Yoav Shoshan

Theory Questions

(1) (a) we would like to find $\vec{w}$ s.t:

$X\vec{w} = \vec{y}$ $\quad \text{rank}(X) = n$

$X \in \mathbb{R}^{n \times d}$ $\quad n < d$

$y \in \mathbb{R}^n$

$w \in \mathbb{R}^d$

denote u_i : the i-th column vector in X. Then

$$X\vec{w} = \sum_{i=1}^{d} w_i \vec{u}_i$$

but within the set $\{u_i\}_{i=1}^{d}$, there exist n orthogonal vectors that can span a space of dimension n. Therfor, we are able to compose any vector $y \in \mathbb{R}^n$ with this linear combination and in particular $\vec{y} = X\vec{w}$. //

(b) we would like to solve

$$\min_w |w|^2 \longrightarrow \min_w \frac{1}{2}|w|^2$$

s.t. $Xw = y \rightarrow Xw - y = 0$

we define The Lagrangian:

$$\mathcal{L} = \frac{1}{2}|w|^2 - \sum_i \lambda_i (Xw - y)_i = \frac{1}{2}|w|^2 - \lambda^T (Xw - y)$$

$$\Rightarrow \frac{\partial f}{\partial w} = w - (\lambda^T X)^T \stackrel{!}{=} 0 \Rightarrow w = X^T \lambda$$

$\Rightarrow$ if we denote x_i the i-th column vector $\Rightarrow$ $w^* = \sum_{i=1}^{n} \lambda_i x_i$ $\quad \vec{\lambda} \in \mathbb{R}^n$, $\vec{x}_i \in \mathbb{R}^d$

and here we have n vectors that represent all the data points

(c) $v^T w^* = v^T X^T \lambda =$ $\quad \begin{pmatrix} v \in \mathbb{R}^d \\ x_i \in \mathbb{R}^d \\ \lambda \in \mathbb{R}^n \end{pmatrix}$

$$(v_1, v_2 \dots d) \begin{pmatrix} | & & | \\ \vec{x}_1 & \cdots & \vec{x}_n \\ | & & | \end{pmatrix} \vec{\lambda}$$

$$= (v^T x_1, v^T x_2 \dots v^T x_n) \vec{\lambda}$$

$$= (K(v, x_1), K(v, x_2) \dots K(v, x_n)) \vec{\lambda}$$

$$= \sum_{i=1}^{n} K(v, x_i) \lambda_i$$

(2) (a) $V_i = \phi(x_i) - \frac{1}{m}\sum_{t=1}^{m}\phi(x_t)$

$K'_{ij} = \langle V_i, V_j \rangle =$

$\left(\phi(x_i) - \frac{1}{m}\sum\phi(x_t)\right)^T\left(\phi(x_j) - \frac{1}{m}\sum\phi(x_t)\right)$

$= \phi(x_i)^T\phi(x_j) - \frac{1}{m}\sum_t\left(\phi(x_t)^T\phi(x_j) + \phi(x_i)^T\phi(x_t)\right)$

$+ \frac{1}{m^2}\sum_{tt'}\phi(x_t)^T\phi(x_{t'}) =$

$= \bar{K}_{ij} - \frac{1}{m}\sum_t(\bar{K}_{tj} + \bar{K}_{it}) + \frac{1}{m^2}\sum_{t,t'=1}^{m}\bar{K}_{tt'}$

(b) the covariance matrix in the dimension of $\phi(x)$:

$\Sigma = \frac{1}{m}\sum_{i=1}^{m}\phi(x_i)\phi(x_i)^T$

The set $\{u_j\}$ are the eigenvectors of the Σ, (principal components) thus:

$\Sigma u_j = \sigma_j u_j$ (σ_j is the eigenvalue of u_j)

$\left(\frac{1}{m}\sum_{i=1}^{m}\phi(x_i)\phi(x_i)^T\right)u_j = \sigma_j u_j$

Define $\alpha_{ji} = \frac{\phi(x_i)^T u_j}{m\sigma_j}$

$$\Rightarrow \quad u_j = \sum_{i=1}^{m} \alpha_{ji}\, \phi(x_i)$$

we have u_j as a linear combination of $\phi(x_i)$ where $\vec{\alpha}_j$ is the coeficients vector. in order to find α_j for a particulem j, we have to solve the equation

$$\sum u_j = \sigma_j u_j \qquad (\sigma_j \neq 0 \text{ for } pc) \Rightarrow$$

$$\left(\frac{1}{m}\sum_i \phi(x_i)\phi(x_i)^T\right)\left(\sum_k \alpha_{jk}\phi(x_k)\right) = \sigma_j \sum_i \alpha_{ji}\phi(x_i)$$

$$\frac{1}{m}\sum_i \phi(x_i)\sum_k \alpha_{jk}\phi(x_i)^T\phi(x_k) = \sigma_j \sum_i \alpha_{ji}\phi(x_i)$$

$$\frac{1}{m}\sum_i \phi(x_i)\sum_k \alpha_{jk} K(x_i, x_k) = \sigma_j \sum_i \alpha_{ji}\phi(x_i)$$

multiplying from left with some vector $\phi(x_l)^T$

$$\frac{1}{m}\sum_i \phi(x_l)^T\phi(x_i)\sum_k \alpha_{jk} K(x_i, x_k) = \sigma_j \sum_i \alpha_{ji}\phi(x_l)^T\phi(x_i)$$

$$\frac{1}{m}\sum_i K(x_l, x_i)\sum_k \alpha_{jk} K(x_i, x_k) = \sigma_j \sum_i \alpha_{ji} K(x_l, x_i)$$

$$\left(\frac{1}{m}\bar{\bar{K}}\bar{\bar{K}}\vec{\alpha}_j\right)_l = \sigma_j (\bar{\bar{K}}\vec{\alpha}_j)_l \qquad \forall l = 1 \dots m$$

$$\bar{\bar{K}}^2\vec{\alpha}_j = \bar{\bar{K}}\, m\sigma_j\vec{\alpha}_j \Rightarrow \bar{\bar{K}}\vec{\alpha}_j = m\sigma_j\vec{\alpha}_j$$

$\Rightarrow$ we have $\vec{\alpha}_j$ if we diagonalize $\bar{\bar{K}}$

(c) $\langle u_j, \phi(x') \rangle = u_j^T \phi(x') =$ (from b)

$$= \sum_{i=1}^{m} \alpha_{ji} \phi(x_i)^T \phi(x') = \sum_{i=1}^{m} \alpha_{ji} K(x_i, x')$$

So if we can calculat kernel we can project any $\phi(x')$ on the subspace sparced by $\{u_j\}_{j=1}^{k}$

If the kernels are calculated with complexity $O(d) \Rightarrow$
$\langle u_j, \phi(x') \rangle$ is calculated with $O(md)$

if we would like to project on the subspace of $\{u_j\}_{j=1}^{k}$ than the complexity becomes $O(kmd)$

(3) $X \sim \lambda e^{-\lambda x}$ $\quad L(\lambda) = \prod_{i=1}^{n} p(x_i)$

$$\log(L(\lambda)) = \sum_{i=1}^{n} \log p(x_i) = \sum_{i=1}^{n} \log \lambda - \lambda x_i$$

$$= n \log \lambda - \sum_{i=1}^{n} \lambda x_i$$

(a) $$0 \overset{!}{=} \frac{\partial \log(L\lambda)}{\partial \lambda} = \frac{n}{\lambda} - \sum_i x_i$$

$$\Longrightarrow \hat{\lambda} = \frac{n}{\sum_{i=1} x_i} = \frac{1}{\bar{x}}$$

(b) If we sample n (iid) values from exp. distribution with $\lambda = 1$ the postrior would then be

$$p(\Lambda = \lambda \mid X_1 = x_1, \ldots X_n = x_n) = \frac{p(\Lambda = \lambda) \prod_{i=1}^{n} p(X_i = x_i \mid \Lambda = \lambda)}{p(X_1 = x_1, \ldots X_n = x_n)}$$

$$\propto p(\Lambda = \lambda) \prod_{i=1}^{n} p(X_i = x_i \mid \Lambda = \lambda) = \quad * \; p(\Lambda = \lambda) \sim \exp(1)$$

$$e^{-\lambda} \prod_{i=1}^{n} \lambda e^{-\lambda x_i}$$

we can try maximizing the log of the postrior and then we are maximizing

$$\log\left(e^{-\lambda} \prod_{i=1}^{n} \lambda e^{-\lambda x_i}\right) = -\lambda + \sum_{i=1}^{n} \left(\log(\lambda) + \log(e^{-\lambda x_i})\right)$$

$$= -\lambda + n \log(\lambda) - \lambda \sum_i x_i$$

$$\Rightarrow \quad 0 = \frac{\partial}{\partial \lambda}\left(n \log \lambda - \lambda\left(\sum_{i=1}^{n} x_i - 1\right)\right) =$$

$$= \frac{n}{\lambda} - \sum_{i=1}^{n} x_i - 1$$

$$\Rightarrow \quad \lambda = \frac{n}{\sum_{i=1}^{n} x_i + 1}$$

Easy to see that this is indeed a maximum $\frac{\partial^2}{\partial \lambda^2}\left(P(\lambda = \lambda | X_1 = x_1, \ldots X_n = x_n)\right) < 0$

(c) For n samples and K possible values of λ, we have (θ - parameters of distributions, Z_i - the dist. type of i-th sample)

$$Q(\theta, \theta^t) = \sum_{i=1}^{n} \sum_{k=1}^{K} P(Z_i = k | X_i = x_i, \theta^t) \log\left(P(X_i = x_i, Z_i = k, \theta)\right)$$

Now, if $Z_i \sim U\{1, \ldots K\}$ we have ...

$$P(X_i = x_i, Z_i = z_i, \theta) = P(X_i = x_i | Z_i = k, \theta) P(Z_i = k, \theta)$$

$$= \lambda_k e^{-\lambda_k x_i} \cdot \frac{1}{K}$$

$$Q(\theta, \theta^t) = \sum_{i=1}^{n} \sum_{k=1}^{K} P(Z_i = k | X_i = x_i, \theta^t) \log\left(\frac{1}{K} \lambda_k e^{-\lambda_k x_i}\right)$$

keeping only terms dependent on λ_k $\sim$

$$\sum_{i=1}^{n} \sum_{k} \underbrace{P(Z_i = k | X_i = x_i, \theta^t)}_{\equiv C_{i,k}^t} (\log \lambda_k - \lambda_k x_i)$$

$$\sum_i \sum_k C_{i,k}^t (\log \lambda_k - \lambda_k x_i)$$

$\Rightarrow$ deriving with respect to λ_k

$$\frac{\partial Q(\theta, \theta^t)}{\partial \lambda_k} = \sum_i c_{i,k}^t \left(\frac{1}{\lambda_k} - x_i \right) \overset{!}{=} 0$$

$$\lambda_k = \sum_{i=1}^{n} c_{i,k}^{\tau} \cdot \frac{1}{\sum_{i=1}^{n} c_{i,k}^t x_i}$$

This is indeed a maximum:

$$\frac{\partial^2 Q}{\partial \lambda^2} = -\frac{1}{\lambda_k^2} \sum_i c_{i,k}^t < 0$$ //

(4) (a) Log likelihood:

$$\log\left(\prod_{i=1}^{n} P(X_i = x_i)\right) = \sum_{i=1}^{n} \log\left(P(X_i = x_i)\right) =$$

$$\sum_i \log\left(\sum_{r=0}^{31} P(X_i = x_i \mid Z_i = r) P(Z_i = r)\right) =$$

$$\sum_i \log\left(\sum_r \theta_r \underbrace{P(X_i = x_i \mid Z_i = r)}\right) = *$$

X is a 5-bit representation of Z_r, with 2 (out of 5) randomly selected digits replaced with "2".

So given x_i, each possible x_i has a chance of $\binom{5}{2}^{-1} = \frac{1}{10}$ to be selected

Define I_{Z_r, x_i} = Indicates if x_i could be built with Z_r and 2-digits replacements

$$= \begin{cases} 1 & Z_r \to x_i \text{ is a possible transformation} \\ 0 & \text{else.} \end{cases}$$

$$\overset{*}{=} \sum_i \log\left(\sum_r \theta_r I_{Z_r, x_i} \cdot \frac{1}{10}\right)$$

(b) We can think of a mixture of models with parameters $\{\theta_r\}_{r=0}^{31}$ and each sample x_i is generated from one of this models. for each EM step then, we have ($\theta \equiv \{\theta_r\}$)

$$Q(\theta, \theta^t) = \sum_{i=1}^{n} \sum_{r=0}^{31} P(z_i = z_r | x_i = x_i, \theta^t) \log(P(x_i = x_i | z_i = z_r, \theta))$$

$$\equiv \sum_i \sum_r C_{i,r}^t \log(P(x_i = x_i | z_i = z_r, \theta))$$

$$= \sum_i \sum_r C_{i,r}^t \log\left(\theta_r I_{z_r, x_i} \cdot \frac{1}{10}\right)$$

Now, we need to maximize $Q(\theta, \theta^t)$ given the constraint:

$\sum_r \theta_r = 1 \implies$ The Lagrangian:

$$\mathcal{L}(\theta, \lambda) = \sum_i \sum_r C_{i,r}^t \log\left(\frac{\theta_r}{10} I_{z_r, x_i}\right) - \lambda\left(\sum_{r=0}^{31} \theta_r - 1\right)$$

$$\frac{\partial \mathcal{L}}{\partial \theta_r} = \sum_i C_{i,r}^t \frac{1}{\frac{\theta_r}{10} I_{z_r, x_i}} \cdot \frac{I_{z_r, x_i}}{10} - \lambda$$

$$= \frac{1}{\theta_r} \sum_i C_{i,r}^t - \lambda \overset{!}{=} 0 \implies$$

$$\theta_r = \frac{1}{\lambda} \sum_i C_{i,r}^t$$

indeed a maximum $\frac{\partial^2 \mathcal{L}}{\partial \theta_r} = -\frac{1}{\theta_r} \sum_i C_{i,r}^t < 0$

λ could be found using the constraint:

$$1 = \sum_{r=0}^{31} \theta_r = \frac{1}{\lambda} \sum_{r=0}^{31} \sum_{i=1}^{n} C_{i,r}^{T} =$$

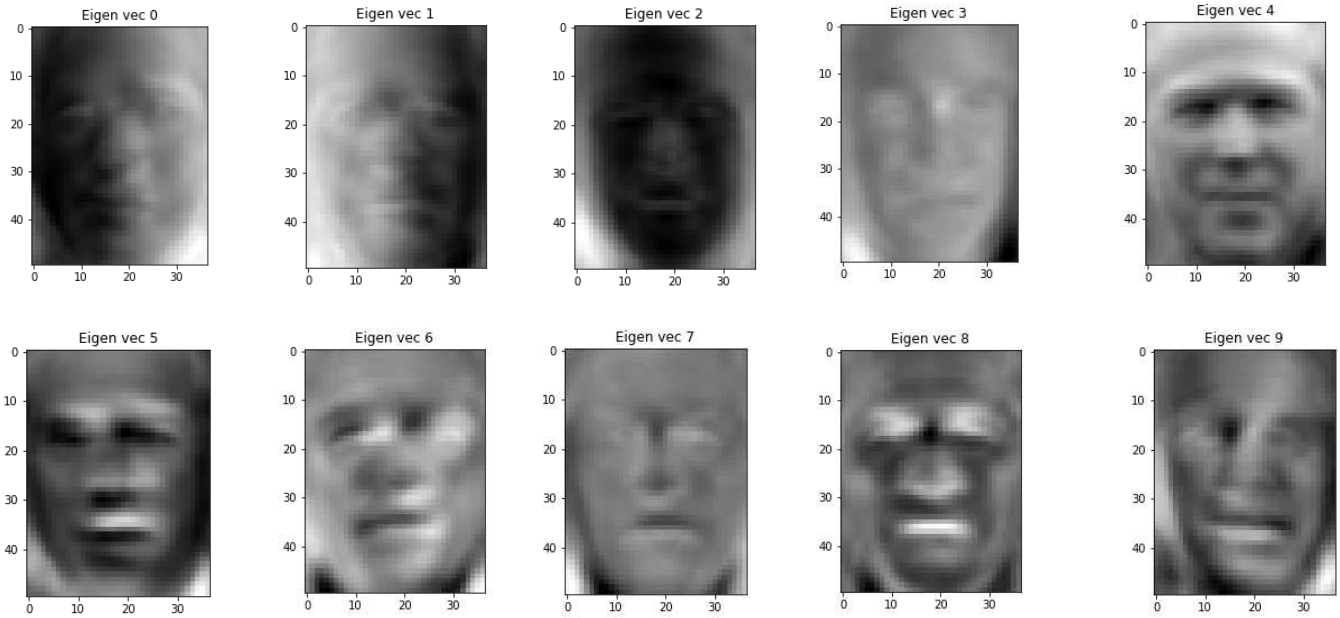
$$\frac{1}{\lambda} \underbrace{\sum_{i=1}^{n} \sum_{r=0}^{31} C_{i,r}^{t}}_{1} = \frac{n}{\lambda} \Longrightarrow \lambda = n$$

$$\Longrightarrow \theta_r = \frac{1}{n} \sum_{i} C_{i,r}^{t}$$

Programming assignment – PCA

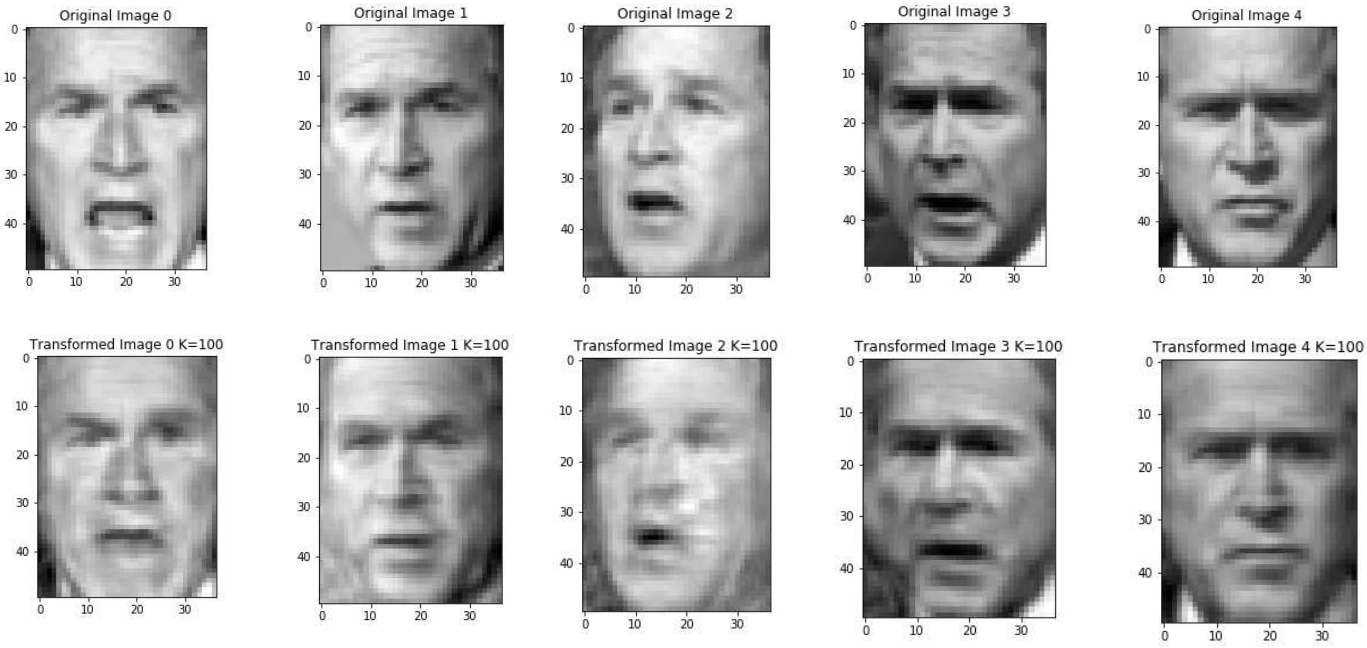
a. See scripts

b. 10 eigenvectors for George W Bush:

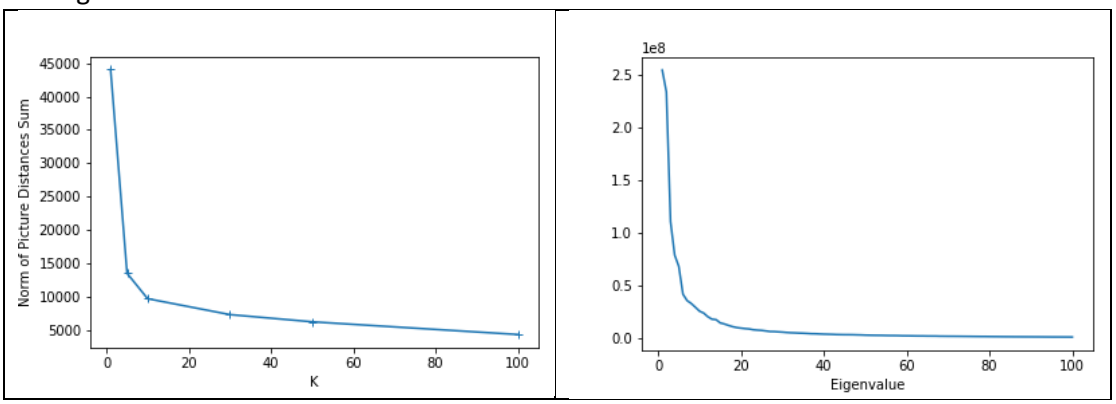


The Eigenvectors are linear combination of features (in our case, pixels) of all the samples. Those linear combinations can be viewed as new features which has the maximal variance across all samples. We can see that some of the eigenvectors give greater weights to pixels in the nose area and some to pixels in the eyes area in the pictures. Some also emphasize the shading of the pictures. each such “trait” represented by the eigenvectors is considered as a trait with high variance within the data (the original George W Bush pictures).

c. 5 Original Pictures (upper Pictures) vs Transformed Pictures for k = 100(lower pictures)

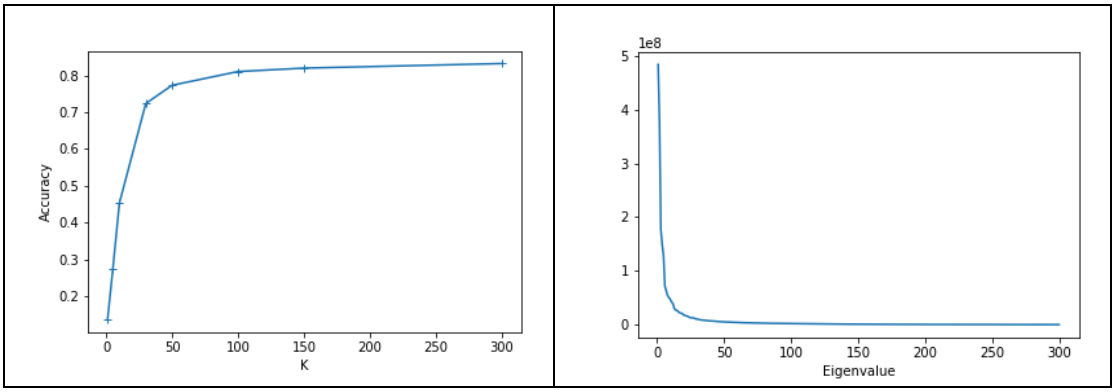


We can see that projecting the Pictures on a subspace spanned by 100 eigenvectors, we are still able to reconstruct the pictures so that they are recognizable by the human eye. For K<50 we did not had such remarkable results (the reconstructed pictures were of too poor quality). We can examine how closer the transformed pictures are to the original ones by looking at the sum of distances wrt k



We can see that the more eigenvectors we use, the closer the pictures are. But, we do reach a point from which adding more eigenvectors does not improve the distances as it did for smaller values of k. This is due to the fact that the eigenvalues themselves decrease with k and reach a value that is significantly smaller than the first few eigenvalues (see eigenvalues wrt k plot as well), meaning the data has smaller variance in those additional eigenvectors.

d. Test accuracy wrt k while using 75% of the whole data as training data and 25% as test data:



Here as well we can see that the accuracy increases with k as expected, but rach saturation around k=50. In this value of k we can also see that the eigenvectors reach saturation at a value which is much smaller than the first few eigenvalues. This means that adding more eigenvectors does not improve our ability to distinguish between pictures as it did in smaller k's because those eigenvectors represent smaller variance.