Implementations

December 17, 2019

1 Compute PCs

Given a set of M training images (all of size $L \times N$), we can represent them by a set of $LN \times 1$ vectors: $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$. The computation of the PCs (eignefaces) is done by the following: First compute average face $\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$. Then construct covariance matrix C:

$$C = AAT$$

where $A = [\Gamma_1 - \Psi, \Gamma_2 - \Psi, \ldots, \Gamma_M - \Psi]$. Your paper used the notation $\Phi_i := \Gamma_i - \Psi$. The eigenfaces $u_i, i = 1, 2, \ldots, M$ is just a set of M orthonormal eigenvectors of C associated with non-zero eigenvalues.

2 Reconstructing an image using eigenfaces

Given any image Γ and a set of eigenfaces $u_1, u_2, \ldots, u_{M'}$, $M^{'} \leq M$. We can reconstruct this image as:

$$\hat{\Gamma} = \sum_{i=1}^{M^{'}} w_i u_i + \Psi$$

Where $w_i = u_i^T (\Gamma - \Psi)$

3 Performance of reconstruction using MSE

The MSE (mean-square-error) between the reconstructed image and original image is computed by

$$MSE = \frac{1}{NL} \sum_{i=1}^{L} \sum_{j=1}^{N} [\hat{\Gamma}(i,j) - \Gamma(i,j)]^2$$

where the images are assumed to be of size $L \times N$.

4 Efficient calculation of eigenfaces

Note that the size of C is $NL \times NL$. For our set of 190 training images, this matrix will be of size 31266×31266. But we only care about at most $M = 190$ eigenvectors of this matrix.

If one tries to compute the eigenvalues and eigenvectors of C directly, one will end up with 31266 eigenvalues, and at most 190 of those will be nonzero values (or not small values such as 1.32×10^{-7}). And that is assuming your MATLAB does not give you an "out of memory" error. So we seek an efficient way to compute the M eigenvectors we are interested in.

This is done by the following: consider an eigenvector ν_i and its corresponding eigenvalue μ_i for the matrix $A^T A$:

$$A^T A\nu_i = \mu_i\nu_i$$

If we multiply the above equation on both side from the left by the matrix A we will get:

$$AA^T A\nu_i = A\mu_i\nu_i$$

So the vector $A\nu_i$ will be an eigenvector of the matrix $C = AA^T$ and its associated eigenvalue remains to be μ_i. We know that for $\mu_i \neq \mu_j$, $A\nu_i and A\nu_j$ are orthogonal because for symmetric matrices, eigenvectors associated with different eigenvalues are orthogonal (but not necessarily of unit norm).

Note that the matrix $A^T A$ is of size $M \times M$. For our training set, 190×190. So instead of performing eigenvalue decomposition of the matrix C (which is of the size 31266×31266) and throw away the zero eigenvalues, we could perform eigenvalue decomposition on the matrix $A^T A$ (size 190×190), obtain a set of 190 orthonormal eigenvectors $\nu_1, \nu_2, \ldots, \nu_1 90$, and translate them to eigenvectors for C by

$$\nu_i = \frac{A\nu_i}{\|A\nu_i\|_2}, i = 1, 2, \ldots, M$$

The normalization in the above equation is important because the eigenfaces as defined in the paper need to be orthonormal. If you do not normalize the ν_i, you will be in trouble when trying to reconstruct faces.