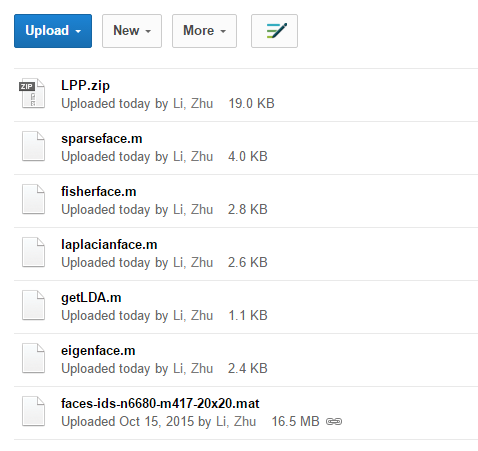
Homework # 3: Subspace Methods in Image Recognition

**Description:**

The objective of this homework is to consolidate and practice the knowledge and skills we learn on component analysis and subspace learning, especially the PCA, LDA and Graph Embedding. We will do an exercise based on a small face data set, which can be downloaded from:

https://umkc.box.com/s/p4levwprgjns5ctg7fne9stwu7twkpza

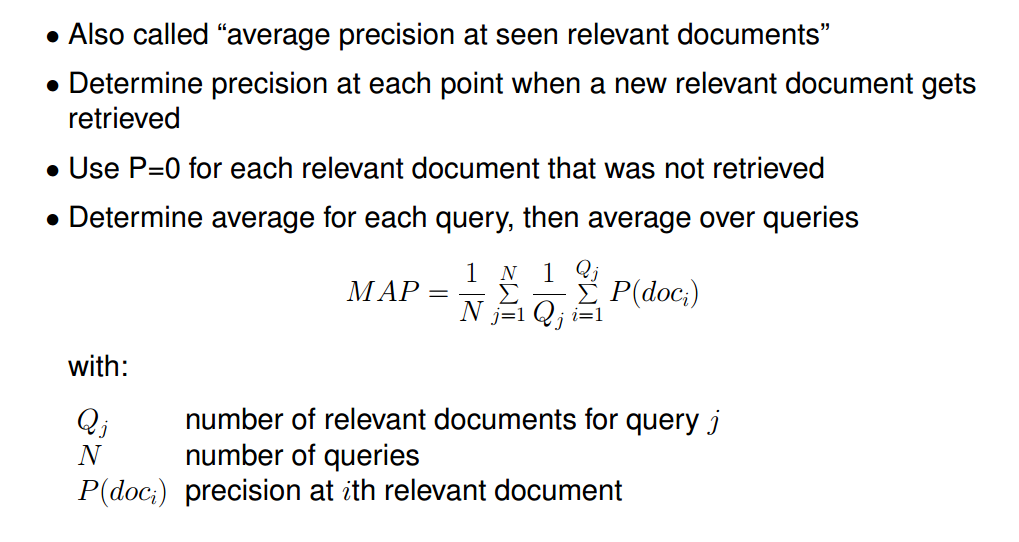


A face data set of 6680 images from 417 subjects, are provided, you should include a special group of 1 subject 10 images from any celebrity on web, to augment the data set.

The basic functions for PCA/LDA/Graph Embedding (LPP) are also provided, as in eigenface.m, fisherface.m, laplacianface.m, and the zipped folder LPP which offers Laplacian embedding solution LPP.m.

The input images are of size 20x20 pels, and students are required to compute the PCA/Eigenface model A0: 400 x d0, and LDA/Fisherface model on top of the Eigenface model, A1: d0 x d1, as well as a Laplacian embedding model, A2: d0xd2.

Then project the face images to these three models, and evaluate the performance as the mean average precision as discussed in the class. Notice that we will have total m=417+1=418 subjects, and we need to separate the training from the testing. For each class/subject, we can leave 1 image out for testing, and use the rest for training.



Hint: you just need to compute different index from faces and ids data, to separate training and querying, e.g, .

|  |
| --- |
| %loading faces, ids, of nx400 and nx1  load faces-ids-n6680-m417-20x20.mat;  n=length(ids);  uid = unique(ids); m = length(uid);  % query data index:  q\_indx = zeros(1, m);  for k=1:m  offs = find(ids==uid(k)); q\_indx(k) = offs(1);  end  %training data index  train\_indx = setdiff([1, n], q\_indx); |

**Deliverable and Grading:**

1. Extra data set: pull 10 images from 1 person somewhere (can be your own face images),normalize the images as 20x20 thumbnails, dynamic range scaling to [0, 1], and append it into the face/ids data set, give it id 999. [%10]
2. Matlab/Python Code and plots: function [A0, eigv]=getEigenfacemodel(faces) , plot the Eigenface basis, and corresponding eigen values, explain if you keep 8 dimensions, how much information/energy is lost ? [15%]
3. Matlab/Python Code and plots: function [A1]=getFisherfacemodel(faces, A0, ids) , compute Fisherface model after Eigenface projection (A0), to avoid singularity in covariance matrices, plot Fisherface models [20%]
4. Matlab/Python Code and plots: function [A2, S]=getLaplacianfacemodel(faces, A0, ids) , compute Graph Laplacian embedding model after Eigenface projection (A0), output the affinity matrix you computed as S, and plot the affinity matrix, plot the Laplacian face models, justify your choices of parameters in computing affinity matrix. [25%]
5. Matlab/Python code and plots: [map]=getQueryMAP(q, retrv\_ids), the function to compute the average precision from a single query, test this against your own data set (should consists at least 2000 face images), and then compute mAP for the new subjects you provided with id=999 . Please plot MAP as function of different choices of, eigenface model dimension of d0= [32, 64], and of fisherface/laplacian face dimension of d1/d2 = [8, 16, 24]. Also indicate your choice of heat kernel size and thresholding in affinity modeling for Laplacian. [%30]