PCA Review

CS 510

February 25th, 2013

Colorado State University

Recall the goal: image matching

Probe image, registered to gallery

Registered Gallery of Images

Getting practical…

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

$$x^i = \left[x_1^i, \ldots, x_N^i\right]^T$$

  - 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

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

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

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

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

Forming X

- Now we create the data set matrix X:

$$\overline{X} = \left[\overline{x}^1 \mid \overline{x}^2 \mid \ldots \mid \overline{x}^P \right]$$

  - Note that columns are images, rows are dimensions
  - X is an NxP matrix, num pixels x num images.

Covariance #1: XX^T

- Therefore we define the covariance matrix

$$\Omega = \overline{X}\,\overline{X}^T$$

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

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

Colorado State University

PCA ≡ SVD(Ω)

- Perform singular value decomposition on the covariance matrix:

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

$$\Omega V = \Lambda V$$

  – Notation changes:
    - Λ ≡ diagonal matrix of Eigenvalues
    - V ≡ 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

$$V = \left[v_1 \mid v_2 \mid \ldots \mid v_P \right]$$

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

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

Data projection

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

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

← *Remember to first center this new point/image*

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

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

Data Compression

- But what about all the zero Eigenvalues?
  - If P < N, ∃(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

$$V = \left[v_1 \mid v_2 \mid \ldots \mid v_K \right]$$

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

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

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

Data Compression (II)

- So the compressed representation has
  - ~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

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

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

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

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

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: Whitened 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

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

- This is much smaller
  - PxP, instead of NxN
  - Order images squared, not pixels.

Colorado State University

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.

Colorado State University

Snapshot (II)

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

Put it together - Eigenfaces