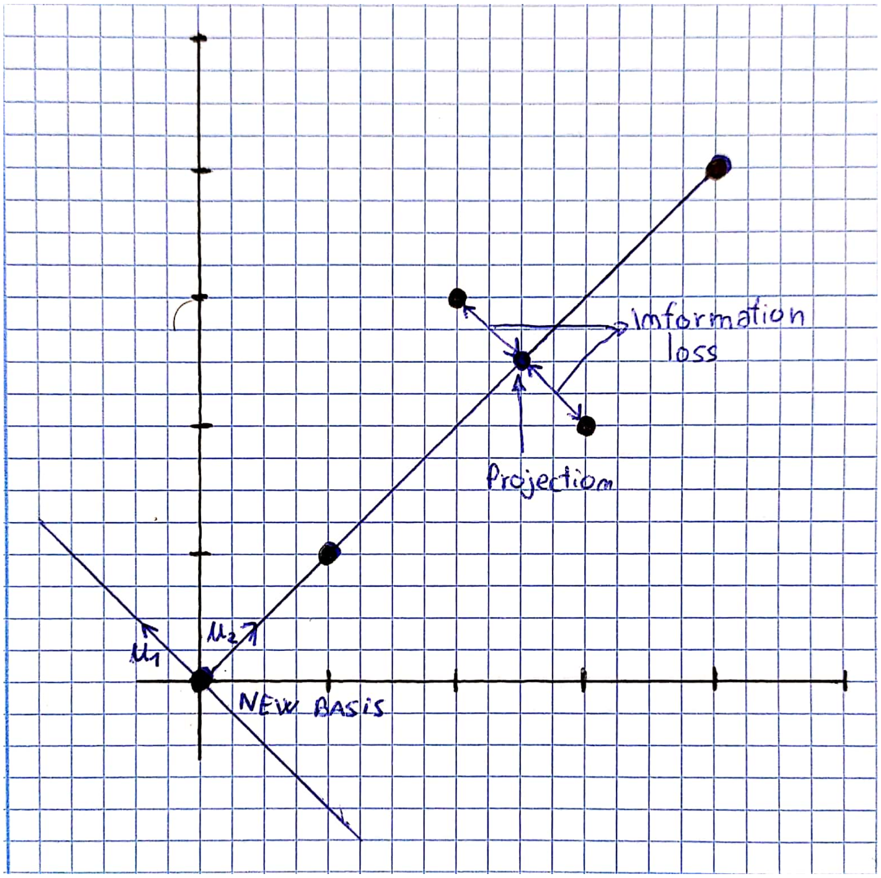
   The corresponding eigenvectors are

$$U = \text{eig}(\Sigma) = \begin{pmatrix} -0.7 & 0.7 \\ 0.7 & 0.7 \end{pmatrix},$$

   where each column corresponds to one component. The eigenvalues are $\lambda_1 = 0.2$ and $\lambda_2 = 3.8$.

   d)

$$X_{proj} = U^T X = \begin{pmatrix} 0 & 0 & 0.7 & -0.7 & 0 \\ -2.8 & -1.4 & 0.7 & 0.7 & 2.8 \end{pmatrix}$$


Information
loss
Projection
u_1
u_2
NEW BASIS

e)

Since $\lambda_1 < \lambda_2$, we choose $u_2 = (0.7, 0.7)^T$ Then,

$$X_{proj} = u_2^T X = \begin{pmatrix} -2.8 & -1.4 & 0.7 & -0.7 & 2.8 \end{pmatrix}$$

$$X_{reproj} = u_2 X_{proj} = \begin{pmatrix} -2 & -1 & 0.5 & 0.5 & 2 \\ -2 & -1 & 0.5 & 0.5 & 2 \end{pmatrix}$$

$$X'_{reproj} = X'_{reproj} + \mu = \begin{pmatrix} 0 & 1 & 2.5 & 2.5 & 4 \; 0 & 1 & 2.5 & 2.5 & 4 \end{pmatrix}$$

f)

$$X_{proj} = U^T x = 4.2$$

$$X_{reproj} = U x_{proj} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$

$$X'_{reproj} = x'_{reproj} + \mu = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$$

2. (In Jupyter Notebook) We are now going to use the sklearn library to apply pca to the dataset from the last exercise. In concrete, open the ipython notebook called “P4.ipynb” and using the class pca.decomposition.PCA, you have to:

   a) Show the basis found using PCA (these are the eigenvectors that you found in the previous exerciese).

   b) Find the variance associated to each component of the basis, which corresponds to the eigenvalues from the previous exercise.

   c) Show the original data projected in the new basis.

   d) Project the point x=[2,3] to the component with maximum variance.

3. (In Jupyter Notebook) In this exercise we are going to use sklearn with a real dataset, to see how can PCA help us. We are going to use the Iris dataset, which is perhaps the best known database to be found in the pattern recognition literature. The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant, which are Iris Setosa, Iris Versicolour and Iris Virginica. The dataset has 4 features, which are sepal length, sepal width, petal length, and petal width, all in cm. The first problem that we have if we want to work with this dataset is that we are not able to visualize data in four dimensions. In this practice we are going to use PCA to reduce the dimensionality of the dataset so we can visualize it and analyze it better. Open the ipython notebook “P4.ipynb” and follow the instructions for this exercise.