PCA

Ron Parr
CPS 271

Principle Components Analysis

- Idea:
  - Given data points in d-dimensional space, project into lower dimensional space while preserving as much information as possible
    - E.g., find best planar approximation to 3D data
    - E.g., find best planar approximation to 10^4 D data
  - In particular, choose projection that minimizes squared error in reconstructing original data

Why do we care?

- Lower dimensional representations permit
  - Compression
  - Noise filtering
- As preprocessing for classification
  - Reduces feature space dimension
    - Simpler Classifiers
    - Possibly better generalization
  - May facilitate simple (nearest neighbor) methods

Review of a Few Linear Algebra Facts

- A set of vectors is orthonormal if:
  - All vectors in the set have norm 1
  - Any two different vectors have dot-product 0
- Any vector in a linear space can be expressed as a weighted combination of norm 1 vectors – specifically, the vectors than form a basis for the space

PCA: Find Projections to Minimize Reconstruction Error

Assume data is set of d-dimensional vectors, $\mathbf{x}^n = \langle x_1^n \ldots x_d^n \rangle$

where n^{th} vector is $\mathbf{x}^n = \sum_{i=1}^{d} z_i^n \mathbf{u}_i; \quad \mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$

We can represent these in terms of any d orthonormal basis vectors

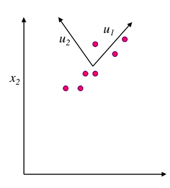
PCA: given M<d. Find $\langle \mathbf{u}_1 \ldots \mathbf{u}_M \rangle$

that minimizes $E_M \equiv \sum_{n=1}^{N} ||\mathbf{x}^n - \hat{\mathbf{x}}^n||^2$

where $\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^{M} z_i^n \mathbf{u}_i$

Mean

$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}^n$



Review: Eigenvectors

- Matrix A has eigenvector u with eigenvalue λ if:

$$Au = \lambda u$$

- For symmetric A (scaled) eigenvectors:
  - Are orthogonal
  - Have real eigenvalues
  - Form an orthonormal basis for A
  - (See appendix C)

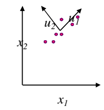
Review: Projection

- Orthonormal basis -> trivial projection
- Suppose U is our basis
  (formed by first k eigenvectors)
- Suppose we want to project a new x

$$w = (U^T U)^{-1} U^T x$$
$$= U^T x$$

- Note: We typically assume x has mean subtracted already

PCA



PCA: given M<d. Find $\langle \mathbf{u}_1 \ldots \mathbf{u}_M \rangle$

that minimizes $E_M \equiv \sum_{n=1}^{N} ||\mathbf{x}^n - \hat{\mathbf{x}}^n||^2$

where $\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^{M} z_i^n \mathbf{u}_i$

Note we get zero error if M=d.

Therefore, $E_M = \sum_{i=M+1}^{d} \sum_{n=1}^{N} [\mathbf{u}_i^T (\mathbf{x}^n - \bar{\mathbf{x}})]^2$

$$= \sum_{i=M+1}^{d} \mathbf{u}_i^T \Sigma \ \mathbf{u}_i$$

This minimized when $\boldsymbol{u}_i$ is eigenvector of Σ, i.e., when: $\Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i$

Covariance matrix: $\Sigma = \sum_n (\mathbf{x}^n - \bar{\mathbf{x}})(\mathbf{x}^n - \bar{\mathbf{x}})^T$

Equivalent problem: Maximize variance in the dimensions we keep

Justifying Use of Eigenvectors

- We want to minimize: $u^T \Sigma u$
- Subject to: $u^T u = 1$
- Use Lagrange Multipliers to minimize:

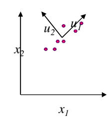
$$u^T \Sigma u - \lambda u^T u$$

- Take the gradient, set to 0:

$$\Sigma u - \lambda' u = 0$$

- True when we use eigenvalues, vectors

PCA



Minimize $E_M = \sum_{i=M+1}^{d} \mathbf{u}_i^T \Sigma \ \mathbf{u}_i$

$\rightarrow \quad \Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i$ Eigenvector of Σ

Eigenvalue

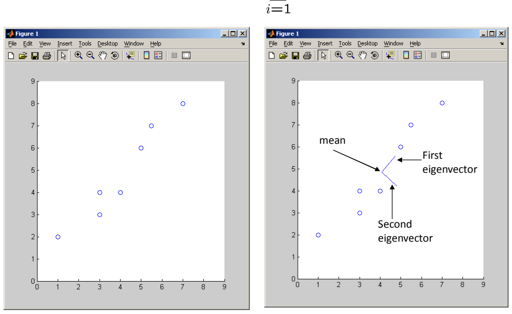
$\rightarrow \quad E_M = \sum_{i=M+1}^{d} \lambda_i$

PCA algorithm 1:

1. X ← Create N x d data matrix, with one row vector x^n per data point
2. A ← subtract mean $\bar{x}$ from each row vector x^n in X
3. Σ ← covariance matrix of A
4. Find eigenvectors and eigenvalues of Σ
5. PC's ← the M eigenvectors with largest eigenvalues

PCA Example

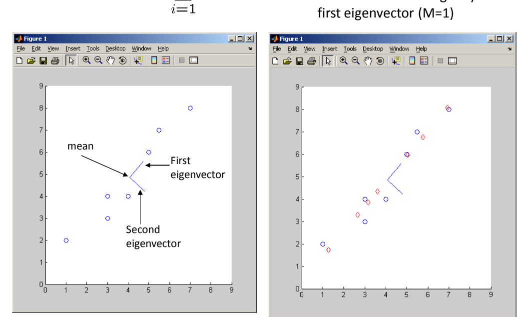
$$\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^{M} z_i^n \mathbf{u}_i$$



PCA Example

$$\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^{M} z_i^n \mathbf{u}_i$$

Reconstructed data using only first eigenvector (M=1)



Applying PCA

- Example data set: Images of faces
  - (Famous Eigenface approach [Turk & Pentland], [Sirovich & Kirby])
- Each datum is a point in image space
- Each point vector of luminance values
- Vectors are long, e.g., 256x256=64K
- These form columns of A, $\Sigma = AA^T$
- Problem: AA^T is unreasonably large!

A Clever Workaround

- Note that N<<d(=64K)
- Use $L=A^TA$ instead of $\Sigma = AA^T$
- Suppose v is eigenvector of L
- Av is eigenvector of Σ

$$Lv = \gamma v$$
$$A^T A v = \gamma v$$
$$AA^T A v = \gamma A v$$
$$\Sigma(Av) = \gamma(Av)$$

Application to Eigenfaces

- m=hundreds-thousands of faces
- Keep k~m/10 eigenvectors (eigenfaces)
- Achieve:
  - Low reconstruction error
  - Relatively high classification accuracy (across faces)
  - Robust measure of faceness
- Example:
  http://www.cs.princeton.edu/~cdecoro/eigenfaces/

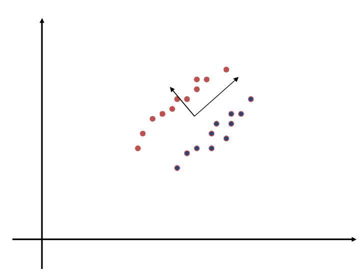
Summary of PCA Uses

- Data compression
  (compress data by representing entire data set as coefficients for a small number of principle components)
- Noise filtering
  (assume low eigenvalue components correspond to noise)
- Feature selection for supervised learning
  (assumes low eigenvalue components are noise/irrelevant features)
- Nearest neighbor classification
  (assumes subpace of principle components is a more natural space in which to measure distances)
- Direct classification
  (assume distance to span of principle components is an indicator of class membership)
- Visualization
  (assume the first 2 or 3 principle components show the interesting relationships that exist in the data)

Shortcomings

- Requires carefully controlled data: (for example)
  - All faces centered in frame
  - Same size
  - Some sensitivity to angle
- Completely knowledge free method
  - (sometimes this is good)
  - Doesn't know that faces are wrapped around 3D objects (heads)
  - Makes no effort to preserve class distinctions

PCA Problem Data Set



PCA doesn't know about labels!

PCA Conclusions

- PCA finds orthonormal basis for data
- Sorts dimensions in order of importance
- Discard low significance dimensions to:
  - Get compact description
  - Ignore noise
  - Improve classification (hopefully)
- Not magic:
  - Doesn't know class labels
  - Can only capture linear variations
- One of many types of dimensionality reduction!