Homework 9 Calculation and Plotting

May 26, 2020

[1]:
```
import numpy as np
import scipy.linalg
import matplotlib.pyplot as plt
```

[2]:
```
A1 = np.asarray([[2.0,2.0],
                         [-1.0,2.0],
                         [1.0,1.0],
                         [-2.0,3.0],
                        ])
```

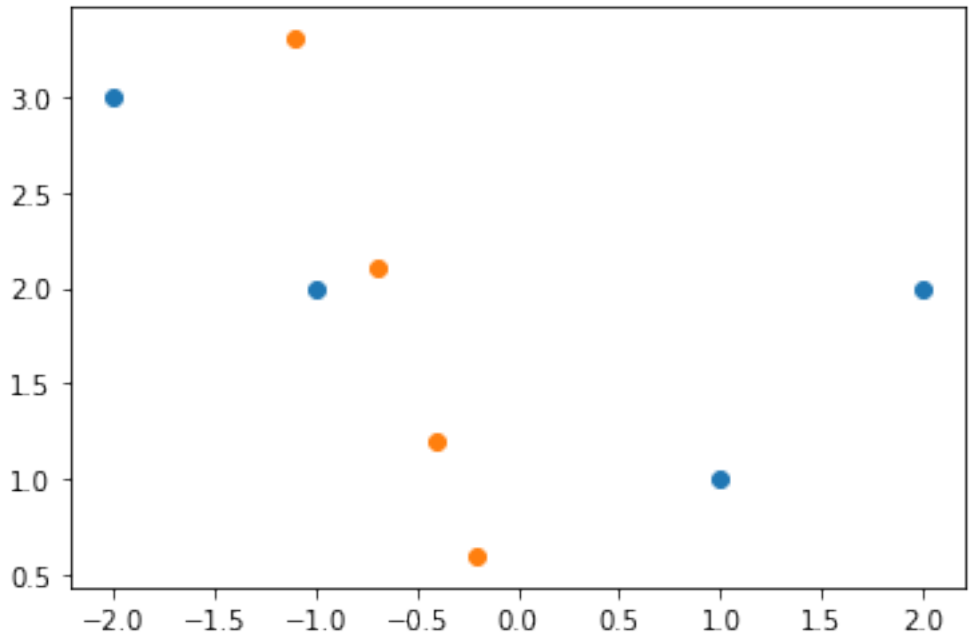
[3]:
```
class P1:
    def solve(A):
        AtA = A.T@A
        n = AtA.shape[0]
        plt.scatter(A[:,0], A[:,1])
        plt.show()
        print(np.poly(AtA))
        print(np.roots(np.poly(AtA)))
        roots = np.roots(np.poly(AtA))
        r = len(roots)
        I = np.eye(n)

        for root in roots:
            Ei = AtA-root*I
            print("eiggen vectors of eigen value " + str(root))
            ns = scipy.linalg.null_space(Ei)
        # vs have to self calc.

        print(np.linalg.eig(AtA))
```

[4]:
```
U1 = np.array([[4,7,2,11],
               [8, -1, 4, -3],
               [0, 0, 0, 0],
               [0, 0, 0, 0]
              ])
scipy.linalg.null_space(U1)
```

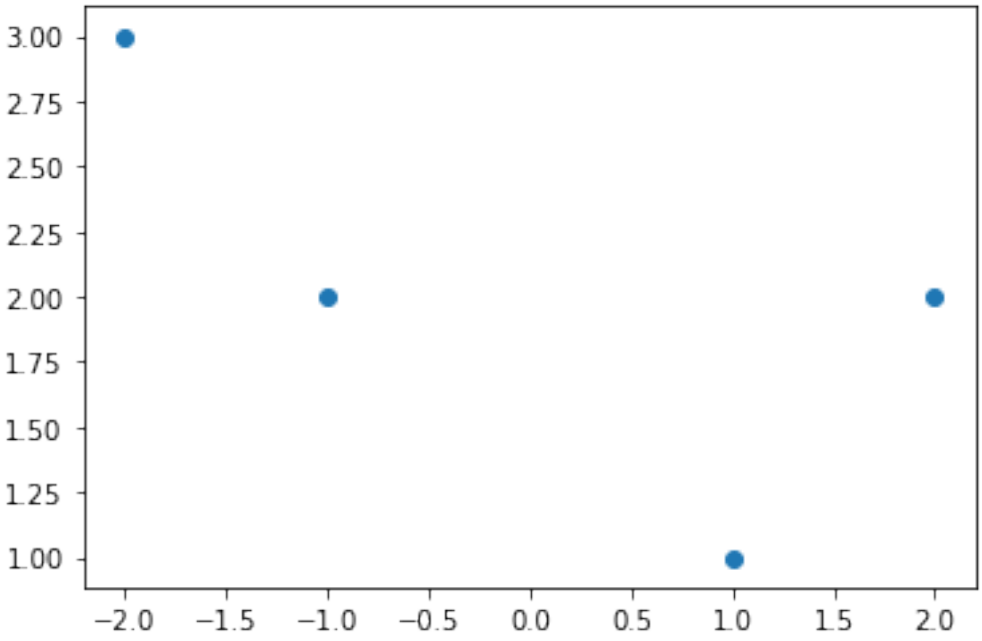
[4]:
```
array([[ 0.        ,  0.45241393],
       [-0.84515425, -0.12926112],
       [ 0.16903085, -0.87897563],
       [ 0.50709255,  0.07755667]])
```

[5]:
```
vt1 = np.array([[-1, 3]])
v1 =  vt1.T/np.linalg.norm(vt1)
Proj1 = A1@(v1)@(v1.T)
print(Proj1)
print("Problem 1-2")
plt.scatter(A1[:,0], A1[:,1])
plt.scatter(Proj1[:,0], Proj1[:,1])
plt.show()
```

```
[[-0.4  1.2]
 [-0.7  2.1]
 [-0.2  0.6]
 [-1.1  3.3]]
Problem 1-2
```



[6]:
```
P1.solve(A1)
```

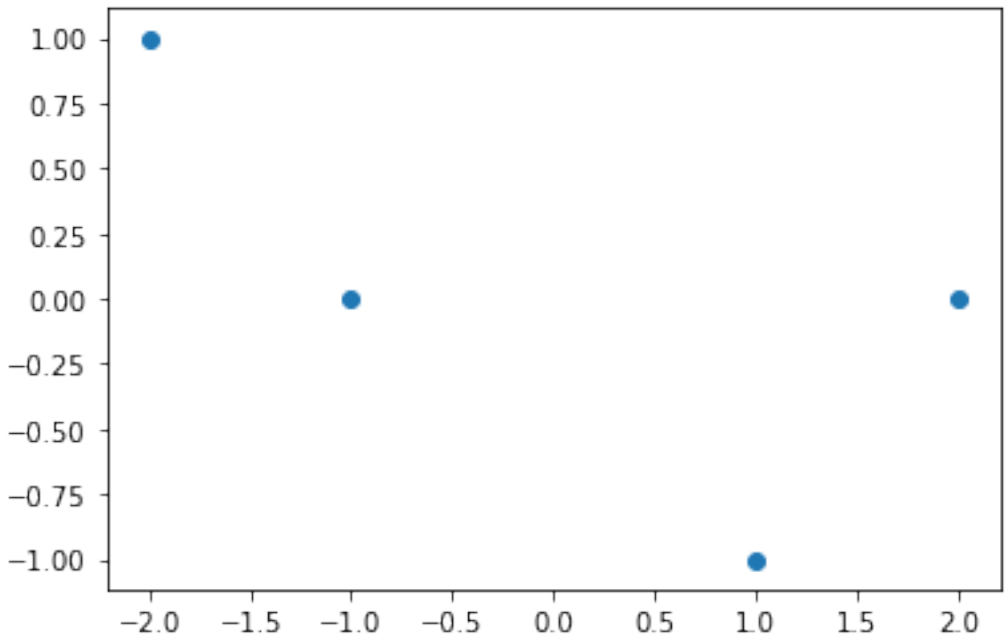



```
[  1. -28. 171.]
[19.  9.]
eiggen vectors of eigen value 19.0
eiggen vectors of eigen value 9.0
(array([ 9., 19.]), array([[-0.9486833 ,  0.31622777],
       [-0.31622777, -0.9486833 ]]))
```

[7]:
```
class P2:
    def solve(A):
        m = np.mean(A, axis=0)
        Am = A-m
        print(m, Am)
        P1.solve(Am)
```

[8]:
```
P2.solve(A1)
```

```
[0. 2.] [[ 2.  0.]
 [-1.  0.]
 [ 1. -1.]
 [-2.  1.]]
```

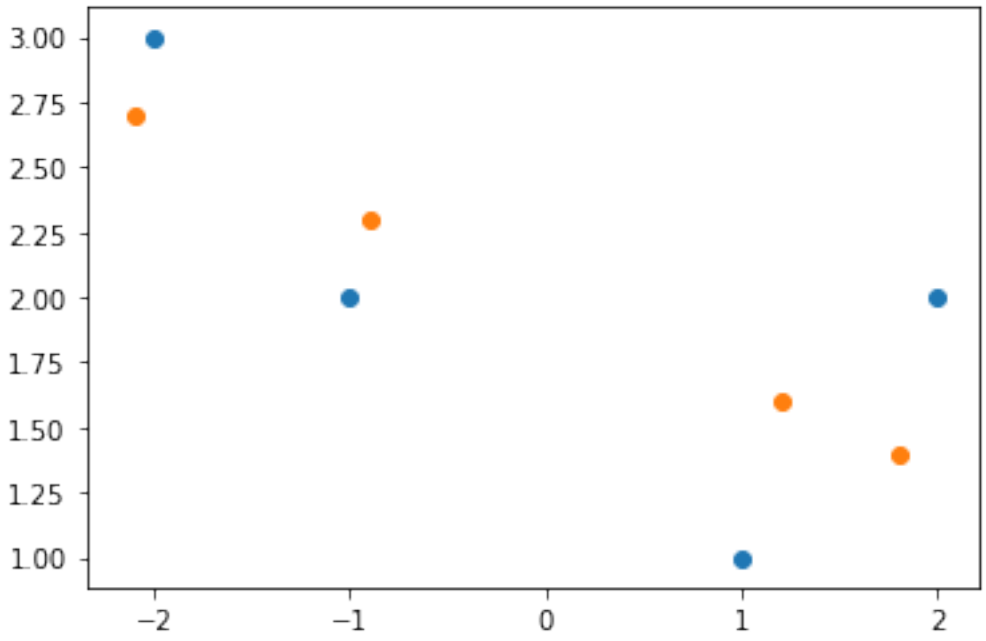



```
[  1. -12.  11.]
[11.  1.]
eiggen vectors of eigen value 11.0
eiggen vectors of eigen value 1.0
(array([11.,  1.]), array([[ 0.9486833 ,  0.31622777],
       [-0.31622777,  0.9486833 ]]))
```

[34]:
```
# self calc A1 PCA vector 1
m1 = np.mean(A1, axis=0)
print(m1)
Am1 = A1-m1
print(Am1)
B2 = Am1.T@Am1
print("AmT@Am\n", B2)
print("eigenvs\n", np.roots(np.poly(B2)))

print(np.linalg.eig(B2))
vt12 = np.array([[-3, 1]])
vt122 = np.array([[1, 3]])
#print("u12\n", Am1@(vt12.T))
#print("u22\n", Am1@(vt122.T))
v12 =  vt12.T/np.linalg.norm(vt12)
Proj12 = Am1@(v12)@(v12.T)
Proj12 = Proj12 + m1
print(Proj12)
print("Problem 1-4")
```

```
plt.scatter(A1[:,0], A1[:,1])
plt.scatter(Proj12[:,0], Proj12[:,1])
plt.show()
```

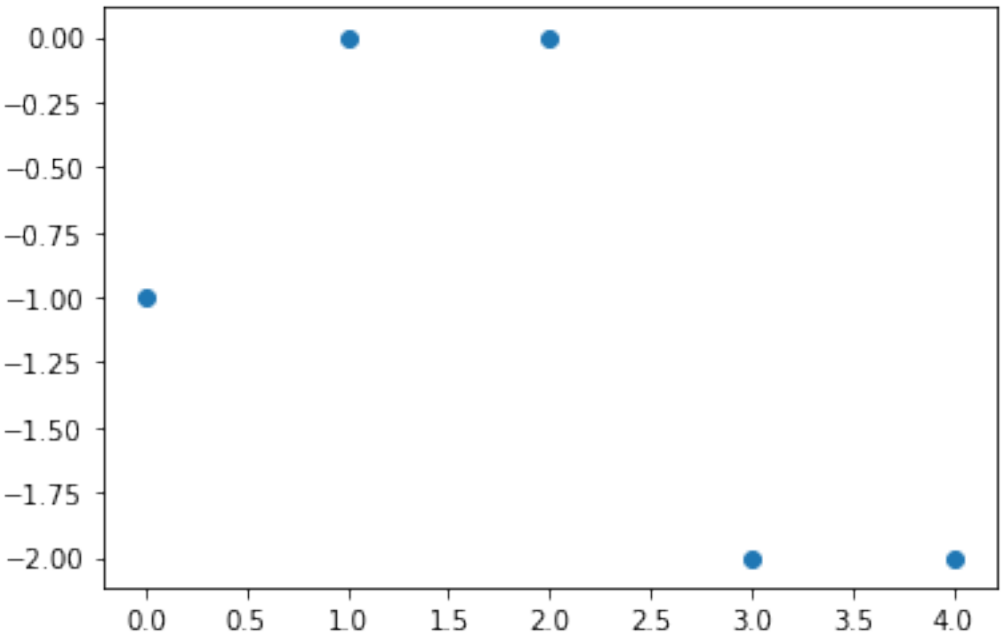
```
[0. 2.]
[[ 2.  0.]
 [-1.  0.]
 [ 1. -1.]
 [-2.  1.]]
AmT@Am
 [[10. -3.]
 [-3.  2.]]
eigenvs
 [11.  1.]
(array([11.,  1.]), array([[ 0.9486833 ,  0.31622777],
       [-0.31622777,  0.9486833 ]]))
[[ 1.8  1.4]
 [-0.9  2.3]
 [ 1.2  1.6]
 [-2.1  2.7]]
Problem 1-4
```



[35]:
```
A3 = np.array([[0, -1],
               [4, -2],
               [3, -2],
               [1, 0],
```

```
              [2, 0]
             ])
```

[36]:
```
P1.solve(A3)
```

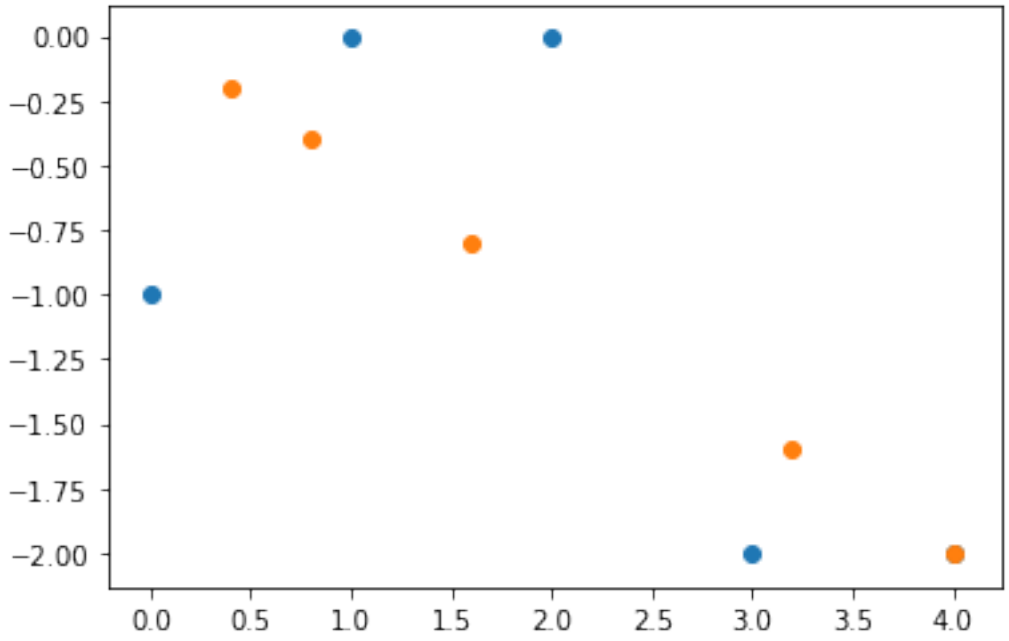



```
[  1. -39.  74.]
[37.  2.]
eiggen vectors of eigen value 37.0
eiggen vectors of eigen value 2.0
(array([37.,  2.]), array([[ 0.89442719,  0.4472136 ],
       [-0.4472136 ,  0.89442719]]))
```

[42]:
```
vt31 = np.array([[2, -1]])
vt32 = np.array([[1, 2]])
v31 =  vt31.T/np.linalg.norm(vt31)
Proj3 = A3@(v31)@(v31.T)
print(Proj3)
print("Problem 2-1")
plt.scatter(A3[:,0], A3[:,1])
plt.scatter(Proj3[:,0], Proj3[:,1])
plt.show()
```

```
[[ 0.4 -0.2]
 [ 4.  -2. ]
 [ 3.2 -1.6]
 [ 0.8 -0.4]]
```

```
 [ 1.6 -0.8]]
Problem 2-1
```

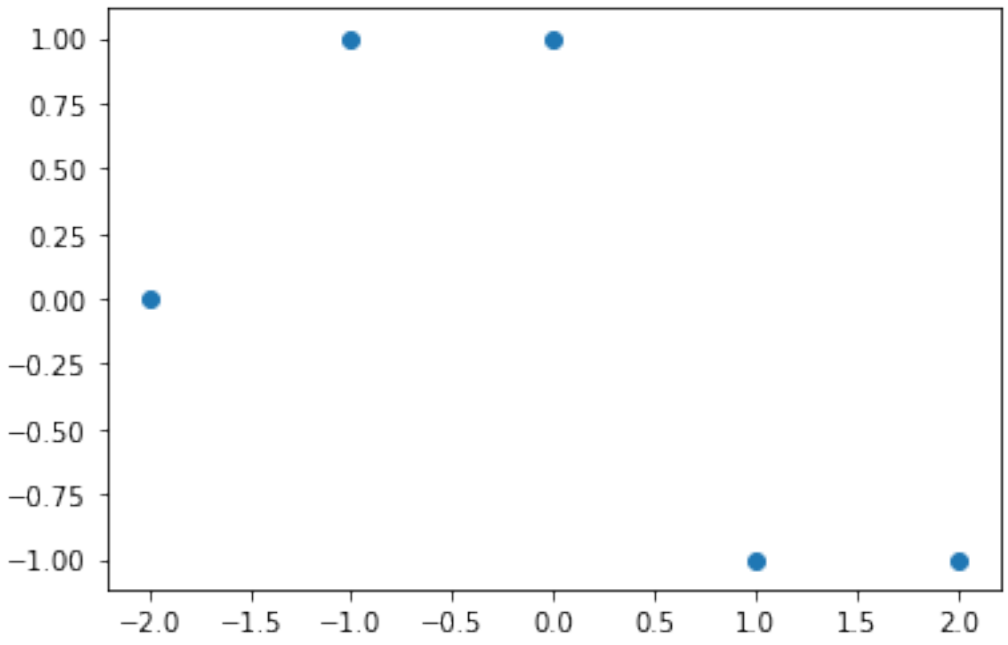


[43]:
```
A3@vt31.T
A3@vt32.T
```

[43]:
```
array([[-2],
       [ 0],
       [-1],
       [ 1],
       [ 2]])
```

[44]:
```
P2.solve(A3)
```

```
[ 2. -1.] [[-2.  0.]
 [ 2. -1.]
 [ 1. -1.]
 [-1.  1.]
 [ 0.  1.]]
```

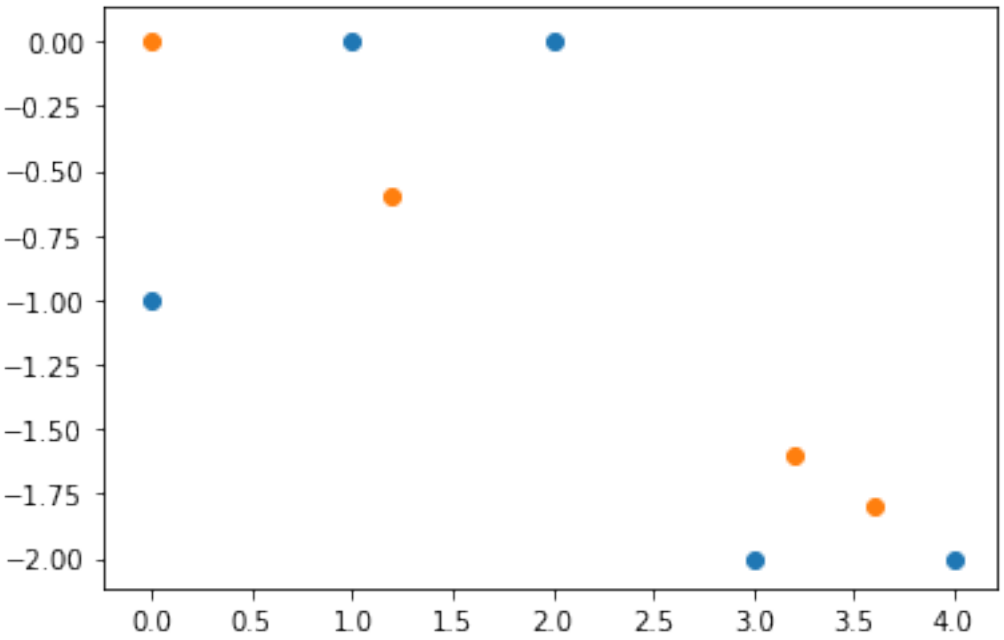



```
[  1. -14.  24.]
[12.  2.]
eiggen vectors of eigen value 12.0
eiggen vectors of eigen value 2.0
(array([12.,  2.]), array([[ 0.89442719,  0.4472136 ],
       [-0.4472136 ,  0.89442719]]))
```

[50]:
```
### PCA of Q2

m3 = np.mean(A3, axis=0)
print(m1)
Am3 = A3-m3
print(Am3)
B3 = Am3.T@Am3
print("AmT@Am\n", B3)
print("eigenvs\n", np.roots(np.poly(B3)))

print(np.linalg.eig(B3))
vt32 = np.array([[2, -1]])
vt322 = np.array([[1, 2]])
#print("u12\n", Am1@(vt12.T))
#print("u22\n", Am1@(vt122.T))
v32 =  vt32.T/np.linalg.norm(vt32)
Proj32 = Am1@(v32)@(v32.T)
Proj32 = Proj32 + m3
print(Proj32)
```

```
print("Problem 2-4")
plt.scatter(A3[:,0], A3[:,1])
plt.scatter(Proj32[:,0], Proj32[:,1])
plt.show()
```

```
[0. 2.]
[[-2.  0.]
 [ 2. -1.]
 [ 1. -1.]
 [-1.  1.]
 [ 0.  1.]]
AmT@Am
 [[10. -4.]
 [-4.  4.]]
eigenvs
 [12.  2.]
(array([12.,  2.]), array([[ 0.89442719,  0.4472136 ],
       [-0.4472136 ,  0.89442719]]))
[[ 3.6 -1.8]
 [ 1.2 -0.6]
 [ 3.2 -1.6]
 [ 0.   0. ]]
Problem 2-4
```



[ ]: