zero-base /

Life Changing Education

Chapter 23. PCA eigenface

Olivetti 데이터

zero-base /

데이터

`sklearn.datasets.` **`fetch_olivetti_faces`** (*data_home=None, shuffle=False, random_state=0, download_if_missing=True*) [source]

Load the Olivetti faces data-set from AT&T (classification).

Download it if necessary.

Classes	40
Samples total	400
Dimensionality	4096
Features	real, between 0 and 1

zero-base /

AT&T와 캠브리지 대학의 공동 연구 데이터

코드

zero-base /

데이터 읽기

```
from sklearn.datasets import fetch_olivetti_faces

faces_all = fetch_olivetti_faces()
print(faces_all.DESCR)
```

- 이 데이터는 얼굴 인식용으로 사용할 수 있지만,
- 우리는 특정 인물의 데이터(10장)만 이용해서 PCA 실습용으로 사용

zero-base /

특정 샘플만 선택

```
K = 20
faces = faces_all.images[faces_all.target == K]
```

```
faces
```

```
array([[[0.5165289 , 0.5123967 , 0.5082645 , ..., 0.42975205,
         0.42561984, 0.41735536],
        [0.5082645 , 0.5123967 , 0.5206612 , ..., 0.42975205,
         0.42975205, 0.4214876 ],
        [0.4876033 , 0.5123967 , 0.5289256 , ..., 0.4338843 ,
         0.42975205, 0.42975205],
        ...,
```

zero-base /

무슨 데이터가 있는거지?

```
import matplotlib.pyplot as plt

N = 2
M = 5
fig = plt.figure(figsize=(10, 5))
plt.subplots_adjust(top=1, bottom=0, hspace=0, wspace=0.05)
for n in range(N*M):
    ax = fig.add_subplot(N, M, n+1)
    ax.imshow(faces[n], cmap=plt.cm.bone)
    ax.grid(False)
    ax.xaxis.set_ticks([])
    ax.yaxis.set_ticks([])

plt.suptitle("Olivetti")
plt.tight_layout()
plt.show()
```

zero-base /

Hello~

zero-base /

두 개의 성분으로 분석

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)

X = faces_all.data[faces_all.target == K]
W = pca.fit_transform(X)

X_inv = pca.inverse_transform(W)
```

zero-base /

분석된 결과를 확인해보자

```
N = 2
M = 5
fig = plt.figure(figsize=(10, 5))
plt.subplots_adjust(top=1, bottom=0, hspace=0, wspace=0.05)
for n in range(N*M):
    ax = fig.add_subplot(N, M, n+1)
    ax.imshow(X_inv[n].reshape(64, 64), cmap=plt.cm.bone)
    ax.grid(False)
    ax.xaxis.set_ticks([])
    ax.yaxis.set_ticks([])

plt.suptitle("PCA result")
plt.tight_layout()
plt.show()
```

zero-base /

결과

PCA result

zero-base /

원점과 두 개의 eigen face

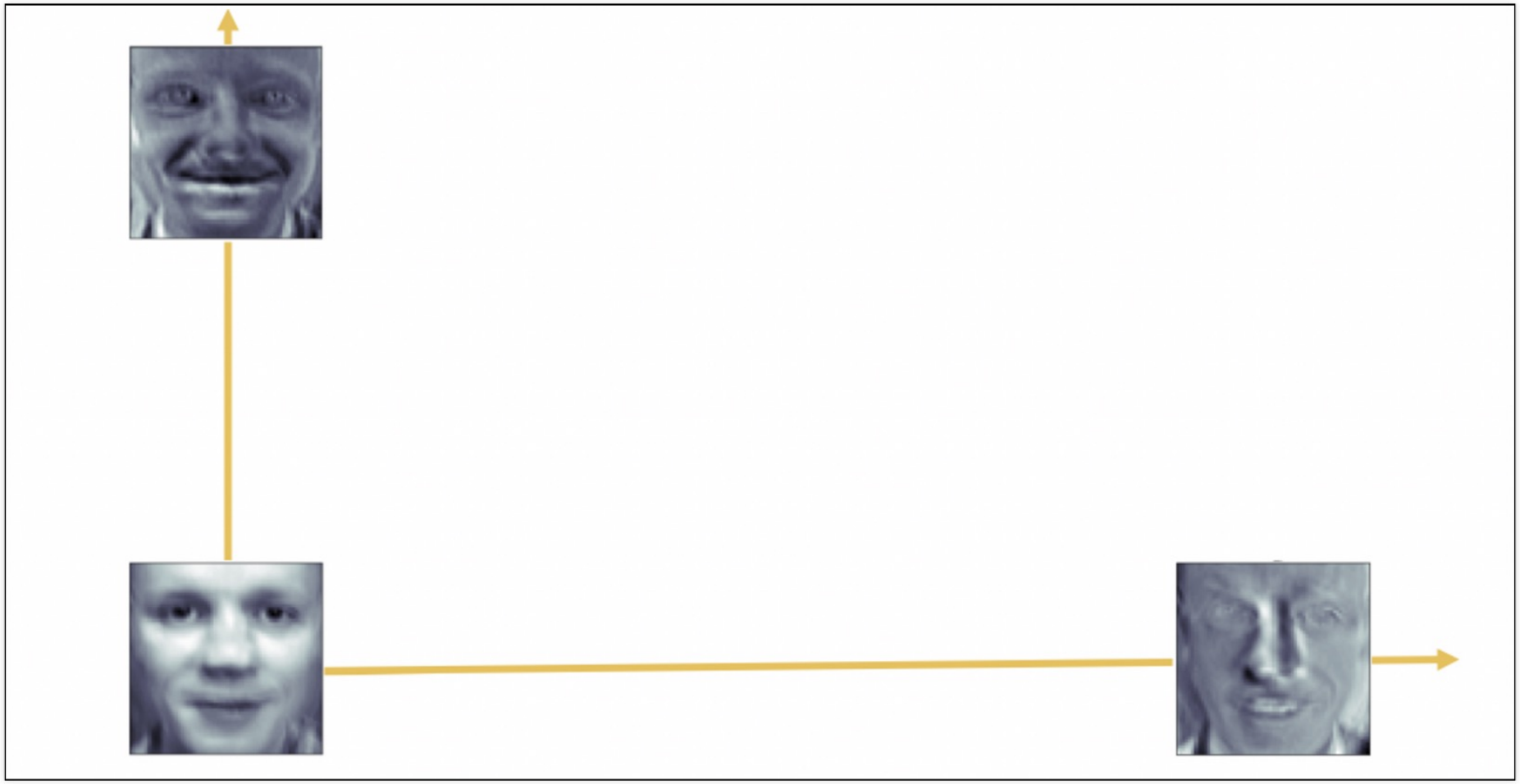
```
face_mean = pca.mean_.reshape(64, 64)
face_p1 = pca.components_[0].reshape(64, 64)
face_p2 = pca.components_[1].reshape(64, 64)

plt.figure(figsize=(12,7))
plt.subplot(131)
plt.imshow(face_mean, cmap=plt.cm.bone)
plt.grid(False); plt.xticks([]); plt.yticks([]); plt.title("mean")
plt.subplot(132)
plt.imshow(face_p1, cmap=plt.cm.bone)
plt.grid(False); plt.xticks([]); plt.yticks([]); plt.title("face_p1")
plt.subplot(133)
plt.imshow(face_p2, cmap=plt.cm.bone)
plt.grid(False); plt.xticks([]); plt.yticks([]); plt.title("face_p2")
plt.show()
```

zero-base /

10장의 사진은 이 세장으로 모두 표현할 수 있다

zero-base /

이렇게 상상해도 된다



zero-base /

가중치 선정

```
import numpy as np

N = 2
M = 5
w = np.linspace(-5, 10, N*M)
w
```

```
array([-5.        , -3.33333333, -1.66666667,  0.        ,  1.66666667,
        3.33333333,  5.        ,  6.66666667,  8.33333333, 10.        ])
```

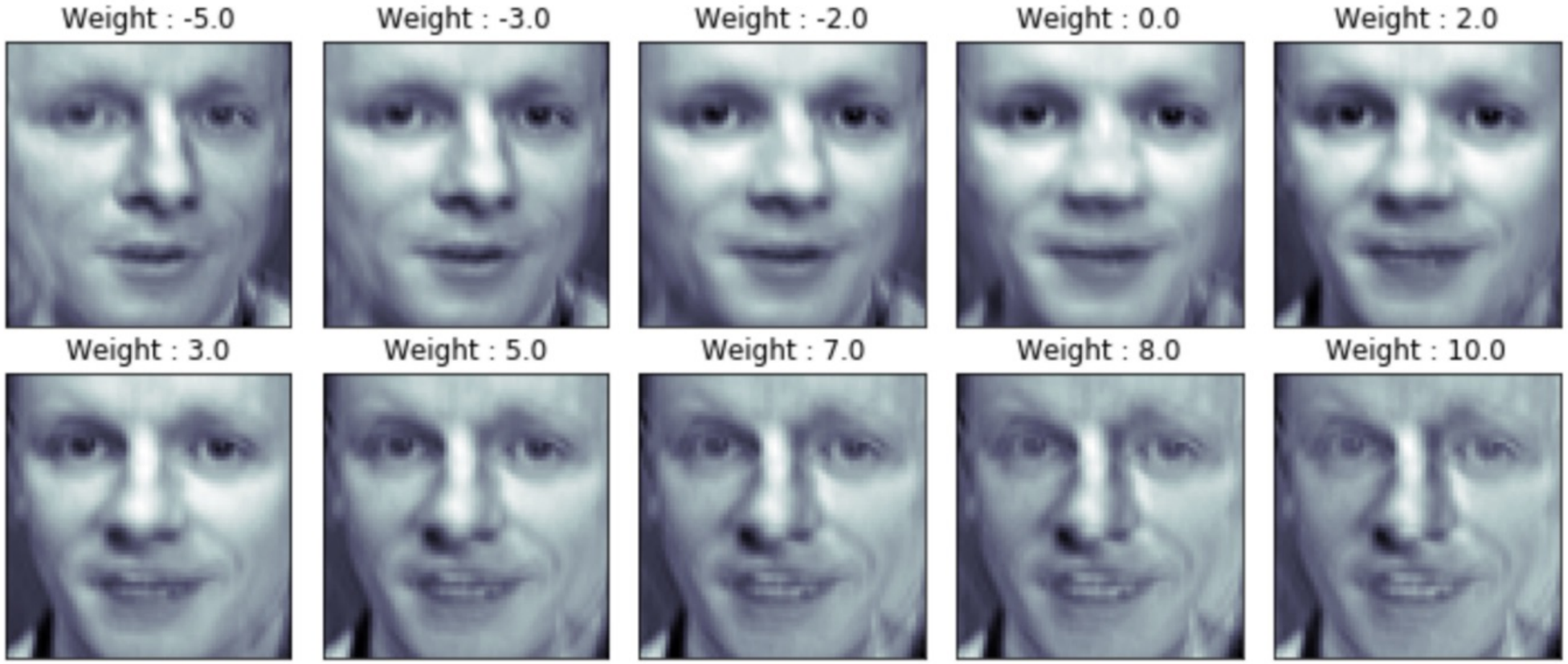
zero-base /

첫번째 성분의 변화

```
fig = plt.figure(figsize=(10, 5))
plt.subplots_adjust(top=1, bottom=0, hspace=0, wspace=0.05)

for n in range(N*M):
    ax = fig.add_subplot(N, M, n+1)
    ax.imshow(face_mean + w[n] * face_p1, cmap=plt.cm.bone)
    plt.grid(False); plt.xticks([]); plt.yticks([])
    plt.title('Weight : ' + str(round(w[n])))

plt.tight_layout()
plt.show()
```

zero-base /

결과



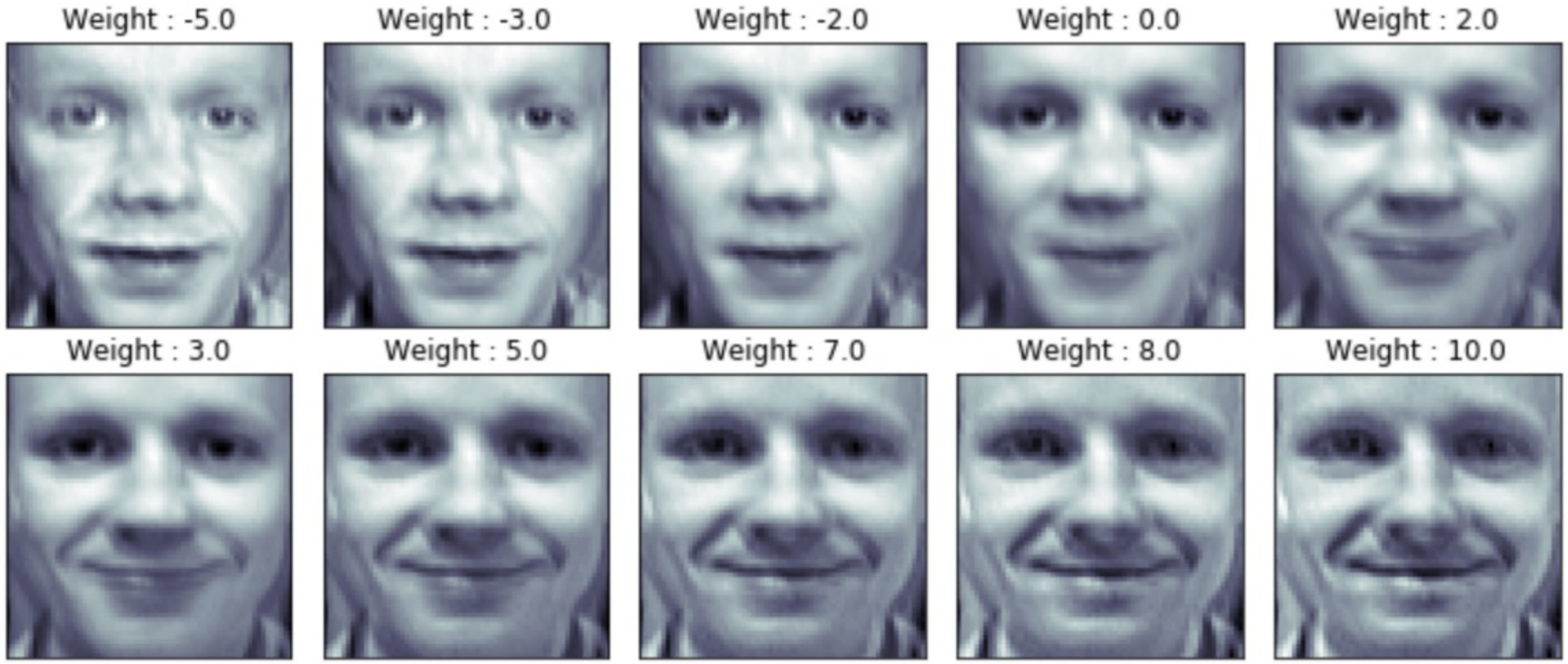
zero-base /

두 번째 성분에 대한 변화

```
fig = plt.figure(figsize=(10, 5))
plt.subplots_adjust(top=1, bottom=0, hspace=0, wspace=0.05)

for n in range(N*M):
    ax = fig.add_subplot(N, M, n+1)
    ax.imshow(face_mean + w[n] * face_p2, cmap=plt.cm.bone)
    plt.grid(False); plt.xticks([]); plt.yticks([])
    plt.title('Weight : ' + str(round(w[n])))

plt.tight_layout()
plt.show()
```

zero-base /

결과



zero-base /

두 개의 성분을 다 표현해보기

```
nx, ny = (5, 5)
x = np.linspace(-5, 8, nx)
y = np.linspace(-5, 8, ny)
w1, w2 = np.meshgrid(x, y)
w1, w2
```

```
(array([[-5.  , -1.75,  1.5 ,  4.75,  8.  ],
        [-5.  , -1.75,  1.5 ,  4.75,  8.  ],
        [-5.  , -1.75,  1.5 ,  4.75,  8.  ],
        [-5.  , -1.75,  1.5 ,  4.75,  8.  ],
        [-5.  , -1.75,  1.5 ,  4.75,  8.  ]]),
 array([[-5.  , -5.  , -5.  , -5.  , -5.  ],
        [-1.75, -1.75, -1.75, -1.75, -1.75],
        [ 1.5 ,  1.5 ,  1.5 ,  1.5 ,  1.5 ],
        [ 4.75,  4.75,  4.75,  4.75,  4.75],
        [ 8.  ,  8.  ,  8.  ,  8.  ,  8.  ]]))
```

zero-base /

shape 조정

```
w1.shape
```

```
(5, 5)
```

```
w1 = w1.reshape(-1,)
w2 = w2.reshape(-1,)
w1.shape
```

```
(25,)
```

zero-base /

다시 합성

```
fig = plt.figure(figsize=(12, 10))
plt.subplots_adjust(top=1, bottom=0, hspace=0, wspace=0.05)

N = 5
M = 5

for n in range(N*M):
    ax = fig.add_subplot(N, M, n+1)
    ax.imshow(face_mean + w1[n] * face_p1 + w2[n] * face_p2, cmap=plt.cm.bone)
    plt.grid(False); plt.xticks([]); plt.yticks([])
    plt.title('Weight : ' + str(round(w1[n], 1)) + ', ' + str(round(w2[n], 1)))

plt.show()
```

zero-base /

결과

zero-base /

뭐 이런 느낌