

# Worksheet

## Part 1. Objectives

- **Long-term goal:** Understand the nature of eigenvectors and eigenvalues
- Understand how to apply Power Iteration and its variants
- See an application of eigenvector extraction

## Part 2. Inverse Iteration with a Shift

Suppose  $A$  is an  $4 \times 4$  matrix with the eigenvalues  $-8, -6, 5,$  and  $1$ . Suppose you apply the following iterative algorithm:  $x_{k+1} = \frac{(A-\sigma I)^{-1}x_k}{\|(A-\sigma I)^{-1}x_k\|}$  and choose  $\sigma = 4$ . Assuming that the algorithm converges on an eigenvector, which eigenvalue of  $A$  is associated with that eigenvector?

## Part 3. Principal Component Analysis (PCA)

Informally, if you have a bunch of scattered data, PCA will let you determine a vector for the direction of greatest spread and the direction of least spread. These directions can be found by finding the eigenvectors and eigenvalues of the covariance matrix for the data. In this problem, you are given a scattered 2D point set with the x coordinates stored in  $\mathbf{x}$  and the y coordinates in  $\mathbf{y}$ .

In 2D, the sample covariance matrix for a data set of  $n$  points  $(x_i, y_i)$  can be formed in the following manner:  $\frac{1}{n-1} \begin{bmatrix} (x_i - \mu_x)^2 & (x_i - \mu_x)(y_i - \mu_y) \\ (y_i - \mu_y)(x_i - \mu_x) & (y_i - \mu_y)^2 \end{bmatrix}$  where  $\mu_x$  is the average of the x coordinates and similarly  $\mu_y$  for y.

The sample covariance matrix is symmetric, so it will have  $n$  orthogonal eigenvectors. The dominant eigenvector (the one associated with the eigenvalue with the greatest absolute value) will be the principal direction of the data set and the eigenvector associated with the eigenvalue of least magnitude will be the direction of least spread. These eigenvectors form the principal directions of the data.

One can compute an object-aligned bounding box by projecting the data points onto the normalized versions of the two principal eigenvectors. Specifically, use the centroid as the origin of the box with the principal eigenvectors forming the local coordinate system. Project the vectors from the centroid to the scattered points onto the normalized eigenvectors and keep track of the maximum and minimum extents. Use those extents to find the corners of the bounding box. Put the corners of the bounding box into a numpy array named `corners`. The order of the corners is important (purely for grading reasons). If  $e_1$  is the dominant eigenvector and  $e_2$  the other eigenvector, order the points so that the first point is: the minimum in directions  $e_1$  and  $e_2$ , the second is the minimum

$e_1$  and maximum  $e_2$ , the third is maximum  $e_1$  and maximum  $e_2$ , and the last is maximum  $e_1$  and minimum  $e_2$ .

*Hints:*

- The eigenvectors can be extracted using the Python function `numpy.linalg.eig`.

INPUTS: `x` and `y`

OUTPUT: `corners`

```
import numpy as np
import numpy.linalg as la
import matplotlib.pyplot as plt
import matplotlib.lines as lines

# Compute the centroid of the points
xm = x.mean()
ym = y.mean()

# Create the covariance matrix
cv_mat = np.zeros((2,2))
for i in range(n):
    cv_mat[0][0]+=((x[i]-xm)**2)
    cv_mat[1][1]+=((y[i]-ym)**2)
    cv_mat[0][1]+=((y[i]-ym)*(x[i]-xm))
    cv_mat[1][0]+=((x[i]-xm)*(y[i]-ym))

cv_mat/=(n-1)
# Complete the code by finding the eigenvectors
# ...and computing the bounding box
# Put your code here

# End of your code
fig = plt.figure()
ax = fig.add_subplot(111,aspect="equal")
draw_box(corners,ax)
plt.plot(x, y, "o",color="red")
plt.show()
```