
COMP 435 Spring 2010

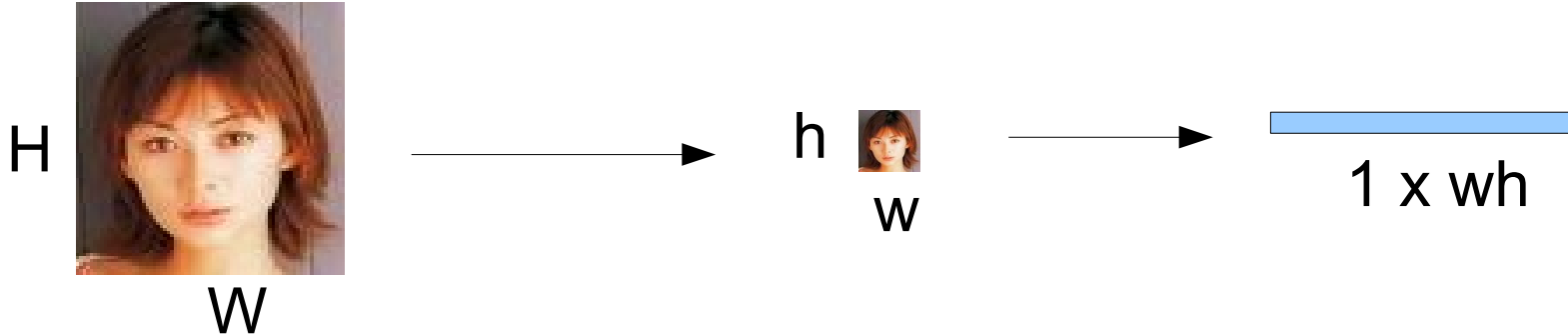
Tutorial on

**PCA and LDA in Face
Recognition**

Outline

- Show how to compute with matlab on concepts and techniques we just learnt:
 - PCA/Eigenface
 - LDA/Fisherface
 - Bayesian Classifier, linear and quadratic cases
 - K-NN classifier

Processing Face Images



- Face image of size $H \times W$, eg, 240×200 , is scaled first to $h \times w$, eg, 24×20
- Then it is converted to a vector by stacking rows in images.
 - In matlab: $v = \text{im}(:)'$; which is a row vector

Images to Icons to Vectors

- Matlab code:

```
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 % (1) Eigenface - PCA
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 if (doEigenFace)
15 % download the att_face.zip to the directory below
16 facepath='D:/zli/teaching/comp435/matlab/data/att_faces/';
17 nSubj=40; nPic=10;
18 % icon size
19 h=20; w=18;
20 %count
21 if (0)
22 t=1;
23 % create face icons
24 % load face images, re-size to icons of size h x w
25 for j=1:nSubj
26 for k=1:nPic
27 fim_name=sprintf('%s%d/%d.pgm', facepath, j, k);
28 im = imread(fim_name);
29 icn = imresize(im, [h, w], 'bilinear');
30 faces(t,1:w*h) = (icn(:))'; t=t+1;
31 end
32 end
33 if (1)
34 save face_icons_20x18.mat faces;
35 end
36 else % 2nd time load it directly
37 load face_icons_20x18.mat;
38 end
```

PCA

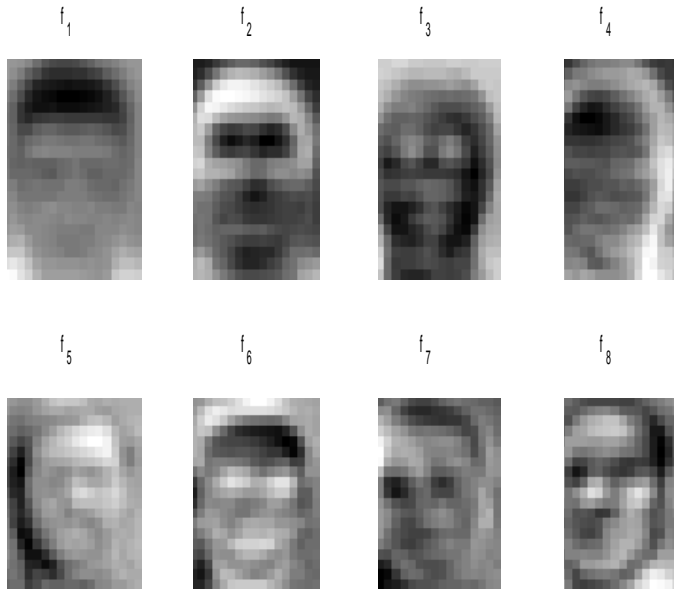
- For all face images stacked together into $n \times wh$ matrix, faces,
- Eigenface is computed by
 - Compute its covariance, $S = \text{cov}(\text{faces})$
 - Compute its eigenvectors and eigenvalues, $[A, L] = \text{eig}(S)$
 - Need some re-ordering by eigen values.

Matlab Code

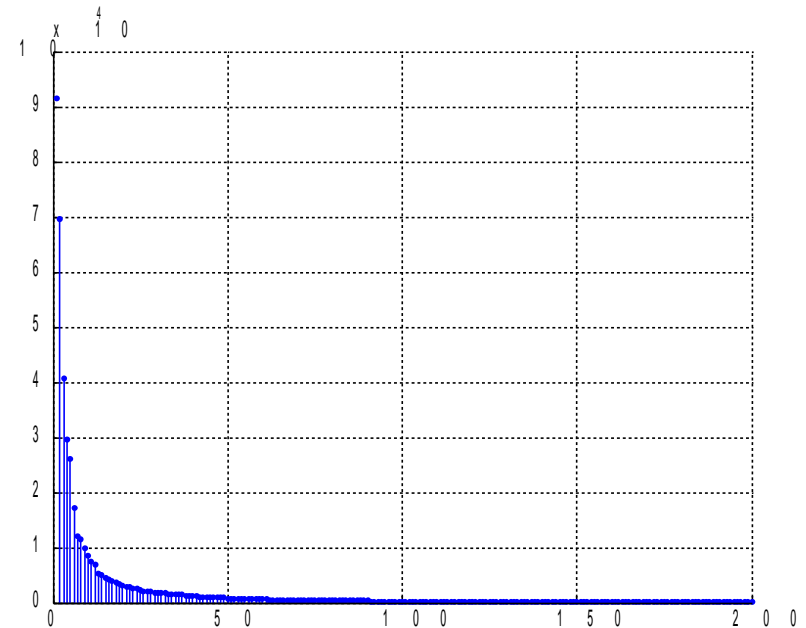
```
47 - load face_icons_20x18.mat;
48
49 - faces = double(faces);
50 - mfaces = mean(faces); mfaces2d = reshape(mfaces, h, w);
51 % covariance
52 - S = cov(faces);
53 % eigen value decomposition
54 - [EigFaces, L]=eig(S);
55 % notice that pca eigen values are sorted from small to big by default,
56 % have to correct that.
57 - [EigVal, indx] = sort(diag(L), 'descend');
58
59 - figure(1); stem(EigVal(1:180), '.'); title('PCA eigen values');
60 - figure(2);
61 - subplot(2,4, 1); colormap('gray'); imagesc(mfaces2d); title('mean f_0');
62 % pca dimensions we want
63 - m = 7;
64 - for k=1:m
65 -     subplot(2, 4, k+1);
66 -     eigf = EigFaces(:, indx(k));
67 -     eigf2 = reshape(eigf, h, w);
68 -     if (1)
69 -         colormap('gray'); imagesc(eigf2); axis off;
70 -     else
71 -         stem(eigf, '.');
72 -     end
73 -     str=sprintf('eigen f_%d', k'); title(str);
74 - end
```

Eigen Faces

- First 8 Eigenfaces:



Eigen Values



- **LDA is a supervised learning**
 - Need both the templates, faces
 - And the label, faceIds.
 - `[A, nClass]=myLDA(faces, faceIds);`

Classification

- Once faces projected via either PCA or LDA to some feature space X , then we can use either Bayesian or kNN classifier to recognize the label of unknown faces

```
172  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
173  % test classifiers
174  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
175 - [rec_labels1, err]=classify(qx, tx, tx_labels, 'linear');
176 - [rec_labels2, err]=classify(qx, tx, tx_labels, 'quadratic');
177 - [rec_labels3] = knnClassify(qx, tx, tx_labels);
178 - fprintf('\n Probe face ids: '); fprintf('%d ', qSubj);
179 - fprintf('\n Recognized ids, Bayesian-linear : '); fprintf('%d ', rec_labels1);
180 - fprintf('\n Recognized ids, Bayesian-quadratic : '); fprintf('%d ', rec_labels2);
181 - fprintf('\n Recognized ids, 1-NN classify : '); fprintf('%d ', rec_labels3);
182
```

Demo

- Open up your computer and matlab
- Let us walk thru `face_classify()`