CS189: Introduction to Machine Learning

Homework 6

Due: April 30, 2013

Problem 1: Eigenfaces

In this homework, you will implement the Eigenfaces algorithm to find celebrity look-alikes amongst a bunch of student faces. This simple approach works surprisingly well for face recognition and is also fairly fast.

You are provided three files as part of the homework.

- `CelebrityDatabase.zip` contains images of (aligned) faces of celebrities from which you will construct eigenfaces.
- `StudentDatabase.zip` contains images of aligned students' faces for which you will find the best matching celebrity faces.
- `mask.mat` contains a boolean mask that zeroes out the background in the face images. In order to keep only the relevant pixels in a face image and convert it into a vector execute:

  ```
  unmasked_pixels=find(mask); im_vector=im(unmasked_pixels);
  ```

  In order to obtain a masked image from `im_vector`, use:

  ```
  full_im = zeros(size(binary_mask));
  full_im(unmasked_pixels) = im_vector;
  ```

You need to do the following steps to find eigenfaces and the best matches. Note that you are NOT allowed to use any inbuilt functions for PCA in MATLAB.

1. Compute eigenfaces from the celebrity faces (after applying the binary mask). Visualize the top 10 eigenfaces. What kind of variations do the top eigenfaces seem to correspond to?

2. Reconstruct celebrity faces from only the top 10 eigenfaces and visualize some of them with the original image and the reconstructed image side-by-side. Plot the average L_2 error as a function of the number of eigenfaces used to reconstruct the original image.

3. For each image in `StudentDatabase.zip`, find the two nearest celebrity faces using Euclidean distance. Display the student face and the nearest celebrity faces side-by-side.

4. Repeat the above with Mahalanobis distance. Recall that the Mahalanobis distance of two vectors $\vec{x}$ and $\vec{y}$ under the data distribution with covariance matrix Σ is given as $d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}$. If $X = U\Sigma V^T$ (where X is the data matrix, what is the covariance matrix of the PCA transformed data? What is the justification for using this distance metric? Display the student face and the nearest celebrity faces side-by-side for this distance metric.

You need to include all the figures, numbers and answers to the questions in your report.

Problem 2: Consider the following 2×2 matrices:

$$A = \begin{bmatrix} 2 & 2 \\ 1 & -1 \end{bmatrix} \text{ and } A = \begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix}$$

For each of the above matrices, work out the following steps in order to compute the SVD of the above matrix.

1. Compute U by calculating eigenvectors AA^T.

2. Compute the entries of Σ by calculating the positive square roots of the eigenvalues of AA^T.

3. Compute V by calculating eigenvectors of $A^T A$.

4. Verify that $A = U\Sigma V^T$.