Dimensionality reduction

Pattern Recognition and Machine Learning - MuMeT 2017

Davide Abati

June 22th, 2017

University of Modena and Reggio Emilia

Agenda

Principal Component Analysis

Eigenfaces

Principal Component Analysis

PCA

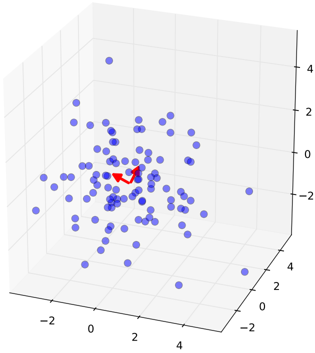
- Linear dimensionality reduction model
  - Subspace projection is linear
  - Reconstruction is linear
- Projects data in a new space subject to:
  - the variable having highest variance in the original space is projected on the first axis, the one having the second highest variance on the second axis, and so on.
  - axis of the new space are orthogonal (covariance is zero).

- Arrange your data in a $n \times d$ matrix X, where n is the number of samples and d is data dimensionality
- Compute the mean μ (d-dimensional vector) of all samples
- Compute convariance matrix

$$\Sigma = (X - \mu)^T (X - \mu)$$

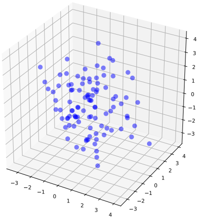
- Pick the first m eigenvectors of Σ (ordered by decreasing eigenvalues), where m is the dimensionality you want your data to be projected to
- Arrange such eigenvectors in a $d \times m$ matrix E
- Compute the projected samples as $P = X \cdot E$
- You can compute the reconstruction as $\tilde{X} = P \cdot E^T$

PCA: plotting components

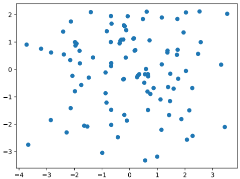


PCA: projecting and reconstructing (2D)

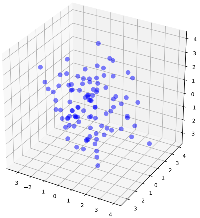
original data



projection

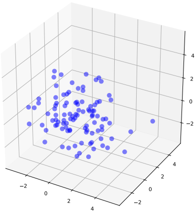


reconstruction

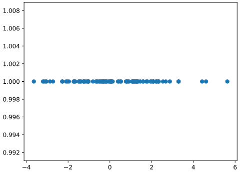


PCA: projecting and reconstructing (1D)

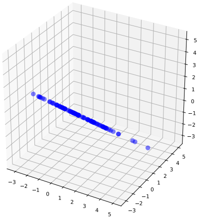
original data



projection



reconstruction



Eigenfaces

Eigenfaces

Famous algorithm for face recognition. Training is as simple as:

- load faces and annotations from the Olivetti dataset (datasets.get_faces_dataset takes care of loading and flattening images)

- Select a number of principal components and fit a PCA on training faces

To classify a test image:

- Project the image in the reduced spaces built in the training phase
- Perform **nearest neighbor classification**:
  - Roughly speaking,
    choose the class of the nearest training example (in the reduced space)

Each Olivetti image is 112×92. Once flattened, is a vector of 10304 pixels:

- The convariance matrix is 10304×10304
- Computing eigenvectors and eigenvalues is a pain
- Instead, compute the covariance matrix of transposed X:

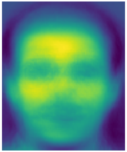
$$\Sigma = (X - \mu)(X - \mu)^T$$

- Once selected the principal components $\tilde{E}$ of this weirdo space, you can compute the original eigenvectors just like:

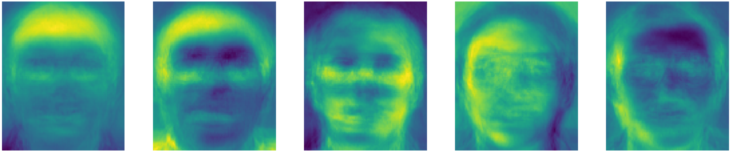
$$E = X^T \cdot \tilde{E}$$

Eigenfaces: some plots

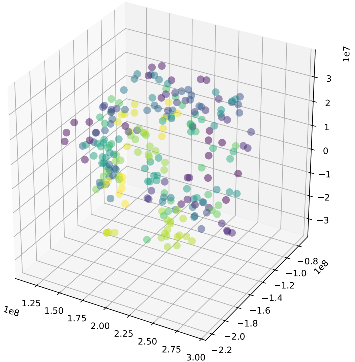
- Mean face:



- Eigenvectors:



Eigenfaces: face space



Eigenfaces: how many dimensions?