02a.eigen_face_steps

April 21, 2018

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np
        from eigenface.yalefaces import YaleFaceDb
        np.set_printoptions(precision=2, suppress=True, formatter={'float': '{: 0.2f}'.format}
```

```
In [2]: db = YaleFaceDb()

        def gen_images():
            images = np.empty(shape=(3,4,4))
            images[0,:,:] = np.array([[1,2,3,4],[5,6,7,8],[5,6,7,8],[5,6,7,8]])
            images[1,:,:] = np.array([[2,3,4,5],[6,7,8,9],[5,6,7,8],[5,6,7,8]])
            images[2,:,:] = np.array([[0,2,4,6],[3,5,7,9],[5,6,7,8],[5,6,7,8]])
            return images
        # images = gen_images().astype(dtype=np.float)
        images = db.get_list().astype(dtype=np.float)
        labels = db.get_label()
        print(images[:,:,:,0])
        images.shape
```

```
[[[ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  ...,
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]]

 [[ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  ...,
  [ 255.00  255.00  255.00 ...,  254.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  253.00  240.00  240.00]
  [ 255.00  255.00  255.00 ...,  255.00  254.00  252.00]]

 [[ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
```

```
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  ...,
  [ 255.00  255.00  254.00 ...,  247.00  250.00  255.00]
  [ 255.00  255.00  255.00 ...,  243.00  242.00  254.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  252.00]]

 ...,
 [[ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  ...,
  [ 254.00  254.00  253.00 ...,  221.00  220.00  231.00]
  [ 255.00  255.00  255.00 ...,  117.00  162.00  190.00]
  [ 255.00  225.00  200.00 ...,  27.00  40.00  101.00]]

 [[ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  ...,
  [ 255.00  242.00  202.00 ...,  217.00  220.00  219.00]
  [ 255.00  250.00  253.00 ...,  241.00  248.00  216.00]
  [ 251.00  243.00  251.00 ...,  139.00  222.00  223.00]]

 [[ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  ...,
  [ 254.00  254.00  254.00 ...,  253.00  254.00  254.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
  [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]]]
```

Out[2]: (165, 100, 100, 1)

In [3]:
```
# M traiing images with width NxN
# ==> np.array with N^2 row (number of pixels), M column (number of images)
def convert_matrix_presentation(images):
    vector2d = []
    for image in images:
        vector = image.flatten()
        vector2d.append(vector)
    return np.array(vector2d)

def test_matrix_presentation():
    return convert_matrix_presentation(images)
vector_matrix = test_matrix_presentation()
print(vector_matrix)
print(vector_matrix.shape)
```

```
[[ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]
 [ 255.00  255.00  255.00 ...,  255.00  254.00  252.00]
 [ 255.00  255.00  255.00 ...,  255.00  255.00  252.00]
 ...,
 [ 255.00  255.00  255.00 ...,  27.00  40.00  101.00]
 [ 255.00  255.00  255.00 ...,  139.00  222.00  223.00]
 [ 255.00  255.00  255.00 ...,  255.00  255.00  255.00]]
(165, 10000)
```

```
In [4]: mean_vector = vector_matrix.mean(axis=0)
        print(mean_vector)
        print(mean_vector.shape)
```

```
[ 248.67  248.90  248.86 ...,  209.83  212.42  214.73]
(10000,)
```

```
In [5]: vector_mean_matrix = vector_matrix[:,:] - mean_vector
        print(vector_mean_matrix)
```

```
[[ 6.33  6.10  6.14 ...,  45.17  42.58  40.27]
 [ 6.33  6.10  6.14 ...,  45.17  41.58  37.27]
 [ 6.33  6.10  6.14 ...,  45.17  42.58  37.27]
 ...,
 [ 6.33  6.10  6.14 ..., -182.83 -172.42 -113.73]
 [ 6.33  6.10  6.14 ..., -70.83  9.58  8.27]
 [ 6.33  6.10  6.14 ...,  45.17  42.58  40.27]]
```

```
In [6]: covariance_matrix = np.matmul(vector_mean_matrix,vector_mean_matrix.T) # vector_matrix
        print(covariance_matrix.shape)
        print(covariance_matrix)
```

```
(165, 165)
[[ 29355065.79  8846414.05  9457439.14 ...,  581126.85  2089275.27  566094.10]
 [ 8846414.05  25373990.32  20047641.41 ..., -2797640.88 -3135223.47 -2588681.63]
 [ 9457439.14  20047641.41  25803591.49 ..., -847127.80  72573.62 -918147.55]
 ...,
 [ 581126.85 -2797640.88 -847127.80 ...,  42143862.91  32331982.33  21220256.16]
 [ 2089275.27 -3135223.47  72573.62 ...,  32331982.33  37888239.75  16034989.58]
 [ 566094.10 -2588681.63 -918147.55 ...,  21220256.16  16034989.58  32993349.42]]
```

```
In [7]: u, eigen_value, eigen_vector_vi = np.linalg.svd(covariance_matrix)
        print("u:", u.shape, "\n", u)
        print("eigen_value:\n", eigen_value)
        print("eigen_vector_vi: ", eigen_vector_vi.shape, "\n", eigen_vector_vi)
```

```
u: (165, 165)
 [[-0.06 -0.04  0.02 ...,  0.00  0.00 -0.08]
 [ 0.03  0.05  0.02 ...,  0.00  0.00 -0.08]
 [ 0.03  0.03  0.00 ...,  0.00  0.00 -0.08]
 ...,
 [ 0.04 -0.07  0.08 ...,  0.00  0.00 -0.08]
 [ 0.04 -0.07  0.04 ...,  0.00  0.00 -0.08]
 [ 0.05 -0.09  0.11 ...,  0.00  0.00 -0.08]]
eigen_value:
 [ 1074499909.05  973989965.73  693086965.82  399710027.64  386363173.15  302872688.11  209279
  5159310.30  5084672.34  4940174.67  4776120.35  4691205.32  4616503.28  4518791.45  4455352.
  0.00]
eigen_vector_vi:  (165, 165)
 [[-0.06  0.03  0.03 ...,  0.04  0.04  0.05]
 [-0.04  0.05  0.03 ..., -0.07 -0.07 -0.09]
 [ 0.02  0.02  0.00 ...,  0.08  0.04  0.11]
 ...,
 [ 0.00 -0.00  0.00 ..., -0.00 -0.00 -0.00]
 [ 0.00 -0.00 -0.00 ..., -0.00 -0.00 -0.00]
 [-0.08 -0.08 -0.08 ..., -0.08 -0.08 -0.08]]
```

```
In [8]: # Check Covar.x = r.x
        print('Check Eigen Properties Wx = rx:')
        print('Wx:', np.matmul(covariance_matrix, eigen_vector_vi[0,:]))
        print('rx:', eigen_value[0] * eigen_vector_vi[0,:])
```

```
Check Eigen Properties Wx = rx:
Wx: [-62514077.58  29641461.92  33867343.71 -195682964.92  43701962.63  27569130.16 -49524581.9
  64446837.28  28256621.50  64620596.47 -141553928.33  46116786.13  22975904.66 -23609906.41  4
  14573830.44  43016134.71  56745566.52  70993865.57 -176335605.10  47028808.85  76074210.49  1
rx: [-62514077.58  29641461.92  33867343.71 -195682964.92  43701962.63  27569130.16 -49524581.9
  64446837.28  28256621.50  64620596.47 -141553928.33  46116786.13  22975904.66 -23609906.41  4
  14573830.44  43016134.71  56745566.52  70993865.57 -176335605.10  47028808.85  76074210.49  1
```

```
In [9]: eigen_vector_u0 = np.matmul(vector_mean_matrix.T, eigen_vector_vi[0,:])
        print(eigen_vector_u0)
```

```
[ 2.67 -1.35 -1.73 ...,  415.28  397.83  415.44]
```

```
In [10]: eigen_vector_ui = np.matmul(vector_mean_matrix.T, eigen_vector_vi[:,:].T).T
         print(eigen_vector_ui)
```

```
[[ 2.67 -1.35 -1.73 ...,  415.28  397.83  415.44]
 [ 53.18  50.94  51.45 ..., -276.83 -274.72 -262.91]
 [ 108.12  106.08  102.10 ..., -71.50 -81.29 -65.64]
 ...,
```

```
 [ 0.00  0.00  0.00 ..., -0.00 -0.00  0.00]
 [ 0.00  0.00  0.00 ..., -0.00 -0.00 -0.00]
 [-0.00 -0.00 -0.00 ..., -0.00 -0.00 -0.00]]
```

```
In [11]: print('Check Eigen Properties Wx = rx:')
         print('Wx:', np.matmul(np.matmul(vector_mean_matrix.T,vector_mean_matrix), eigen_vecto
         print('rx:', eigen_value[0] * eigen_vector_ui[0,:])
```

```
Check Eigen Properties Wx = rx:
Wx: [ 2869581037.27 -1448751383.92 -1861773809.08 ...,  446218945567.74  427472261589.23  44638
rx: [ 2869581037.27 -1448751383.92 -1861773809.08 ...,  446218945567.74  427472261589.22  44638
```

```
In [12]: norms = np.linalg.norm(eigen_vector_ui, axis=1)
         print(norms)
```

```
[ 32779.57  31208.81  26326.54  19992.75  19656.12  17403.24  14466.50  13263.08  12870.51  122
  1675.16  1647.79  1635.35  1630.89  1618.45  1607.96  1582.94  1559.10  1541.22  1527.69  151
```

```
In [13]: norm_ui = np.divide(eigen_vector_ui.T, norms).T
         print(norm_ui)
         (norm_ui[2]*norm_ui[2]).sum()
```

```
[[ 0.00 -0.00 -0.00 ...,  0.01  0.01  0.01]
 [ 0.00  0.00  0.00 ..., -0.01 -0.01 -0.01]
 [ 0.00  0.00  0.00 ..., -0.00 -0.00 -0.00]
 ...,
 [ 0.01  0.01  0.01 ..., -0.00 -0.00  0.00]
 [ 0.00  0.00  0.00 ..., -0.01 -0.00 -0.01]
 [-0.02 -0.01 -0.01 ..., -0.00 -0.01 -0.01]]
```

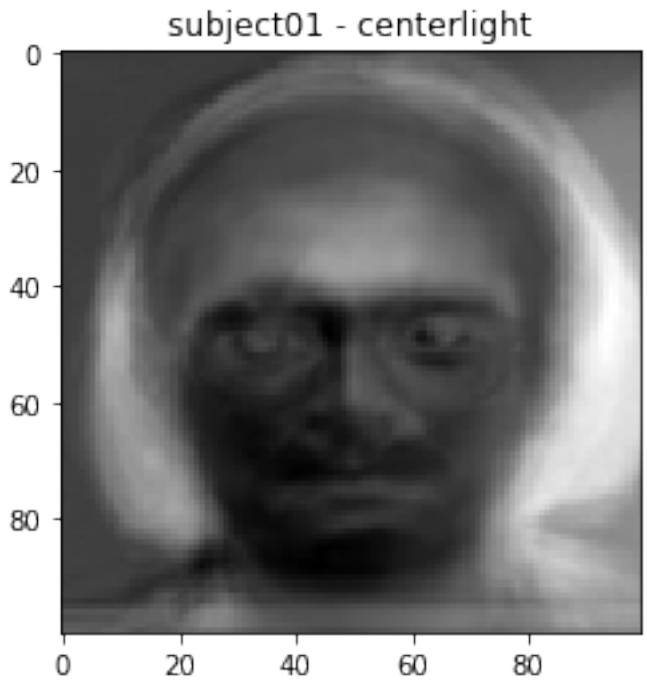
```
Out[13]: 1.0000000000000036
```

```
In [14]: eigen_faces = norm_ui.reshape(images.shape)
         print("Eigen Faces:\n", eigen_faces[:,:,:,0])
```

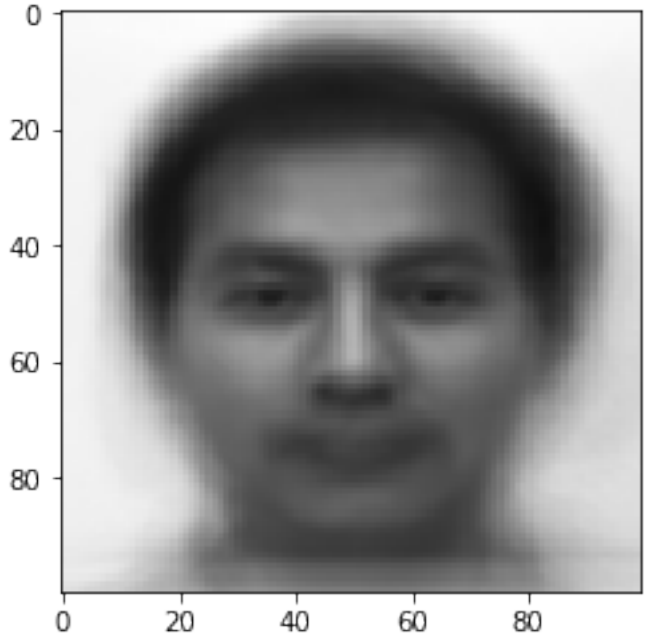
```
Eigen Faces:
 [[[ 0.00 -0.00 -0.00 ...,  0.00  0.00  0.00]
  [ 0.00  0.00  0.00 ...,  0.00  0.00  0.00]
  [ 0.00  0.00  0.00 ...,  0.00  0.00  0.00]
  ...,
  [ 0.00  0.00  0.00 ...,  0.02  0.02  0.02]
  [ 0.00  0.00  0.00 ...,  0.01  0.01  0.01]
  [ 0.01  0.01  0.01 ...,  0.01  0.01  0.01]]

 [[ 0.00  0.00  0.00 ...,  0.00  0.00  0.00]
```

```
 [ 0.00  0.00  0.00 ...,  0.00  0.00  0.00]
 [ 0.00  0.00  0.00 ...,  0.00  0.00  0.00]
 ...,
 [ 0.00  0.00  0.00 ..., -0.01 -0.01 -0.00]
 [ 0.00  0.00  0.00 ..., -0.01 -0.01 -0.01]
 [ 0.00  0.00  0.00 ..., -0.01 -0.01 -0.01]]

[[ 0.00  0.00  0.00 ..., -0.00 -0.00 -0.00]
 [ 0.01  0.00  0.00 ..., -0.00 -0.00 -0.00]
 [ 0.01  0.01  0.01 ..., -0.00 -0.00 -0.00]
 ...,
 [ 0.01  0.01  0.01 ..., -0.01 -0.01 -0.00]
 [ 0.01  0.01  0.01 ..., -0.00 -0.00 -0.00]
 [ 0.01  0.01  0.01 ..., -0.00 -0.00 -0.00]]

...,
[[ 0.01  0.01  0.01 ..., -0.01 -0.00 -0.00]
 [ 0.00  0.00  0.00 ..., -0.00 -0.00 -0.00]
 [ 0.01  0.01  0.01 ..., -0.00 -0.00 -0.00]
 ...,
 [ 0.01  0.01  0.01 ..., -0.02 -0.01 -0.00]
 [ 0.01  0.00 -0.00 ..., -0.00 -0.00 -0.01]
 [ 0.01  0.01  0.01 ..., -0.00 -0.00  0.00]]

[[ 0.00  0.00  0.00 ...,  0.02  0.01  0.01]
 [ 0.00  0.00  0.00 ...,  0.01  0.01  0.01]
 [ 0.01  0.01  0.01 ...,  0.01  0.01  0.01]
 ...,
 [-0.01 -0.01 -0.01 ..., -0.00  0.00  0.00]
 [ 0.01  0.01  0.00 ..., -0.00  0.01  0.00]
 [ 0.00  0.00  0.01 ..., -0.01 -0.00 -0.01]]

[[-0.02 -0.01 -0.01 ...,  0.00 -0.00  0.00]
 [ 0.00  0.01  0.00 ...,  0.01  0.00  0.00]
 [ 0.00  0.00  0.00 ...,  0.01  0.00  0.00]
 ...,
 [-0.01 -0.01 -0.00 ...,  0.01  0.02 -0.00]
 [-0.00 -0.00  0.00 ..., -0.01 -0.00 -0.00]
 [ 0.02  0.01  0.01 ..., -0.00 -0.01 -0.01]]]
```

```
In [15]: plt.imshow(eigen_faces[0,:,:,0].astype(dtype=np.float), cmap='gray')
         plt.title('%s - %s'%(labels[0,0],labels[0,1]))
         plt.show()
```



In [16]:
```
mean_image = (mean_vector).reshape(images.shape[1], images.shape[2]).astype(dtype=np.u
plt.imshow(np.dstack([mean_image,mean_image,mean_image]))
plt.show()
```



In [ ]: