

PCA Review

CS 510

February 25th, 2013

The logo for Colorado State University, featuring a green wavy line and a yellow wavy line.

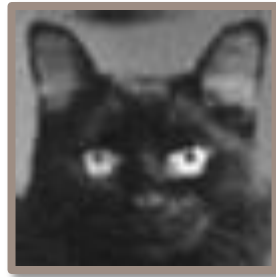
Colorado State University

Recall the goal: image matching



Probe image, registered to gallery

Registered Gallery of Images



Colorado State University

Getting practical...

- Every (2D) image is “stretched out” into a (1D) vector

$$x^i = [x_1^i, \dots, x_N^i]^T$$

- Typically, pixels arranged in scan-line order
 - But any fixed order will do
- Use superscripts to denote the image number, subscripts for dimensions
 - So there are a total of N pixels
- Images as column vectors



Colorado State University

Optional - Normalizing Images

- Give images zero mean & unit length.
 - If so, distance between images (points) is inversely proportional to correlation
 - If so, images (points) lie on N-1 dimensional hypersphere

$$0 = \sum_d x_d^i \quad 1 = \sum_d (x_d^i)^2$$

This step can certainly be omitted, in which case the correlation interpretation no longer applies.

Colorado State University

Mean centering the data set

- We want to origin of the coordinate system to be the average image in the set, so...

$$\bar{x}^i = x^i - m$$

$$m = \frac{1}{P} \sum_{i=1}^P x^i$$

- Zero mean images are no longer zero mean, but samples still lie on a hypersphere
- Center of mass is coordinate origin
- P is the number of images in the data set



Colorado State University

Forming X

- Now we create the data set matrix X :

$$\bar{X} = [\bar{x}^1 \mid \bar{x}^2 \mid \dots \mid \bar{x}^P]$$

- Note that columns are images, rows are dimensions
- X is an $N \times P$ matrix, num pixels x num images.

Covariance #1: XX^T

- Therefore we define the covariance matrix

$$\Omega = \overline{XX}^T$$

$$\Omega = \begin{bmatrix} \omega_{1,1} & \omega_{1,2} & \cdots \\ \omega_{2,1} & \omega_{2,2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

- Each term ω defines the covariance between two pixel dimensions across the data set



Colorado State University

PCA \equiv SVD(Ω)

- Perform singular value decomposition on the covariance matrix:

$$\Omega = V\Lambda V^T$$

$$\Omega V = \Lambda V$$

– Notation changes:

- $\Lambda \equiv$ diagonal matrix of Eigenvalues
- $V \equiv$ matrix of Eigenvectors (Eigenvectors in columns)



Colorado State University

PCA (II)

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots \\ 0 & \lambda_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

– λ 's are the Eigenvalues

$$V = [v_1 | v_2 | \dots | v_P]$$

– v 's are the Eigenvectors (unit length, orthogonal)



Colorado State University

What do we have?

- A data set of samples expressed as points
- The origin at the center of mass
- A set of Eigenvectors, describing the axes of maximum variance (maximum change)
 - Note that V is an orthonormal basis
- A set of Eigenvalues, giving the amount of variance along each Eigenvector
- At most $\min(N, P-1)$ non-zero Eigenvalues



Colorado State University

Data projection

- We can project any data sample x^i into the space defined by the Eigenvectors:

$$\tilde{x}^i = V^T \bar{x}^i \quad \leftarrow \text{Remember to first center this new point/image}$$

- This is just a geometric rotation, so we can easily get x^i back again:

$$\bar{x}^i = V \tilde{x}^i$$



Colorado State University

Data Compression

- But what about all the zero Eigenvalues?
 - If $P < N$, $\exists(N-(P-1))$ zero Eigenvectors
 - Probably more zeroes in practice.
 - Let K be the number of non-zero Eigenvalues.
- So we can drop all but K Eigenvectors

$$V = [v_1 | v_2 | \dots | v_K]$$

$$\tilde{x}^i = V^T \bar{x}^i$$

$$\bar{x}^i = V \tilde{x}^i$$

- Note that the projected vector is only K elements long!



Colorado State University

Data Compression (II)

- So the compressed representation has
 - $\sim KN$ values to store the Eigenvectors
 - PK values to store the compressed images
 - The original data was PN pixels
 - Data compression if $K < PN/(P+N)$
 - Further compression if you drop more Eigenvectors
 - Dropping small Eigenvalues results in small errors
 - Optimal compression in least squared sense (Sirovich&Kirby)

Data Matching - Review

- Assume a database of P images
 - (Optional) Zero mean and unit length each
 - Center the images to form X
 - Compute V & Λ
 - Drop zero Eigenvalues,
 - Project data

$$\tilde{X} = V^T \bar{X}$$



Colorado State University

Data Matching (II)

- Now, introduce a new probe image y
 - (Optional) Zero mean and unit length y
 - Subtract the data set mean m from y

$$\bar{y} = y - m$$

- Project Y into the Eigenspace

$$\tilde{y} = V^T \bar{y}$$

- Now find the closest x^i

$$match = \underset{i}{Min}(|\tilde{x}^i - \tilde{y}|)$$

- What image x^i do you have?
- How expensive was it to find?



Colorado State University

Data Matching (III)

- When all non-zero Eigenvectors kept, then
- For training images (bases for PCA)
 - The Euclidean distance between images in Eigen space is identical to Euclidean distance in the original image space.
- To the extent new images are “like” the training images, then
 - PCA matching is a cheap way compute Euclidean distance between many image pairs.
 - And, for zero mean, unit length images,
 - Image space and PCA space correlation the same.

Data Matching (IV)

- In practice, assumptions often violated.
 - If Eigenvectors associated with small (but non-zero) Eigenvalues are dropped, then minor dimensions are removed
 - May correspond to noise (or may not)
 - In practice, generally OK (more efficient, minimal damage)
 - Other distance measures often outperform Euclidean distance
 - “Why” is a non-trivial question.
 - Example: Whitenened cosine.
 - Open research topic

Covariance #2: $X^T X$

- We can also define a covariance matrix that defines how images (rather than dimensions) co-vary

$$\Omega = \bar{X}^T \bar{X}$$

- This is much smaller
 - P x P, instead of N x N
 - Order images squared, not pixels.

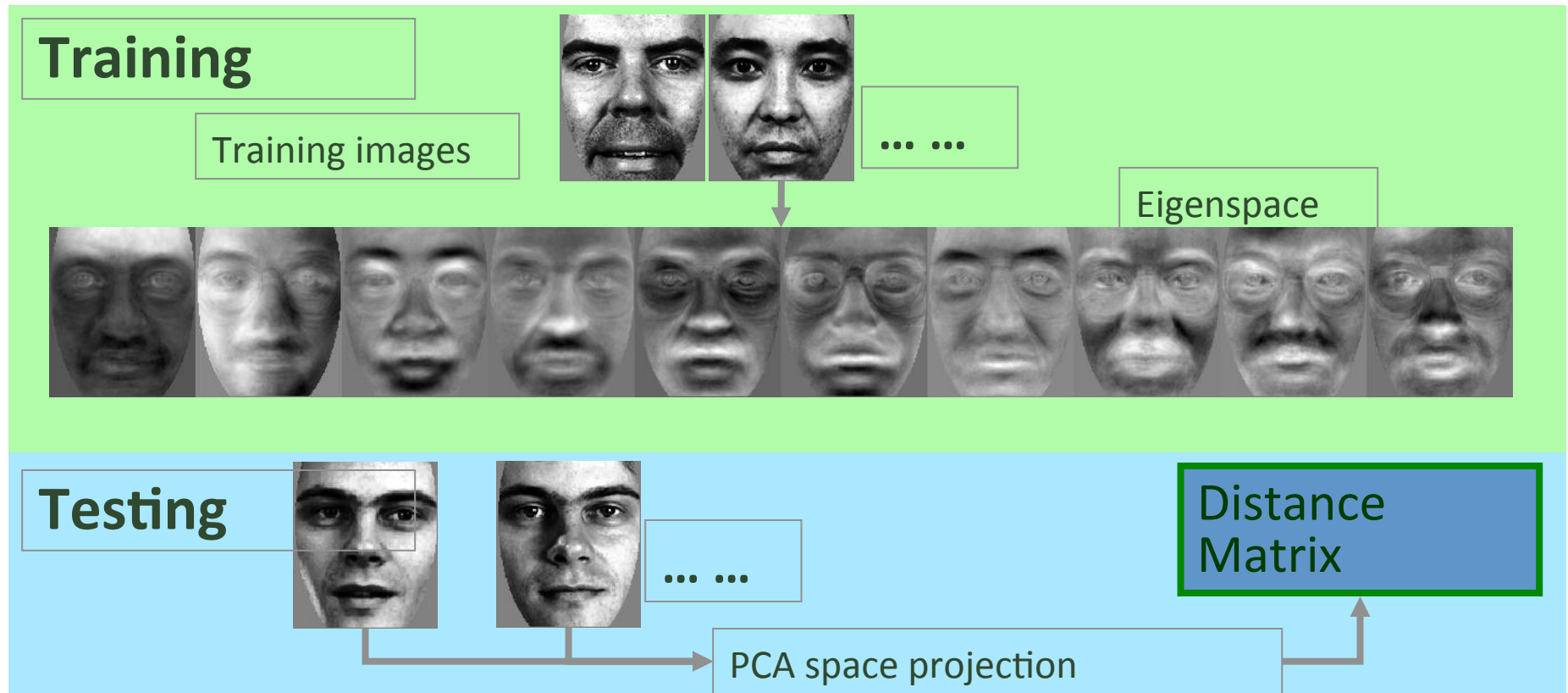
The Snapshot Method

- Linear algebra tells us that:
 - The Eigenvalues of XX^T and X^TX are the same
 - The Eigenvectors of XX^T are X times the Eigenvectors of X^TX (and re-normalized)
- Therefore, compute the Eigenvectors & Eigenvalues of the smaller X^TX to find the Eigenvectors for XX^T .

Snapshot (II)

- Create (centered) X as before.
- Create $\Omega' = X^T X$
- Compute $\Omega' = X^T X = V' \Lambda V'^T$
- Compute $V = \text{norm}(XV')$

Put it together - Eigenfaces



Colorado State University