PCA Exercises Session

The Yale Extended dataset in folder yaleB01 is composed by 65 images of size 192×168. Use the function `preprocessing()` to transform the .pgm files into a matrix X where each column is made by the pixels of a single image. Use `preprocessing(n,m)` to resize the images to size $n \times m$ and insert them into the matrix X.

1 PCA

Algorithm 1 PCA(X, k), $X \in \mathbb{R}^{d\times n}$, $k > 0$

Compute $\mu = Xe/n$ and $\hat{X} = X - \mu e^T$
Compute the k-truncated SVD $\hat{X} = U\Sigma V^T$
Compute $Y = \Sigma V^T$

- Write a code to compute the PCA $X \sim \mu e^T + UY$ of X with input rank k through the k-truncated SVD of X.
- Apply PCA to the yaleB01 dataset. Draw the average face contained in μ and the first eigenfaces contained in the columns of U. Can you interpret what the eigenfaces represent?

Bonus Did you know that black and white images of faces usually lie in a 9-dimension subspace called the "harmonic plane"?

Let now X be the pixels of the single 640×480 image `yaleB17_P00A+000E+00.pgm`.

- Show the loss of accuracy of the approximation given by PCA on X with varying k by drawing the approximated image. Which k do you think is optimal?
- Plot the function $f(k) = k(640 + 480) + 10\|X - PCA(X, k)\|_F$. For what k it has its minimum?

2 IPCA

Let now W be a $d \times n$ binary matrix that has α% of entries equal to zero, randomly chosen among all entries. Reshape the images in the yaleB01 dataset to size 96×84 and then form the matrix X.

Algorithm 2 PI(W, X, k), $X \in \mathbb{R}^{d\times n}$, $W \in \{0,1\}^{d\times n}$, $k > 0$

Initialize $Y \in \mathbb{R}^{k\times n}$, $U \in \mathbb{R}^{d\times k}$ and $\mu \in \mathbb{R}^d$ randomly.
while $\|W \circ (X - \mu e^T - UY)\|_F$ does not converge **do**
  $\mu_i \leftarrow \frac{\sum_j w_{i,j}(X-UY)_{i,j}}{\sum_j w_{i,j}}$ for every i
  $u_i \leftarrow \left(\sum_j w_{i,j} y_j y_j^T\right)^{-1} \sum_j w_{i,j}(X - \mu e^T)_{i,j} y_j$ for every i
  Compute the slim QR decomposition $U = QR$ and set $U \leftarrow Q$
  $y_j \leftarrow \left(\sum_i w_{i,j} u_i u_i^T\right)^{-1} \sum_i w_{i,j}(X - \mu e^T)_{i,j} u_i$ for every j
end while
Compute $\mu \leftarrow \mu + UYe/n$, $Y \leftarrow Y(I - ee^T/n)$

- Write the code for the PI algorithm.
- For which α can you recover the original faces in X using PI algorithm and $k = 9$?
- Print the conditioning for the $k \times k$ matrices you invert in PI. How do they change with α?

3 RPCA

On the same set of images, the "illumination issues" can be seen as localized corruption of data. Reshape the images in the yaleB01 dataset to size 96×84 and then form the matrix X. We can thus apply a Robust PCA algorithm like IRLS or ADMM.

Algorithm 3 IRLS(X, k, ϵ), $X \in \mathbb{R}^{d\times n}$, $k > 0$, $\epsilon > 0$

Initialize $Y \in \mathbb{R}^{k\times n}$, $U \in \mathbb{R}^{d\times k}$ and $\mu \in \mathbb{R}^d$ randomly.
Initialize W as the $d \times n$ all ones matrix
while $\|W \circ (X - \mu e^T - UY)\|_F$ does not converge **do**
  $E \leftarrow X - \mu e^T - UY$
  $W_{i,j} = \epsilon^2/(E_{i,j}^2 + \epsilon^2)$ for every i, j
  Apply one step of PI with parameters W, Y, U, μ, X
end while
Compute $\mu \leftarrow \mu + UYe/n$, $Y \leftarrow Y(I - ee^T/n)$, $E \leftarrow X - \mu e^T - UY$

Recall here that

- $D_\tau(X)$ is the "singular values thresholding" of a matrix defined as: if $X = U\Sigma V^T$ is its SVD with singular values σ_i, then $D_\tau(X) = U\widetilde{\Sigma}V^T$ with singular values $\widetilde{\sigma}_i = \max\{0, \sigma_i - \tau\}$.
- $S_\tau(X)$ is the "soft thresholding" of a matrix defined as: $[S_\tau(X)]_{i,j} = \mathrm{sign}(X_{i,j}) \max\{0, |X_{i,j}| - \tau\}$

Algorithm 4 ADMM(X, λ, μ), $X \in \mathbb{R}^{d\times n}$, $\lambda, \mu > 0$

Initialize $E = \Lambda = L = 0$
while $\|L\|_\star + \lambda\|E\|_1 + \mathrm{Tr}(\Lambda^T(X - L - E) + \mu\|X - L - E\|_F^2/2)$ does not converge **do**
  $L \leftarrow D_{1/\mu}(\frac{1}{\mu}\Lambda + X - E)$
  $E \leftarrow S_{\lambda/\mu}(\frac{1}{\mu}\Lambda + X - L)$
  $\Lambda \leftarrow \Lambda + \mu(X - L - E)$
end while

- Write the code for the IRLS and ADMM algorithms.
- Apply IRLS with $k = 4$ and $\epsilon = 1/2$ to X. Can you interpret the faces in the columns of UY and the errors in the columns of E?
- Apply ADMM with $\mu = nd/(4\|X\|_1)$, $\lambda = 1/\sqrt{n}$ and compare with IRLS. Which one has the sparsest E? Compute moreover a truncated SVD of the output $L = U\Sigma V^T$ with rank $k = 4$ and compare the eigenfaces (columns of U) with the respective eigenfaces of IRLS.