# 第八章

一、EM算法求解PCA，参数在M步更新公式

1. 在M步，对W和$\sigma^2$求导更新参数

$$\ln P(x,z|\mu,W,\sigma^2)=\sum_{n=1}^{N}\left(\ln P(x_n|z_n)+\ln P(z_n)\right)$$

把 $P(x_n|z_n)=\frac{1}{(2\pi)^{\frac{p}{2}}|\sigma^2 I|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2\sigma^2}(x_n-Wz_n-\mu)^T(x_n-Wz_n-\mu)\right\}$

$P(z_n)=\frac{1}{(2\pi)^{\frac{k}{2}}}\exp\left(-\frac{1}{2}z_n^Tz_n\right)$ $\mu=\bar{x}$ 代入得.

$$\ln\left(P(x,z|\mu,W,\sigma^2)\right)=-\sum_{n=1}^{N}\left\{\frac{p}{2}\ln(2\pi\sigma^2)+\frac{1}{2}tr(E[z_nz_n^T])+\frac{1}{2\sigma^2}||x_n-\bar{x}||^2\right.$$
$$\left.-\frac{1}{\sigma^2}E[z_n^T]W^T(x_n-\bar{x})+\frac{1}{2\sigma^2}tr(E[z_nz_n^T]W^TW)+\frac{k}{2}\ln(2\pi)\right\}$$

① $$\frac{\partial\ln(P(x,z|\mu,W,\sigma^2))}{\partial W}=-\sum_{n=1}^{N}\left\{-\frac{1}{\sigma^2}(x_n-\bar{x})E(z_n)^T+\frac{1}{\sigma^2}WE(z_n z_n^T)\right\}=0$$

$$\Rightarrow\sum_{n=1}^{N}(x_n-\bar{x})E(z_n)^T=\sum_{n=1}^{N}WE(z_n z_n^T)\Rightarrow W_{new}=\left[\sum_{n=1}^{N}(x_n-\bar{x})E(z_n)^T\right]\left[\sum_{n=1}^{N}E(z_n z_n^T)\right]^{-1}$$

② $$\frac{\partial\ln(P(x,z|\mu,W,\sigma^2))}{\partial\sigma^2}=-\sum_{n=1}^{N}\left\{\frac{p}{2}\cdot\frac{1}{2\pi\sigma^2}\cdot 2\pi-\frac{1}{2}||x_n-\bar{x}||^2\cdot\frac{1}{(\sigma^2)^2}+E(z_n^T)W^T(x_n-\bar{x})\cdot\frac{1}{(\sigma^2)^2}\right.$$
$$\left.-\frac{1}{2}tr(E(z_n z_n^T)\cdot W^T W)\cdot\frac{1}{(\sigma^2)^2}\right\}=0$$

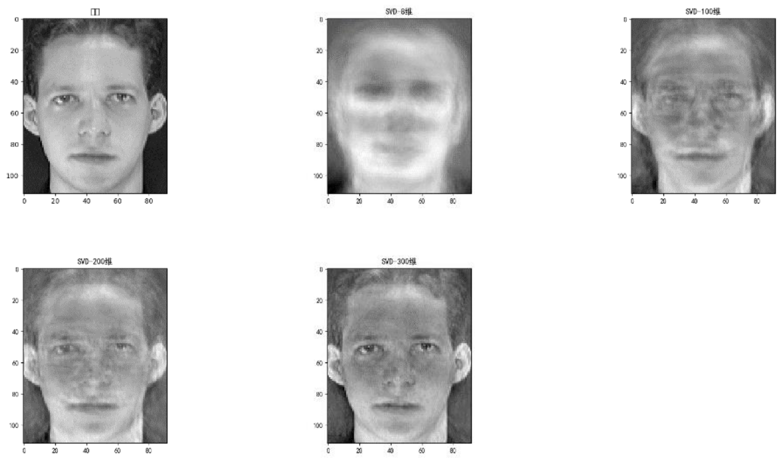
$$\Rightarrow\sigma^2_{new}=\frac{1}{Np}\sum_{n=1}^{N}\left\{||x_n-\bar{x}||^2-2E(z_n^T)W^T(x_n-\bar{x})+tr(E(z_n z_n^T)W_{new}^T W)\right\}$$

二、

（一）基于 SVD 的 PCA

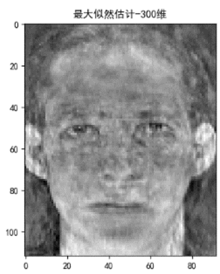
```python
# 使用 SVD 分解计算人脸图像的低维表示
svd_pca(X):  # 10304*400
p, m = X.shape
x_mean = np.mean(X, axis=1).reshape((p, 1))
print(x_mean.shape)
X = X - x_mean  # 均值归一化
A = np.dot(X.T, X) / m  # (400, 400)协方差矩阵
lamda, V = np.linalg.eig(A)  # A 的特征值以列的形式显示
for i in range(m):
    V[:, i:i + 1] /= np.dot(V[:, i:i + 1].T, V[:, i:i + 1])
sorted_indices = np.argsort(-lamda)
chance = [8, 20, 50, 100, 150, 200, 250, 300]  # 降维列表
data = []  # 用来保留降维后重构的数据
for k in chance:
    print("降到{}维，信息量保留为{}".format
          (k, np.sum([lamda[i] for i in range(k)] / np.sum(list(lamda)))))

    U = np.ones((10304, k))
    for i, j in zip(sorted_indices[0:k], range(k)):
        U[:, j:j + 1] = (X @ V[:, i:i + 1]) / np.sqrt(lamda[i])
    Z = U.T @ X
    X1 = U @ Z + x_mean  # 数据还原
    data.append(X1[:, 0])
data = np.array(data)
return data
```



（二）最大似然 PCA

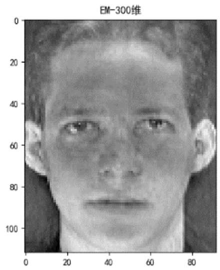
```
# 使用最大似然估计计算人脸图像的低维表示
max_likelihood_estimation_pca(data, k):
p, m = data.shape
mu = np.mean(data, axis=1).reshape((p, 1))
data = data - mu
S = (data @ data.T) / m  # 协方差矩阵
vector_U, value, vector_V = np.linalg.svd(data)
sort_indices = np.argsort(-value)
I = np.eye(k)
sigma2 = sum(value[sort_indices[k:]]) / (p - k)
diag_sorted = np.diag(value[sort_indices[:k]])
W = vector_U[:, 0:k] @ ((diag_sorted - sigma2 * I) ** 0.5)
Z = np.zeros((k, m))
for i in range(m):
    Z[:, i:i + 1] = np.linalg.inv(W.T @ W + sigma2 * I) @ W.T @ (data[:, i:i + 1] - mu)
recon_data = (W @ Z + mu)
return Z, recon_data
```




（三）基于 EM 的标准 PCA

```
# 使用简化的 EM 算法计算人脸图像的低维表示
em_pca(data, k):
p, m = data.shape
# 初始化
W = np.random.randn(p, k)
Z = np.random.randn(k, m)
x_mean = np.mean(data, axis=1).reshape(p, 1)
for epoch in range(50):
    print(epoch)
    # E 步
    x_mean = np.mean(data, axis=1).reshape(p, 1)
    data = data - x_mean
    Z = np.linalg.inv(W.T @ W) @ W.T @ data
```

```python
        # M 步
        W = data @ Z.T @ np.linalg.inv(Z @ Z.T)
    recon_data = (W @ Z + x_mean)
    return Z, recon_data
```




（四）主函数调用

```python
def main():
    X = get_data()
    print(X.shape)
    # SVD
    data = svd_pca(X)
    x = data[5].reshape(112, 92)
    plt.title("SVD-200 维")
    plt.imshow(x, cmap='gray')
    plt.show()

    # 最大似然估计
    Z, recon_X = max_likelihood_estimation_pca(X, 300)
    x = recon_X[:, 0].reshape(112, 92)
    plt.title("最大似然估计-300 维")
    plt.imshow(x, cmap='gray')
    plt.show()

    # em
    Z, recon_X = em_pca(X, 300)
    x = recon_X[:, 0].reshape(112, 92)
    plt.title("EM-300 维")
    plt.imshow(x, cmap='gray')
    plt.show()
```