Imperial College London

Lecture 14: Dimensionality Reduction with PCA

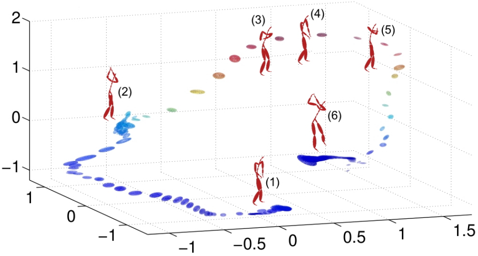
Recommended reading:
Bishop, Chapter 12.1

Duncan Gillies and Marc Deisenroth

Department of Computing
Imperial College London

17 February 2016

Motivation



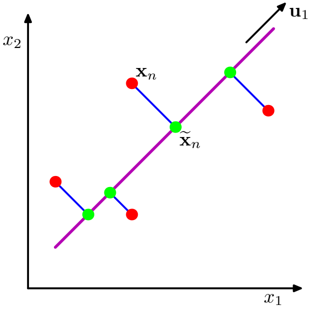
3-dimensional representation of 18-dimensional motion capture data (Deisenroth & Mohamed, 2012)

- High dimensional real data often possesses a lower intrinsic dimensionality ⏩ Easier to work with
- Dimensionality reduction: Find this lower dimensional representation
- Visualization
- Data compression

Key Idea of Dimensionality Reduction

- Project data onto a lower-dimensional manifold that preserves as much information as possible
- Think of it as data compression
- Principal Component Analysis (PCA): Find a (linear) projection that
  - Minimizes reconstruction error (Pearson, 1901)
  - Maximizes the variance (signal) of the projected data (Hotelling, 1933)
  - Maximize the mutual information between original and projected data (Linsker 1988)

Illustration: Orthogonal Projection



From PRML (Bishop, 2006)

- Two-dimensional data $\boldsymbol{x} = [x_1, x_2]^\top$ projected onto a one-dimensional linear manifold (affine subspace) with direction $\boldsymbol{u}_1$.
- Red: Original data, Green: Projected data

Refresher: Orthogonal Projection onto Sub-Vectorspaces

- Basis $\boldsymbol{u}_1, \ldots, u_M$ of a subspace $A \subset \mathbb{R}^D$
- Define $\boldsymbol{U} = [\boldsymbol{u}_1|...|\boldsymbol{u}_M] \in \mathbb{R}^{D\times M}$
- Project $\boldsymbol{x} \in \mathbb{R}^D$ onto subspace A:

$$\boldsymbol{U}(\boldsymbol{U}^\top \boldsymbol{U})^{-1}\boldsymbol{U}^\top \boldsymbol{x}$$

- If $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$ form an orthonormal basis ($\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$), then the projection simplifies to

$$\boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{x}$$

How to do it...

- Objective: Find orthogonal projection that minimizes the overall projection error

$$J = \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$$

where $\tilde{\boldsymbol{x}}_n$ is the projection of $\boldsymbol{x}_n$

Derivation (1)

- Define orthonormal basis of $\mathbb{R}^D = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D]$, such that $\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$

Derivation (1)

- Define orthonormal basis of $\mathbb{R}^D = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D]$, such that $\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$
- Then, every data point $\boldsymbol{x}_n$ can be written as a linear combination of the basis vectors:

$$\boldsymbol{x}_n = \sum_{i=1}^{D} \alpha_{ni} \boldsymbol{u}_i$$

Derivation (1)

- Define orthonormal basis of $\mathbb{R}^D = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D]$, such that $\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$
- Then, every data point $\boldsymbol{x}_n$ can be written as a linear combination of the basis vectors:

$$\boldsymbol{x}_n = \sum_{i=1}^{D} \alpha_{ni} \boldsymbol{u}_i$$

▸▸ Rotation of the standard coordinates to a new coordinate system defined by the basis $[\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D]$.

Derivation (1)

- Define orthonormal basis of $\mathbb{R}^D = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D]$, such that $\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$
- Then, every data point $\boldsymbol{x}_n$ can be written as a linear combination of the basis vectors:

$$\boldsymbol{x}_n = \sum_{i=1}^{D} \alpha_{ni} \boldsymbol{u}_i$$

▶▶ Rotation of the standard coordinates to a new coordinate system defined by the basis $[\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D]$.

▶▶ Original coordinates x_{ni} are replaced by α_{ni}, $i = 1, \ldots, D$

Derivation (1)

- Define orthonormal basis of $\mathbb{R}^D = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D]$, such that $\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$
- Then, every data point $\boldsymbol{x}_n$ can be written as a linear combination of the basis vectors:

$$\boldsymbol{x}_n = \sum_{i=1}^{D} \alpha_{ni} \boldsymbol{u}_i$$

  ⏩ Rotation of the standard coordinates to a new coordinate system defined by the basis $[\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D]$.

  ⏩ Original coordinates x_{ni} are replaced by α_{ni}, $i = 1, \ldots, D$
- Exploit orthonormality of $\boldsymbol{u}_i$ and obtain $\alpha_{nj} = \boldsymbol{x}_n^\top \boldsymbol{u}_j$, such that

$$\boldsymbol{x}_n = \sum_{i=1}^{D} (\boldsymbol{x}_n^\top \boldsymbol{u}_i) \boldsymbol{u}_i$$

Derivation (2)

Objective

Approximate

$$\boldsymbol{x}_n = \sum_{i=1}^{D} (\boldsymbol{x}_n^\top \boldsymbol{u}_i)\boldsymbol{u}_i$$

using a $M \ll D$ many basis vectors

▶▶ Projection onto a lower-dimensional subspace

Derivation (2)

Objective

Approximate

$$\boldsymbol{x}_n = \sum_{i=1}^{D} (\boldsymbol{x}_n^\top \boldsymbol{u}_i)\boldsymbol{u}_i$$

using a $M \ll D$ many basis vectors

▶▶ Projection onto a lower-dimensional subspace

- Lower-dimensional subspace of dimension M can be represented by $M \ll D$ basis vectors, such that

$$\tilde{\boldsymbol{x}}_n = \underbrace{\sum_{i=1}^{M} z_{ni}\boldsymbol{u}_i}_{\text{lower-dim. subspace}} + \underbrace{\sum_{i=M+1}^{D} b_i\boldsymbol{u}_i}_{\text{rest}}$$

Derivation (3)

$$\tilde{\boldsymbol{x}}_n = \underbrace{\sum_{i=1}^{M} z_{ni}\boldsymbol{u}_i}_{\text{lower-dim. subspace}} + \underbrace{\sum_{i=M+1}^{D} b_i\boldsymbol{u}_i}_{\text{rest}}$$

Derivation (3)

$$\tilde{\boldsymbol{x}}_n = \underbrace{\sum_{i=1}^{M} z_{ni}\boldsymbol{u}_i}_{\text{lower-dim. subspace}} + \underbrace{\sum_{i=M+1}^{D} b_i\boldsymbol{u}_i}_{\text{rest}}$$

- Choose z_{ni}, $\boldsymbol{u}_i$, b_i such that the squared reconstruction error

$$J = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$$

is minimized

Derivation (3)

$$\tilde{\boldsymbol{x}}_n = \underbrace{\sum_{i=1}^{M} z_{ni}\boldsymbol{u}_i}_{\text{lower-dim. subspace}} + \underbrace{\sum_{i=M+1}^{D} b_i\boldsymbol{u}_i}_{\text{rest}}$$

- Choose z_{ni}, $\boldsymbol{u}_i$, b_i such that the squared reconstruction error

$$J = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$$

  is minimized

▸▸ Compute gradients of J w.r.t. all variables

Derivation (4)

Necessary condition for optimum:

$$\frac{\partial J}{\partial z_{ni}} = 0 \quad \Rightarrow \quad z_{ni} = \boldsymbol{x}_n^\top \boldsymbol{u}_i \,, \qquad i = 1, \ldots, M$$

$$\frac{\partial J}{\partial b_i} = 0 \quad \Rightarrow \quad b_i = \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i \,, \qquad i = M+1, \ldots, D$$

Derivation (4)

Necessary condition for optimum:

$$\frac{\partial J}{\partial z_{ni}} = 0 \quad \Rightarrow \quad z_{ni} = \boldsymbol{x}_n^\top \boldsymbol{u}_i \,, \qquad i = 1, \dots, M$$

$$\frac{\partial J}{\partial b_i} = 0 \quad \Rightarrow \quad b_i = \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i \,, \qquad i = M+1, \dots, D$$

Then, the approximation error only plays a role in dimensions $M+1, \dots, D$:

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} \big((\boldsymbol{x}_n - \mathbb{E}[\boldsymbol{x}])^\top \boldsymbol{u}_i \big) \boldsymbol{u}_i$$

Derivation (4)

Necessary condition for optimum:

$$\frac{\partial J}{\partial z_{ni}} = 0 \quad \Rightarrow \quad z_{ni} = \boldsymbol{x}_n^\top \boldsymbol{u}_i\,, \qquad i = 1, \dots, M$$

$$\frac{\partial J}{\partial b_i} = 0 \quad \Rightarrow \quad b_i = \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i\,, \qquad i = M+1, \dots, D$$

Then, the approximation error only plays a role in dimensions $M+1, \dots, D$:

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} \big((\boldsymbol{x}_n - \mathbb{E}[\boldsymbol{x}])^\top \boldsymbol{u}_i\big)\boldsymbol{u}_i$$

▸▸ Displacement vector $\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n$ lies in space orthogonal to the principal subspace (linear combination of the $\boldsymbol{u}_i$ for $i = M+1, \dots, D$)

Derivation (4)

Necessary condition for optimum:

$$\frac{\partial J}{\partial z_{ni}} = 0 \quad \Rightarrow \quad z_{ni} = \boldsymbol{x}_n^\top \boldsymbol{u}_i\,, \qquad i = 1, \ldots, M$$

$$\frac{\partial J}{\partial b_i} = 0 \quad \Rightarrow \quad b_i = \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i\,, \qquad i = M+1, \ldots, D$$

Then, the approximation error only plays a role in dimensions $M+1, \ldots, D$:

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} \big((\boldsymbol{x}_n - \mathbb{E}[\boldsymbol{x}])^\top \boldsymbol{u}_i\big)\boldsymbol{u}_i$$

▶▶ Displacement vector $\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n$ lies in space orthogonal to the principal subspace (linear combination of the $\boldsymbol{u}_i$ for $i = M+1, \ldots, D$)

▶▶ Minimum error is given by the orthogonal projection of $\boldsymbol{x}_n$ onto the principal subspace spanned by $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$

Derivation (5)

From the previous slide:

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} (\boldsymbol{x}_n^\top \boldsymbol{u}_i - \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i)\boldsymbol{u}_i$$

Derivation (5)

From the previous slide:

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} (\boldsymbol{x}_n^\top \boldsymbol{u}_i - \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i)\boldsymbol{u}_i$$

Let's compute our reconstruction error:

$$J = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 = \frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)^\top (\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)$$

Derivation (5)

From the previous slide:

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} (\boldsymbol{x}_n^\top \boldsymbol{u}_i - \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i)\boldsymbol{u}_i$$

Let's compute our reconstruction error:

$$J = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 = \frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)^\top (\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)$$

$$= \frac{1}{N}\sum_{n=1}^{N} \sum_{i=M+1}^{D} (\boldsymbol{x}_n^\top \boldsymbol{u}_i - \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i)^2$$

Derivation (5)

From the previous slide:

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} (\boldsymbol{x}_n^\top \boldsymbol{u}_i - \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i)\boldsymbol{u}_i$$

Let's compute our reconstruction error:

$$J = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 = \frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)^\top(\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{i=M+1}^{D} (\boldsymbol{x}_n^\top \boldsymbol{u}_i - \mathbb{E}[\boldsymbol{x}]^\top \boldsymbol{u}_i)^2$$

$$= \sum_{i=M+1}^{D} \boldsymbol{u}_i^\top \boldsymbol{S} \boldsymbol{u}_i$$

where $\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x}_n - \mathbb{E}[\boldsymbol{x}])^\top$ is the data covariance matrix

Derivation (6)

- What remains: Minimize J w.r.t. $\boldsymbol{u}_i$ under the constraint that the $\boldsymbol{u}_i$ form an orthonormal basis.

Derivation (6)

- What remains: Minimize J w.r.t. $\boldsymbol{u}_i$ under the constraint that the $\boldsymbol{u}_i$ form an orthonormal basis.

Example:

- $M = 1, D = 2$
- Choose basis vector $\boldsymbol{u}_2$ such that $\boldsymbol{u}_2^\top \boldsymbol{S} \boldsymbol{u}_2$ is minimized and $\boldsymbol{u}_2^\top \boldsymbol{u}_2 = 1$

Derivation (6)

- What remains: Minimize J w.r.t. $\boldsymbol{u}_i$ under the constraint that the $\boldsymbol{u}_i$ form an orthonormal basis.

Example:

- $M = 1, D = 2$
- Choose basis vector $\boldsymbol{u}_2$ such that $\boldsymbol{u}_2^\top \boldsymbol{S}\boldsymbol{u}_2$ is minimized and $\boldsymbol{u}_2^\top \boldsymbol{u}_2 = 1$
- Constrained optimization yields (with Lagrange multiplier)

$$\tilde{J} = \boldsymbol{u}_2^\top \boldsymbol{S}\boldsymbol{u}_2 + \lambda(1 - \boldsymbol{u}_2^\top \boldsymbol{u}_2)$$

$$\Rightarrow \frac{\partial \tilde{J}}{\partial \boldsymbol{u}_2} = \boldsymbol{0} \Leftrightarrow \boldsymbol{S}\boldsymbol{u}_2 = \lambda \boldsymbol{u}_2$$

Derivation (6)

- What remains: Minimize J w.r.t. $\boldsymbol{u}_i$ under the constraint that the $\boldsymbol{u}_i$ form an orthonormal basis.

Example:

- $M = 1, D = 2$
- Choose basis vector $\boldsymbol{u}_2$ such that $\boldsymbol{u}_2^\top \boldsymbol{S}\boldsymbol{u}_2$ is minimized and $\boldsymbol{u}_2^\top \boldsymbol{u}_2 = 1$
- Constrained optimization yields (with Lagrange multiplier)

$$\tilde{J} = \boldsymbol{u}_2^\top \boldsymbol{S}\boldsymbol{u}_2 + \lambda(1 - \boldsymbol{u}_2^\top \boldsymbol{u}_2)$$

$$\Rightarrow \frac{\partial \tilde{J}}{\partial \boldsymbol{u}_2} = \boldsymbol{0} \Leftrightarrow \boldsymbol{S}\boldsymbol{u}_2 = \lambda \boldsymbol{u}_2$$

- **Eigenvalue problem**

Derivation (7)

- In general (arbitrary D and $M < D$), we need solve

$$\boldsymbol{S}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \,, \qquad i = 1, \dots, D$$

which requires finding the eigenvectors $\boldsymbol{u}_i$ of the data covariance matrix $\boldsymbol{S}$

Derivation (7)

- In general (arbitrary D and $M < D$), we need solve

$$\boldsymbol{S}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i\,, \qquad i = 1, \ldots, D$$

  which requires finding the eigenvectors $\boldsymbol{u}_i$ of the data covariance matrix $\boldsymbol{S}$

- Corresponding value of the squared reconstruction error:

$$J = \sum_{i=M+1}^{D} \lambda_i$$

  i.e., the sum of the eigenvalues associated with eigenvectors not in the principle subspace

Derivation (7)

- In general (arbitrary D and $M < D$), we need solve

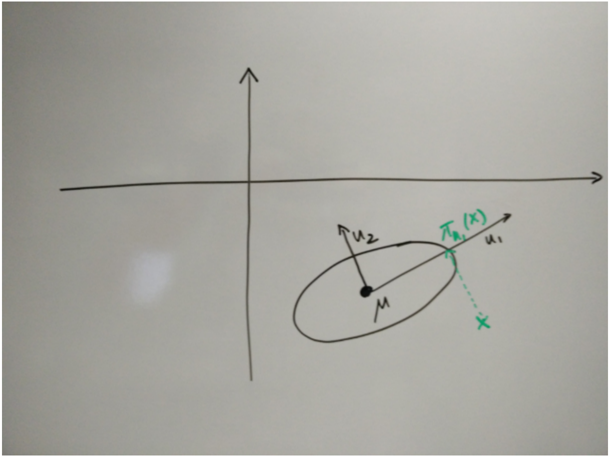
$$\boldsymbol{S}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i\,, \qquad i = 1, \ldots, D$$

  which requires finding the eigenvectors $\boldsymbol{u}_i$ of the data covariance matrix $\boldsymbol{S}$
- Corresponding value of the squared reconstruction error:

$$J = \sum_{i=M+1}^{D} \lambda_i$$

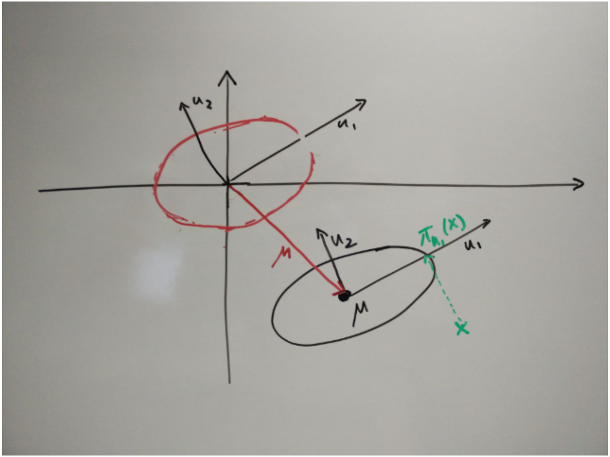
  i.e., the sum of the eigenvalues associated with eigenvectors not in the principle subspace
- Minimizing J requires us to choose the M eigenvectors as the principle subspace that are associated with the M largest eigenvalues.

Geometric Interpretation



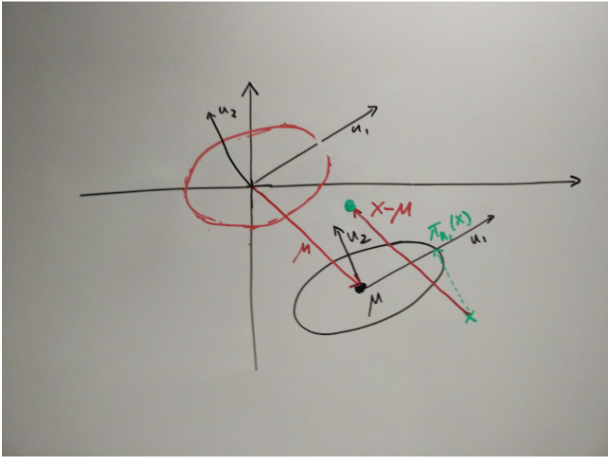
- Objective: Project $\boldsymbol{x}$ onto an affine subspace $\boldsymbol{\mu} + [\boldsymbol{u}_1]$.

Geometric Interpretation



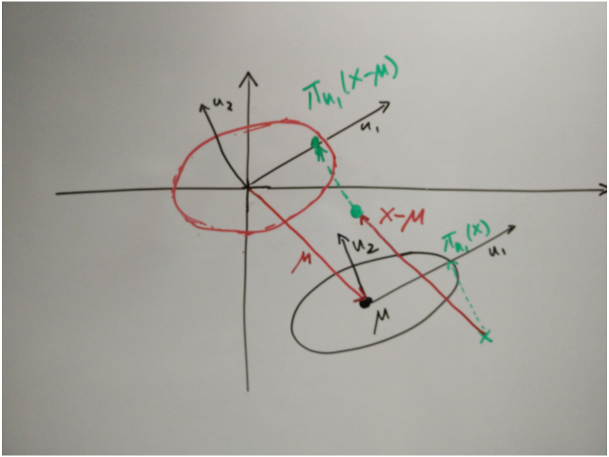
- Shift scenario to the origin (affine subspace $\rightsquigarrow$ subspace)

Geometric Interpretation



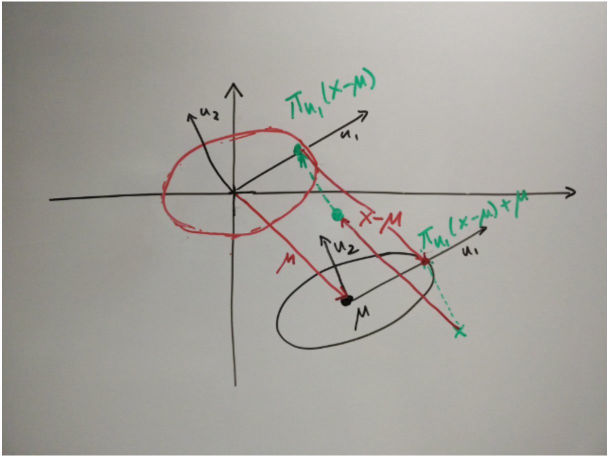
- Shift $\boldsymbol{x}$ as well (onto $\boldsymbol{x} - \boldsymbol{\mu}$).

Geometric Interpretation



- Orthogonal projection of $\boldsymbol{x} - \boldsymbol{\mu}$ onto subspace spanned by $\boldsymbol{u}_1$

Geometric Interpretation

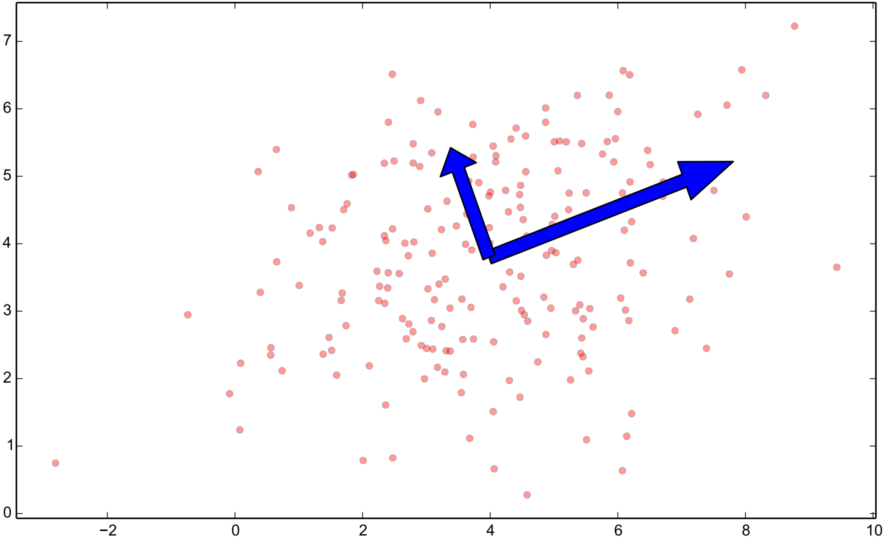


- Move projected point $\pi_{U_1}(\boldsymbol{x})$ back into original (affine) setting.

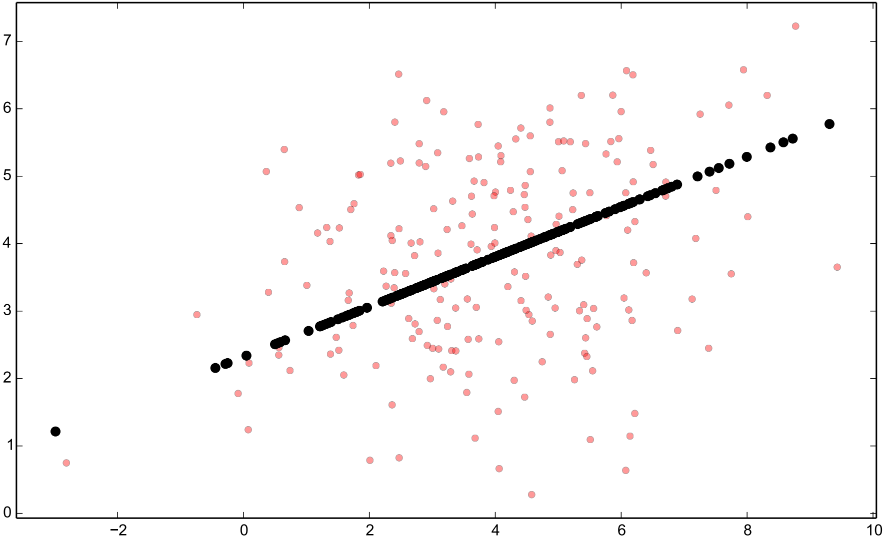
Algorithm

1. Compute the mean $\boldsymbol{\mu}$ of the data matrix $\boldsymbol{X} = [\boldsymbol{x}_1|....|\boldsymbol{x}_N]^\top \in \mathbb{R}^{N\times D}$
2. Mean normalization: Replace all data points $\boldsymbol{x}_i$ with $\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{\mu}$.
3. Compute the eigenvectors and eigenvalues of the data covariance matrix $\boldsymbol{S} = \frac{1}{N}\bar{\boldsymbol{X}}^\top\bar{\boldsymbol{X}}$
4. Choose the eigenvectors associated with the M largest eigenvalues to be the basis of the principal subspace.
5. Collect these eigenvectors in a matrix $\boldsymbol{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_M]$
6. Projected vector (in affine setting): $\boldsymbol{U}\boldsymbol{U}^\top(\boldsymbol{x} - \boldsymbol{\mu}) + \boldsymbol{\mu}$

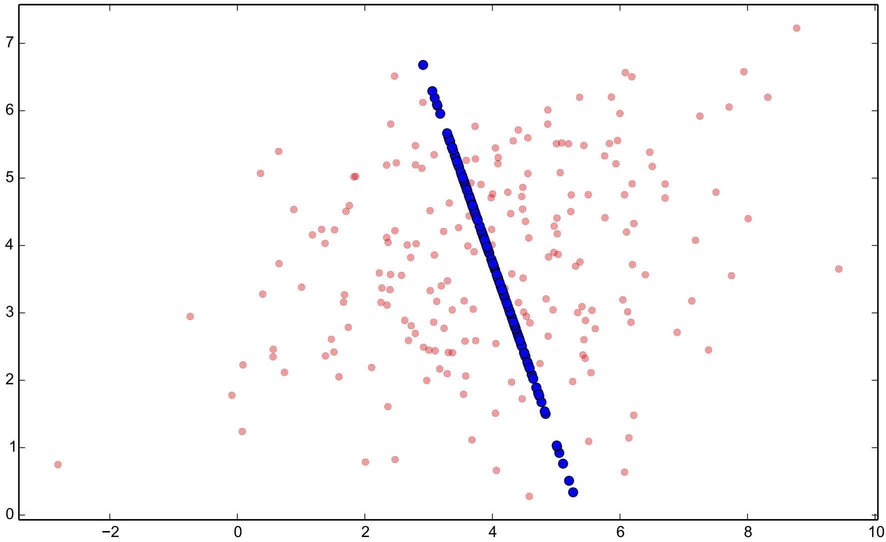
Example 1



Example 1



Example 1



Example 2

- Transform images into vectors
- Perform PCA ⏩ Compression/dimensionality reduction to extract low-dimensional features
- Use these features for face recognition

PCA for High-Dimensional Data

- Fewer data points than dimensions, i.e., $N < D$.
- At least $D - N + 1$ eigenvalues 0.
- Computation time for computing eigenvalues of $\boldsymbol{S}$: $\mathcal{O}(D^3)$
- Rephrase PCA

Reformulating PCA

- Define $\boldsymbol{X}$ to be the $N \times D$ dimensional centered data matrix, whose nth row is $(\boldsymbol{x}_n - \mathbb{E}[\boldsymbol{x}])^\top$ ⏩ Mean normalization

Reformulating PCA

- Define $\boldsymbol{X}$ to be the $N \times D$ dimensional centered data matrix, whose nth row is $(\boldsymbol{x}_n - \mathbb{E}[\boldsymbol{x}])^\top$ ⏩ Mean normalization
- Corresponding covariance: $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^\top\boldsymbol{X}$

Reformulating PCA

- Define $\boldsymbol{X}$ to be the $N \times D$ dimensional centered data matrix, whose nth row is $(\boldsymbol{x}_n - \mathbb{E}[\boldsymbol{x}])^\top$ ⏩ Mean normalization
- Corresponding covariance: $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^\top\boldsymbol{X}$
- Corresponding eigenvector equation:

$$\boldsymbol{S}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \Leftrightarrow \frac{1}{N}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$$

Reformulating PCA

- Define $\boldsymbol{X}$ to be the $N \times D$ dimensional centered data matrix, whose nth row is $(\boldsymbol{x}_n - \mathbb{E}[\boldsymbol{x}])^\top$ ▶▶ Mean normalization
- Corresponding covariance: $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^\top \boldsymbol{X}$
- Corresponding eigenvector equation:

$$\boldsymbol{S}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \Leftrightarrow \frac{1}{N}\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$$

- Transformation (left-multiply by $\boldsymbol{X}$):

$$\frac{1}{N}\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \quad \Leftrightarrow \quad \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top \underbrace{\boldsymbol{X}\boldsymbol{u}_i}_{=:\boldsymbol{v}_i} = \lambda_i \underbrace{\boldsymbol{X}\boldsymbol{u}_i}_{=:\boldsymbol{v}_i}$$

▶▶ $\boldsymbol{v}_i$ is an eigenvector of the $N \times N$-matrix $\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top$, which has the same eigenvalues as the original covariance matrix.

▶▶ Get eigenvalues in $\mathcal{O}(N^3)$ instead of $\mathcal{O}(D^3)$.

Recovering the Original Eigenvectors

- The new eigenvalue/eigenvector equation is

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$$

where $\boldsymbol{v}_i = \boldsymbol{X}\boldsymbol{u}_i$

Recovering the Original Eigenvectors

- The new eigenvalue/eigenvector equation is

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$$

  where $\boldsymbol{v}_i = \boldsymbol{X}\boldsymbol{u}_i$

- We want to recover the original eigenvectors $\boldsymbol{u}_i$ of the data covariance matrix $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^\top \boldsymbol{X}$

Recovering the Original Eigenvectors

- The new eigenvalue/eigenvector equation is

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$$

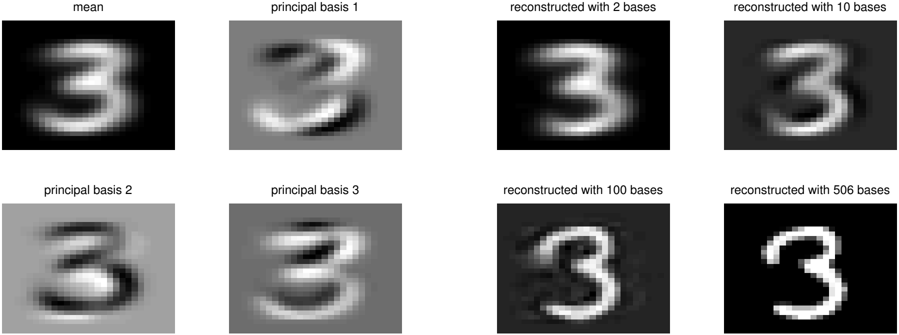
where $\boldsymbol{v}_i = \boldsymbol{X}\boldsymbol{u}_i$

- We want to recover the original eigenvectors $\boldsymbol{u}_i$ of the data covariance matrix $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^\top\boldsymbol{X}$
- Left-multiply eigenvector equation by $\boldsymbol{X}^\top$ yields

$$\underbrace{\frac{1}{N}\boldsymbol{X}^\top\boldsymbol{X}}_{=\boldsymbol{S}}\boldsymbol{X}^\top \boldsymbol{v}_i = \lambda_i \boldsymbol{X}^\top\boldsymbol{v}_i$$

and we recover $\boldsymbol{X}^\top\boldsymbol{v}_i$ as an eigenvector of $\boldsymbol{S}$ with eigenvalue λ_i

Example 3



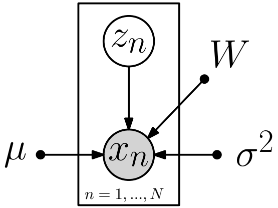
From "Machine Learning, A Probabilistic Perspective" (Murphy, 2012)

- 25 images of MNIST hand-written digits data set
- Left: Vectors of the eigenbasis
- Right: Reconstructions of the original digit

Interpretations of PCA

- Minimum reconstruction error (this course, Bishop, 12.1.2)
- Maximum variance of the data (Bishop, 12.1.1)
- Maximum mutual information between original and projected data
- Latent variable model where the latent variable is the low-dimensional representation of the data (probabilistic PCA, Bishop, 12.2)

Probabilistic PCA



$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}$$
$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$
$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$$

- Find parameters $\boldsymbol{W}, \boldsymbol{\mu}, \sigma^2$ via maximum likelihood
- Integrate out the latent variable $\boldsymbol{z}$, and obtain

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} = \mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{\mu}, \boldsymbol{C})$$
$$\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^\top + \sigma^2\boldsymbol{I}$$

- Posterior on low-dimensional latent variable:

$$p(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z} \,|\, \boldsymbol{M}^{-1}\boldsymbol{W}^\top(\boldsymbol{x} - \boldsymbol{\mu}), \sigma^2\boldsymbol{M}^{-1})$$
$$\boldsymbol{M} = \boldsymbol{W}^\top\boldsymbol{W} + \sigma^2\boldsymbol{I}$$

Properties

- Linear dimensionality reduction technique
- Original formulation: sensitive to scale of variables
- Global optimum (closed-form solution)
- Nonlinear extensions: Kernel PCA, ngeural network (deep) auto-encoders, Isomap, Laplacian Eigenmaps, ...

Applications

- Computer vision: Image compression, face recognition/identification (e.g., Turk & Pentland, 1991)
- Data visualization
- Neuroscience, oceanography, ...

References I

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag, 2006.

[2] M. P. Deisenroth and S. Mohamed. Expectation Propagation in Gaussian Process Dynamical Systems. In *Advances in Neural Information Processing Systems*, pages 2618–2626, 2012.

[3] H. Hotelling. Analysis of a Complex of Statistical Variables into Principal Components. *Journal of Educational Psychology*, 24:417–441, 1933.

[4] R. Linsker. Self-Organization in a Perceptual Network. *IEEE Computer*, 21(3):105–117, 1988.

[5] K. P. Murphy. *Machine Learning: A Proabilistic Perspective*. MIT Press, Cambridge, MA, USA, 2012.

References II

[6] K. Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 2(11):559–572, 1901.

[7] M. Turk and A. Pentland. Face Recognition Using Eigenfaces. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1991.