Principal component analysis

Danijel Skočaj



University of Ljubljana
Faculty of Computer and Information Science

Udine; 5, 12 December 2014

Principal Component Analysis (PCA)

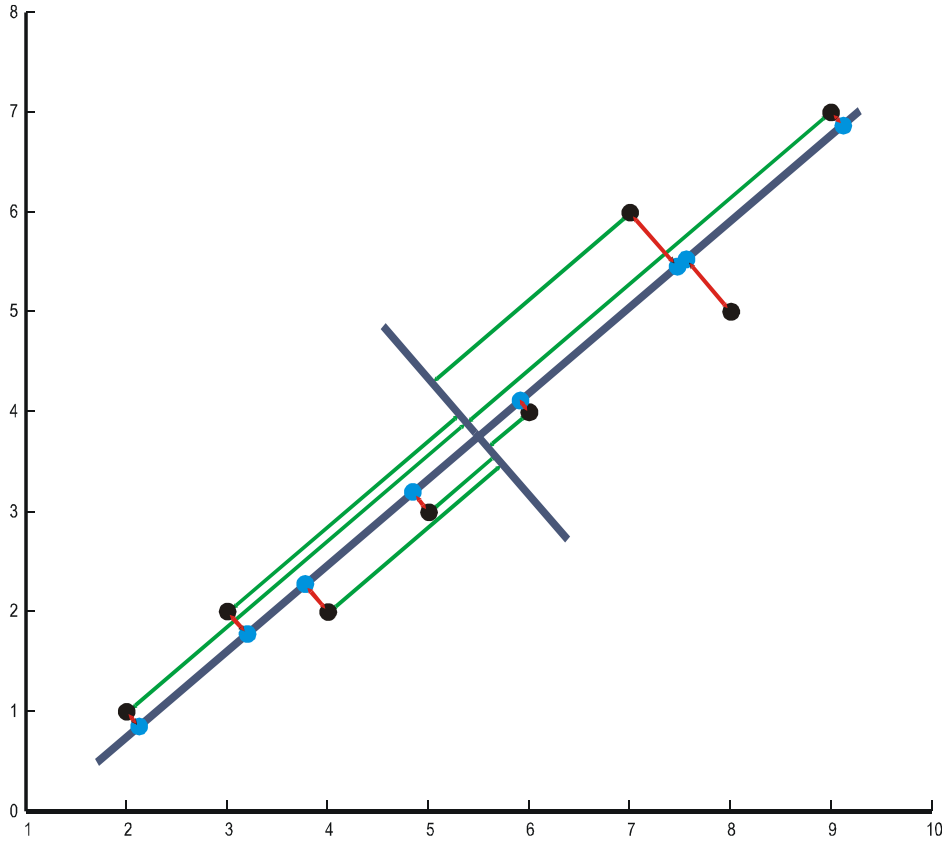
- PCA is a linear transformation from a high-dimensional input space to a low-dimensional feature space, which
  - maximizes variance of projected input vectors
  - minimizes reconstruction error
  - decorrelates input vectors.

- PCA finds in a data-driven way a more "natural" coordinate frame for representing given data.

Principle of PCA

Rotate coordinate frame in order to:

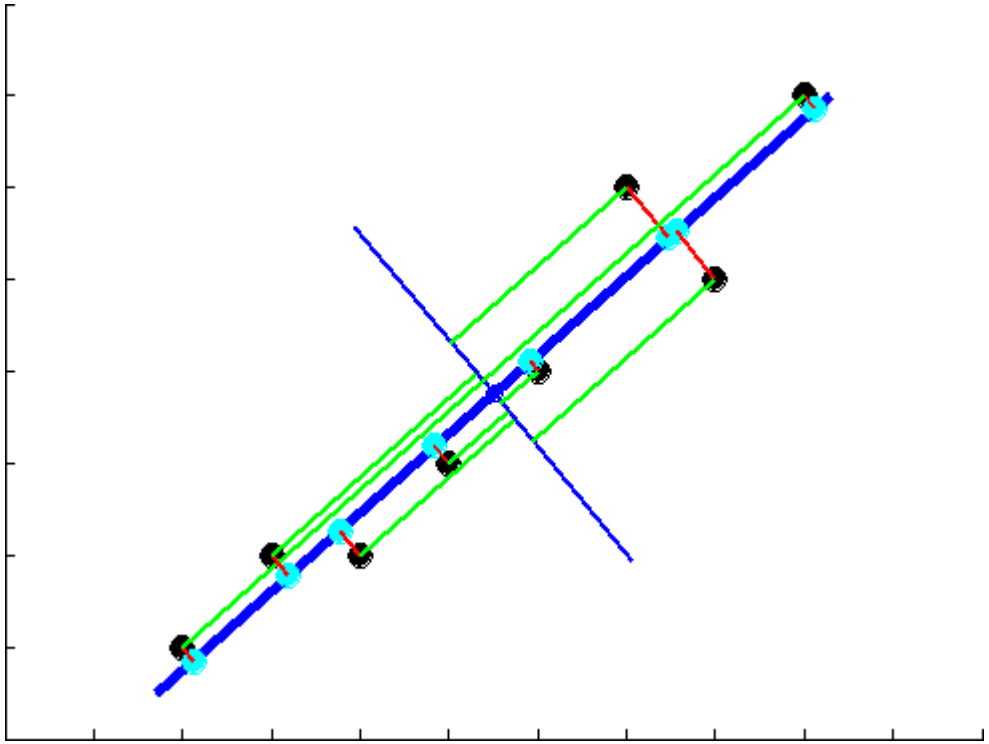
- Maximize variance of projections.
- Minimize reconstruction error.



Principle of PCA

Rotate coordinate frame in order to:

- Maximize variance of projections.
- Minimize reconstruction error.



Formulation of problem

- Let $\mathbf{X} \in \mathbb{R}^{M \times N}$ be a matrix of *N* vectors in *M*-dimensional input space.
- Let $\mathbf{u} \in \mathbb{R}^{M}$ be a direction (a vector of length 1) in the input space.
- The projection of the *j*–th vector $\mathbf{x}_j$ onto the vector $\mathbf{u}$ can be calculated as:

$$a_j = \langle \mathbf{x}_j, \mathbf{u} \rangle = \mathbf{u}^\top \mathbf{x}_j = \sum_{i=1}^{M} x_{ij} u_i$$

- The goal is to find a direction $\mathbf{u}$ that maximizes the variance of the projections of all input vectors $\mathbf{x}_j \ , \ j = 1 \ldots N$.

Variance

- The mean of the projections is

$$\bar{a} = \frac{1}{N}\sum_{j=1}^{N} a_j = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{M} x_{ij}u_i = \sum_{i=1}^{M} u_i\mu_i$$

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{x}_j$$

μ_i

- The variance is

$$\hat{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}\mathbf{1}_{1\times N}$$

$$\sigma^2 = \frac{1}{N}\sum_{j=1}^{N}(a_j - \bar{a})^2 = \frac{1}{N}\sum_{j=1}^{N}\left(\sum_{i=1}^{M} x_{ij}u_i - \sum_{i=1}^{M} u_i\mu_i\right)^2 =$$

$$= \frac{1}{N}\sum_{j=1}^{N}\left(\sum_{i=1}^{M} u_i\hat{x}_{ij}\right)^2 = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{k=1}^{M} u_i\hat{x}_{ij}u_k\hat{x}_{kj} =$$

$$= \sum_{i=1}^{M}\sum_{k=1}^{M} u_i u_k \frac{1}{N}\langle \hat{\mathbf{x}}_{i:}, \hat{\mathbf{x}}_{k:}\rangle = \sum_{i=1}^{M}\sum_{k=1}^{M} u_i c_{ik} u_k = \mathbf{u}^\top \mathbf{C}\mathbf{u}$$

where $\mathbf{C} = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^\top$ is the covariance matrix.

Maximizing variance

- Maximize σ^2 under the constraint that $\|\mathbf{u}\| = 1$
  ⇨ maximize the following function:

$$F(\mathbf{u};\lambda) = \mathbf{u}^\top \mathbf{C}\mathbf{u} - \lambda\left(\sum_{k=1}^{M} u_k^2 - 1\right) = \sum_{i=1}^{M}\sum_{j=1}^{M} u_i c_{ij} u_j - \lambda\left(\sum_{k=1}^{M} u_k^2 - 1\right)$$

- in the following way:

$$\frac{\partial F}{\partial u_l} = \sum_{j=1}^{M} c_{lj}u_j + \sum_{i=1}^{M} u_i c_{il} - \lambda 2 u_l = 0 \;;\; l = 1 \ldots M$$

$$\sum_{k=1}^{M} c_{lk}u_k = \lambda u_l \;;\; l = 1 \ldots M$$

$$\mathbf{C}\mathbf{u} = \lambda\mathbf{u}$$

⇨ compute eigenvectors and eigenvalues of $\mathbf{C}$:

- The largest eigenvalue = maximal variance.
- Corresp. eigenvector = direction with the max. var.

Eigenvalue decomposition

- Compute all eigenvector/eigenvalue pairs: $\mathbf{CU} = \mathbf{U\Lambda}$
- If $\mathbf{U}$ is invertible - **eigenvalue decomposition**:

$$\mathbf{C} = \mathbf{U\Lambda U}^{-1}$$

$\mathbf{C} \in \mathbb{R}^{M \times M} \Rightarrow$ $\mathbf{U} \in \mathbb{R}^{M \times M}$ *M* eigenvectors in columns

$\mathbf{\Lambda} \in \mathbb{R}^{M \times M}$ diagonal matrix with *M* eigenvalues on the diagonal

- Some properties of symmetric real matrices
  - Any two eigenvectors belonging to different eigenvalues are orthogonal.
  - All eigenvalues are real.
  - Every symmetric matrix can be diagonalized.
  - If a real symmetric matrix is of the form $\mathbf{XX}^{\top}$, all its eigenvales are nonnegative.

Singular value decomposition

- SVD decomposes a square matrix $\mathbf{C}$ to:

$$\mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top$$

  - orthonormal matrices $\mathbf{U}$ and $\mathbf{V}$
  - and diagonal matrix $\mathbf{\Lambda}$

$$\mathbf{C}^\top = \mathbf{C} \qquad \mathbf{U}^\top\mathbf{U} = \mathbf{I}$$

$$\mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$$

⇨ EVD and SVD of a real symmetric matrix are equivalent.

- $\mathbf{C} \in \mathbb{R}^{M\times M}$ ⇨ $\mathbf{U} \in \mathbb{R}^{M\times M}$ *M* eigenvectors in columns
  $\mathbf{\Lambda} \in \mathbb{R}^{M\times M}$ *M* eigenvalues on diagonal
- If $\mathbf{C}$ is of form $\mathbf{X}\mathbf{X}^\top$, where $\mathbf{X} \in \mathbb{R}^{M\times N}$ and $N < M$:

$$\mathbf{C} \in \mathbb{R}^{M\times M} \Rightarrow \begin{array}{ll} \mathbf{U} \in \mathbb{R}^{M\times N} & N \text{ eigenvectors in columns} \\ \mathbf{\Lambda} \in \mathbb{R}^{N\times N} & N \text{ eigenvalues on diagonal} \end{array}$$

Inner product matrix

- If M is large, SVD of covariance matrix $\mathbf{C}$ is unfeasible.
- If $N << M$, we can eigendecompose N x N inner product matrix $\mathbf{C}'$, rather than M x M covariance matrix $\mathbf{C}$:

$$\mathbf{C}' = \frac{1}{N}\hat{\mathbf{X}}^\top\hat{\mathbf{X}}$$

- The eigenvalues and the eigenvectors of $\mathbf{C}$ can then be determined from the eigenvalues λ'_i and the eigenvectors $\mathbf{u}'_i$ of $\mathbf{C}'$ as:

$$\begin{aligned}\lambda_i &= \lambda'_i \\ \mathbf{u}_i &= \frac{\hat{\mathbf{X}}\mathbf{u}'_\mathbf{i}}{\sqrt{N\lambda'_i}}\ , \quad i = 1 \ldots N\end{aligned}$$

Algorithm for PCA

Input: data matrix $\mathbf{X}$

Output: mean value $\boldsymbol{\mu}$, eigenvectors $\mathbf{U}$, eigenvalues $\boldsymbol{\lambda}$.

1. Estimate the mean vector: $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$.
2. Center the input data around the mean: $\hat{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}\mathbf{1}_{1\times N}$.
3. **if** $M \leq N$ **then**
4. Estimate the covariance matrix : $\mathbf{C} = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^{\top}$.
5. Perform SVD on $\mathbf{C}$. Obtain eigenvectors $\mathbf{U}$ and eigenvalues $\boldsymbol{\lambda}$.
6. **else**
7. Estimate the inner product matrix: $\mathbf{C}' = \frac{1}{N}\hat{\mathbf{X}}^{\top}\hat{\mathbf{X}}$.
8. Perform SVD on $\mathbf{C}'$. Obtain eigenvectors $\mathbf{U}'$ and eigenvalues $\boldsymbol{\lambda}'$.
9. Determine the eigenvectors $\mathbf{U}$: $\mathbf{u}_i = \frac{\hat{\mathbf{X}}\mathbf{u}_i'}{\sqrt{N\lambda_i'}}$, $i = 1 \ldots N$.
10. Determine the eigenvalues $\boldsymbol{\lambda} = \boldsymbol{\lambda}'$.
11. **end if**

Projection

- The orthonormal matrix $\mathbf{U}^\top$ serves as a linear transformation matrix from a high-dimensional input space to a low-dimensional feature space:

$$\mathbf{U}^\top : \mathbb{R}^M \rightarrow \mathbb{R}^N$$

$$\mathbf{a} = \mathbf{U}^\top \hat{\mathbf{x}}$$

$$a_j = \langle \hat{\mathbf{x}}, \mathbf{u}_j \rangle = \sum_{i=1}^{M} u_{ij} \hat{x}_i \quad , \quad j = 1 \ldots N$$

$$\mathbf{A} = \mathbf{U}^\top \hat{\mathbf{X}}$$

$\mathbf{A} \in \mathbb{R}^{N \times N}$ principal components (coefficients)

$\mathbf{U} \in \mathbb{R}^{M \times N}$ principal vectors (eigenvectors)

$\hat{\mathbf{X}} \in \mathbb{R}^{M \times M}$ mean centered input vectors

Decorrelation

- The mean of the projections is

$$\boldsymbol{\mu}_{\mathbf{A}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{a}_j = \frac{1}{N}\sum_{j=1}^{N}\mathbf{U}^\top\hat{\mathbf{x}} = \mathbf{U}^\top\underbrace{\frac{1}{N}\sum_{j=1}^{N}\hat{\mathbf{x}}}_{=\,0} = \mathbf{0}$$

- The correlation matrix of $\mathbf{A}$ is

$$\mathbf{C}_{\mathbf{A}} = \frac{1}{N}\mathbf{A}\mathbf{A}^\top = \frac{1}{N}\mathbf{U}^\top\hat{\mathbf{X}}(\mathbf{U}^\top\hat{\mathbf{X}})^\top =$$
$$= \mathbf{U}^\top\frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\mathbf{U} = \mathbf{U}^\top\mathbf{C}\mathbf{U} =$$
$$= \mathbf{U}^\top\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top\mathbf{U} = \boldsymbol{\Lambda}$$

$$\mathbf{C} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$$

⇨ The covariance matrix of the transformed data is a diagonal matrix ⇨ transformed vectors are **decorrelated**.

Variance

⇨ The covariance matrix of the transfomed data equals $\mathbf{\Lambda}$

⇨ The variance in the direction of the *i* -th principal axis is equal to the *i* -th eigenvalue:

$$\frac{1}{N}\sum_{j=1}^{N} a_{ij}^2 = \lambda_i$$

- The diagonalization preserves the trace of the matrix.

⇨ The total variance of the input data has been preserved and equals to the sum of the eigenvalues.

$$\mathrm{VAR}(\mathbf{X}) = \sum_{i=1}^{M} \frac{1}{N}\hat{\mathbf{x}}_{i:}\hat{\mathbf{x}}_{i:}^{\top} = \sum_{i=1}^{M} c_{ii} =$$

$$= \sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} \frac{1}{N}\mathbf{a}_{i:}\mathbf{a}_{i:}^{\top} = \mathrm{VAR}(\mathbf{A})$$

Reconstruction

- The orthonormal matrix $\mathbf{U}$ serves as a linear transformation matrix from a low-dimensional feature space to a high-dimensional feature space:

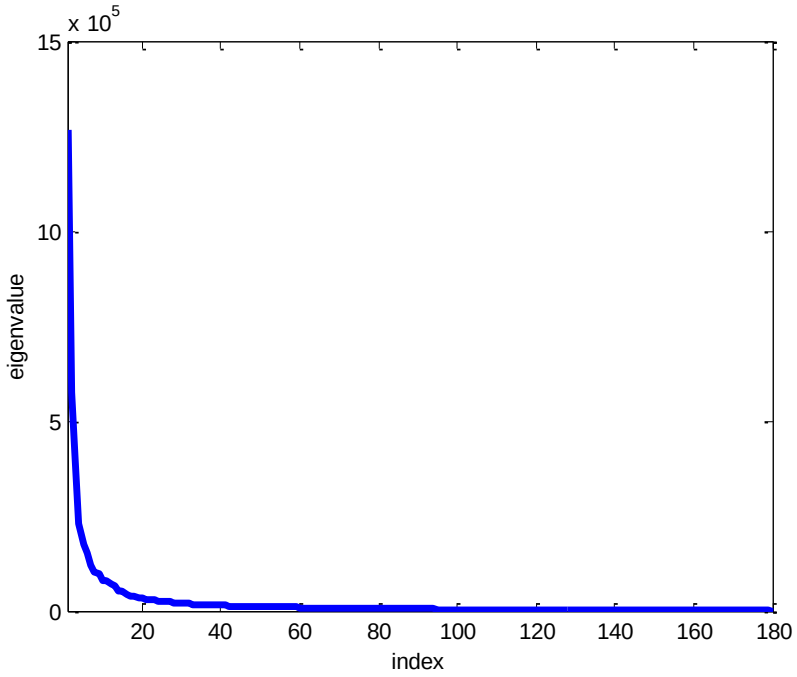
$$\mathbf{U} : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

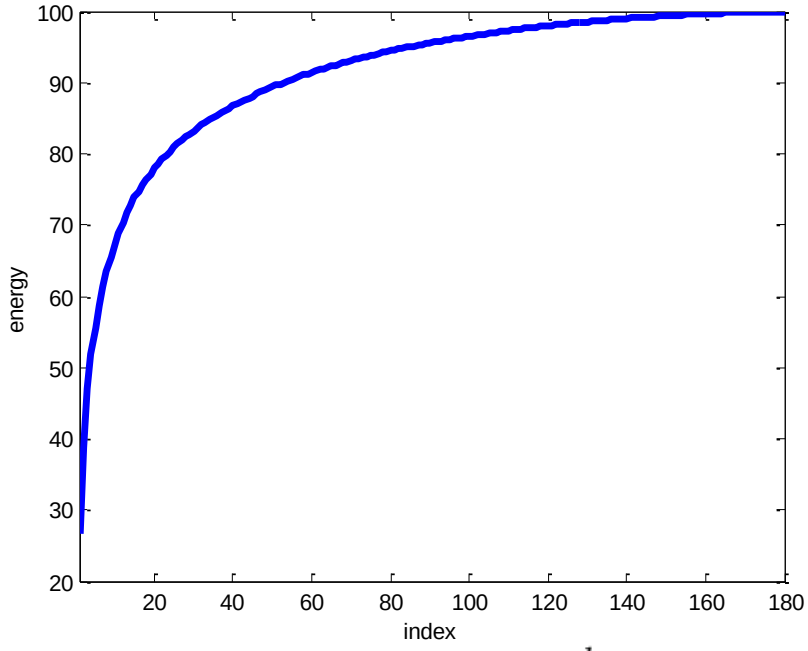
$$\hat{\mathbf{y}} = \mathbf{U}\mathbf{a} = \sum_{j=1}^{N} a_j \mathbf{u}_j$$

- If all *N* principal vectors are used, thus $\mathbf{U} \in \mathbb{R}^{M \times N}$, each input image in $\mathbf{X} \in \mathbb{R}^{M \times N}$ can be perfectly reconstructed without any reconstruction error.

Eigenspectrum and energy

- Most of the variance is contained across the first few principal vectors.





eigenspectrum

energy: $en_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i}$

Projection and reconstruction

- A subset of principal components suffices for a good approximation of the input data.

⇨ Use only *k*, *k*<<*N* principal axes: $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{M \times k}$

- Projection: $\mathbf{U}^\top : \mathbb{R}^M \rightarrow \mathbb{R}^k$

$$\mathbf{a} = \mathbf{U}^\top \hat{\mathbf{x}} = \mathbf{U}^\top (\mathbf{x} - \boldsymbol{\mu})$$

$$a_j = \langle \hat{\mathbf{x}}, \mathbf{u}_j \rangle = \sum_{i=1}^{M} u_{ij} \hat{x}_i = \sum_{i=1}^{M} u_{ij} (x_i - \mu_i) \quad , \quad j = 1 \ldots k$$

- Reconstruction: $\mathbf{U} : \mathbb{R}^k \rightarrow \mathbb{R}^M$

$$\hat{\mathbf{y}} = \mathbf{U}\mathbf{a} = \sum_{j=1}^{k} a_j \mathbf{u}_j$$

$$\mathbf{y} = \hat{\mathbf{y}} + \boldsymbol{\mu} \ .$$

Reconstruction error

- Reconstruction error (residual vector):

$$\mathbf{e} = \hat{\mathbf{x}} - \hat{\mathbf{y}} = \sum_{i=1}^{N} a_i \mathbf{u}_i - \sum_{i=1}^{k} a_i \mathbf{u}_i = \sum_{i=k+1}^{N} a_i \mathbf{u}_i$$

- Squared reconstruction error:

$$e = \|\mathbf{e}\|^2 = \left\| \sum_{i=k+1}^{N} a_i \mathbf{u}_i \right\|^2 \overset{\mathbf{U}^\top \mathbf{U} = \mathbf{I}}{=} \sum_{i=k+1}^{N} a_i^2$$

  = sum of the discarded principal components.

⇨ The expected squared reconstruction error can be approximated with expected values of the variances across discarded eigenvectors:

$$\mathcal{E}(e) = \sum_{i=k+1}^{N} \lambda_i$$

Minimizing reconstruction error

⇨ Maximizing variance in non-discarded eigenvectors
=
minimizing the squared reconstruction error

- PCA finds principal vectors $\mathbf{u}_i, \ i = 1 \ldots k$ and vectors of principal components $\mathbf{a}_i \in \mathbb{R}^k, i = 1 \ldots N$, which minimize the squared reconstruction error

$$e = \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\hat{x}_{ij} - \sum_{l=1}^{k} u_{il} a_{lj} \right)^2$$

- A nonlinear minimization problem
  - Gradient descend algorithm
  - Neural networks
  - EM algorithm

Probabilistic approach

- PCA can be considered as a limiting case of a linear gaussian model, when the noise is infinitesimally small and equal in all directions.
- Expectation-Maximization algorithm [Roweis 97]:
  - E-step: $\mathbf{A} = (\mathbf{U}^\top \mathbf{U})^{-1} \mathbf{U}^\top \hat{\mathbf{X}}$
  - M-step: $\mathbf{U} = \hat{\mathbf{X}} \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1}$
- Stopping criterion:

$$\sigma^2 = \frac{1}{N-k} \sum_{j=k+1}^{N} \lambda_j = \frac{1}{N-k} \left(\mathrm{VAR}(\hat{\mathbf{X}}) - \sum_{j=1}^{k} \lambda_j \right) =$$

$$= \frac{1}{N-k} \left(\mathrm{VAR}(\hat{\mathbf{X}}) - \mathrm{VAR}(\mathbf{A}) \right)$$

- At convergence the columns of $\mathbf{U}$ span the principal subspace.

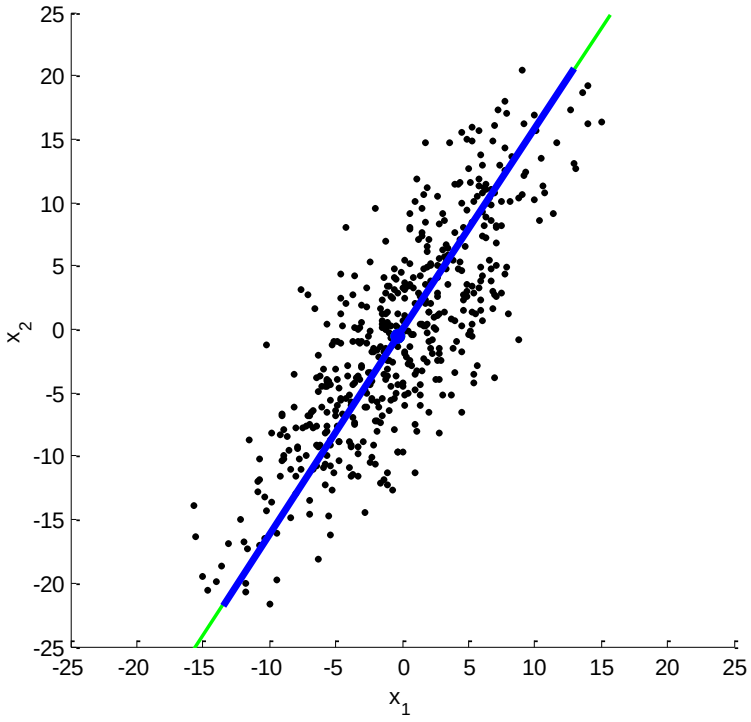
EM algorithm for PCA

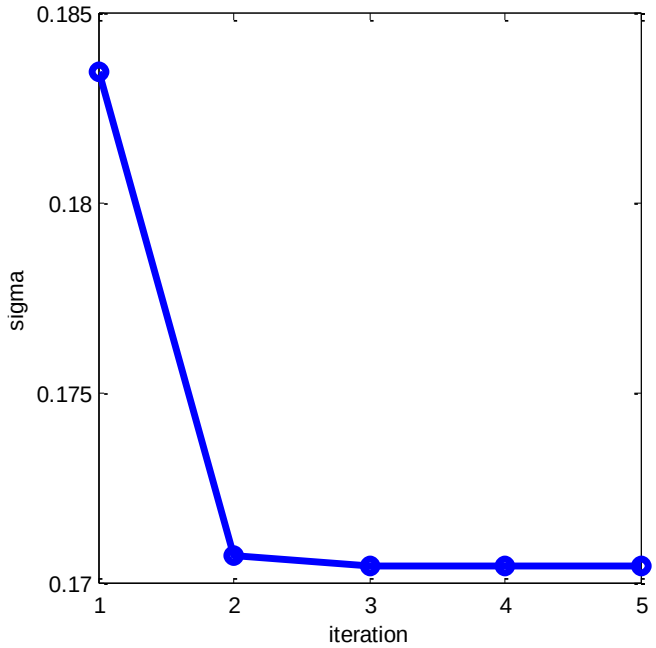
Input: data matrix $\mathbf{X}$, number of principal axes to be estimated k.

Output: mean value $\boldsymbol{\mu}$, eigenvectors $\mathbf{U}$, eigenvalues $\boldsymbol{\lambda}$.

1. Estimate the mean vector: $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$.
2. Center the input data around the mean: $\hat{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}\mathbf{1}_{1\times N}$.
3. Set elements of $\mathbf{U} \in \mathbb{R}^{M\times k}$ to random values.
4. **repeat**
5. E-step: $\mathbf{A} = (\mathbf{U}^\top\mathbf{U})^{-1}\mathbf{U}^\top\hat{\mathbf{X}}$
6. M-step: $\mathbf{U} = \hat{\mathbf{X}}\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}$
7. **until** change in $\sigma^2 = \frac{1}{N-k}\left(\mathrm{VAR}(\hat{\mathbf{X}}) - \mathrm{VAR}(\mathbf{A})\right) \leq thr$
8. Orthogonalize $\mathbf{U}$.
9. Project input data on $\mathbf{U}$: $\mathbf{A} = \mathbf{U}^\top\hat{\mathbf{X}}$
10. Perform PCA on $\mathbf{A}$. Obtain $\mathbf{U}'$ and $\boldsymbol{\lambda}'$.
11. Rotate $\mathbf{U}$ for $\mathbf{U}'$: $\mathbf{U} = \mathbf{U}\mathbf{U}'$.
12. Determine the eigenvalues $\boldsymbol{\lambda} = \boldsymbol{\lambda}'$.

Convergence of EM algorithm





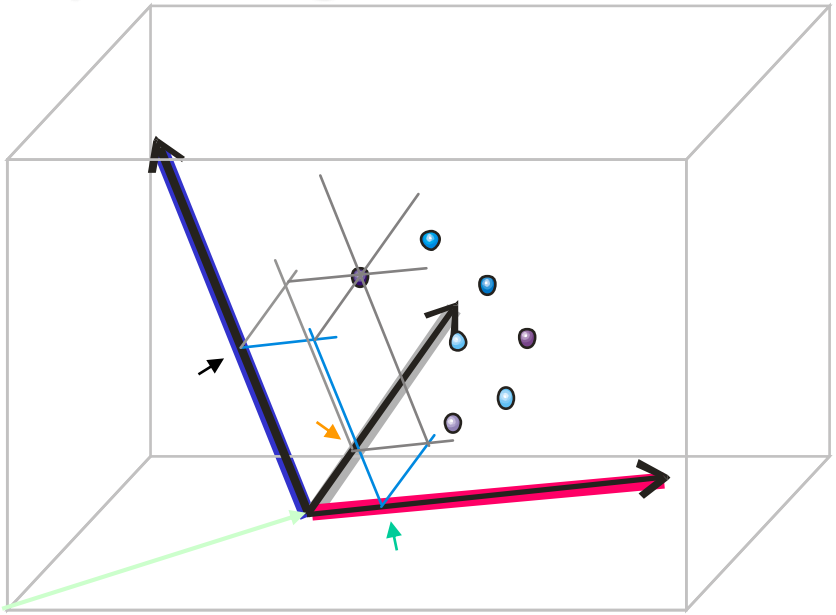
Subspace representation

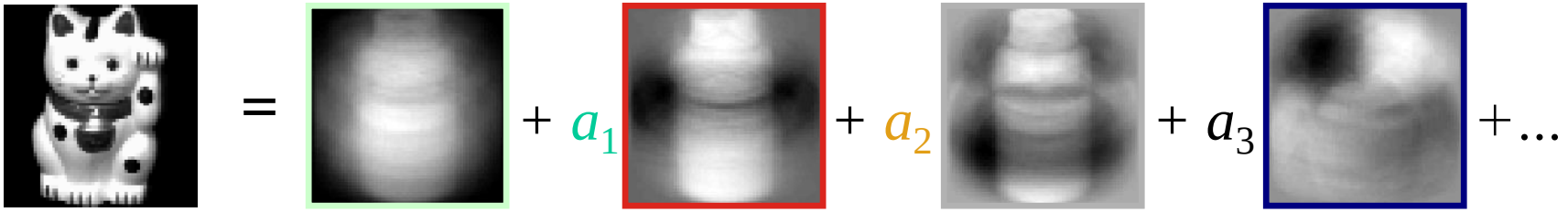
- An image is represented by a linear combination of principal images.
- Projection:

$$\mathbf{a} = \mathbf{U}^\top(\mathbf{x} - \boldsymbol{\mu})$$

- Reconstruction:

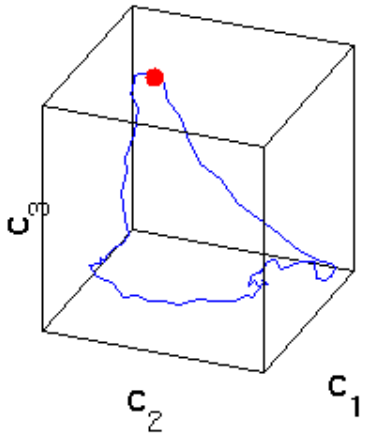
$$\mathbf{x} \approx \mathbf{U}\mathbf{a} + \boldsymbol{\mu}$$





Object recognition and pose estimation





new image:



Recognition:



An object is represented as
a manifold in the principal subspace.

Pose estimation:

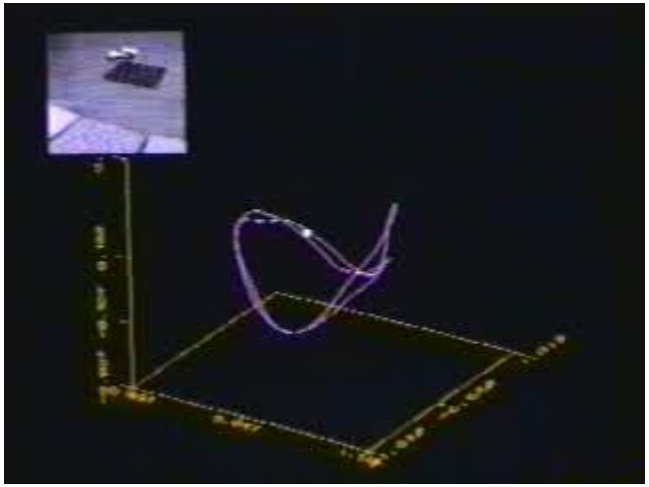


PCA-based representation

- PCA-based representation is efficient because:
  - is easy to build (learn)
    - EVD or SVD
  - does not require a huge amount of memory
    - it is optimal in the sense of minimizing reconstruction error
  - enables generalization on unseen views
    - using interpolation between projections of training images
  - enables fast recognition (matching)
    - searching for the closest point in a low-dimensional space.

Example

- Object recognition and pose estimation

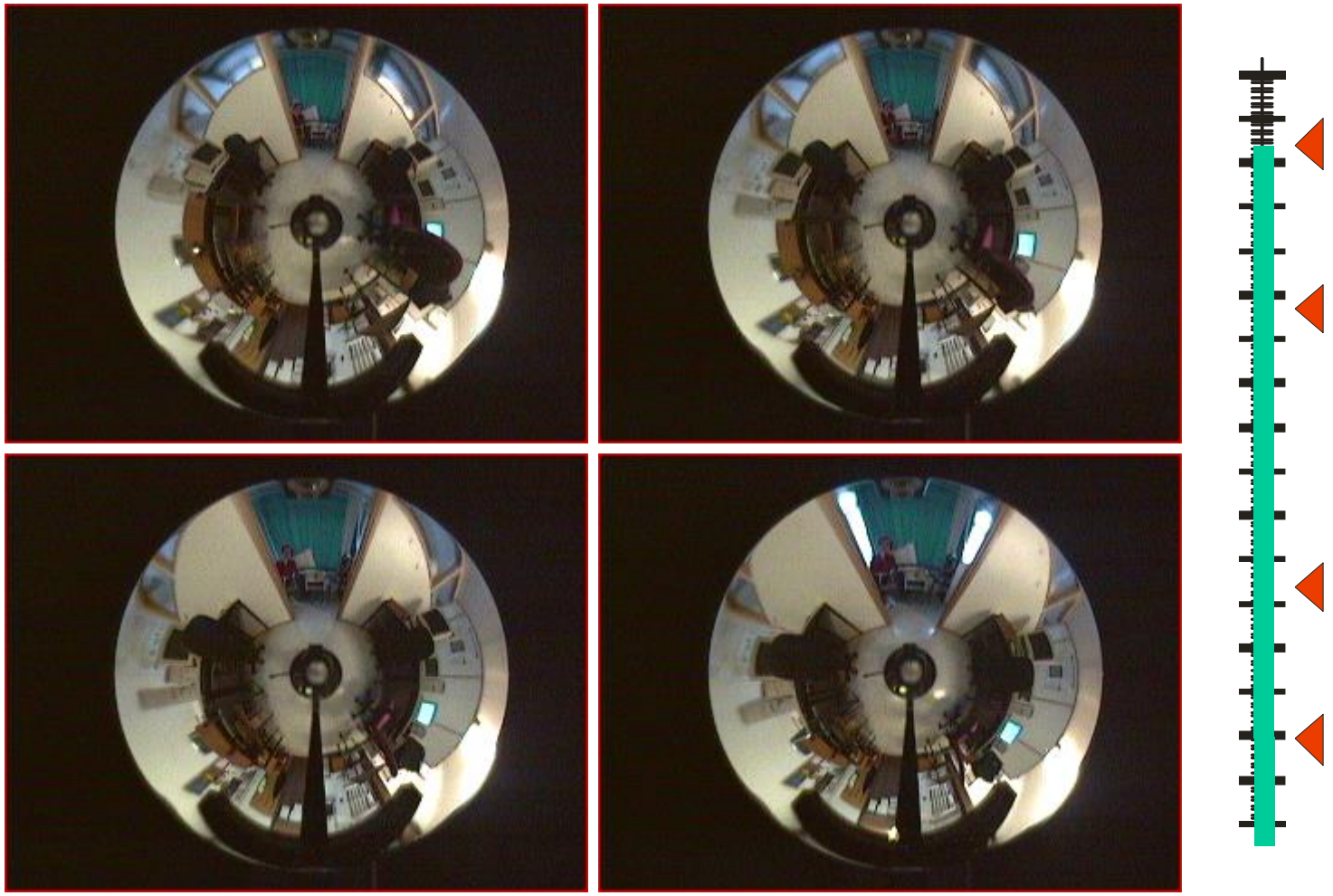




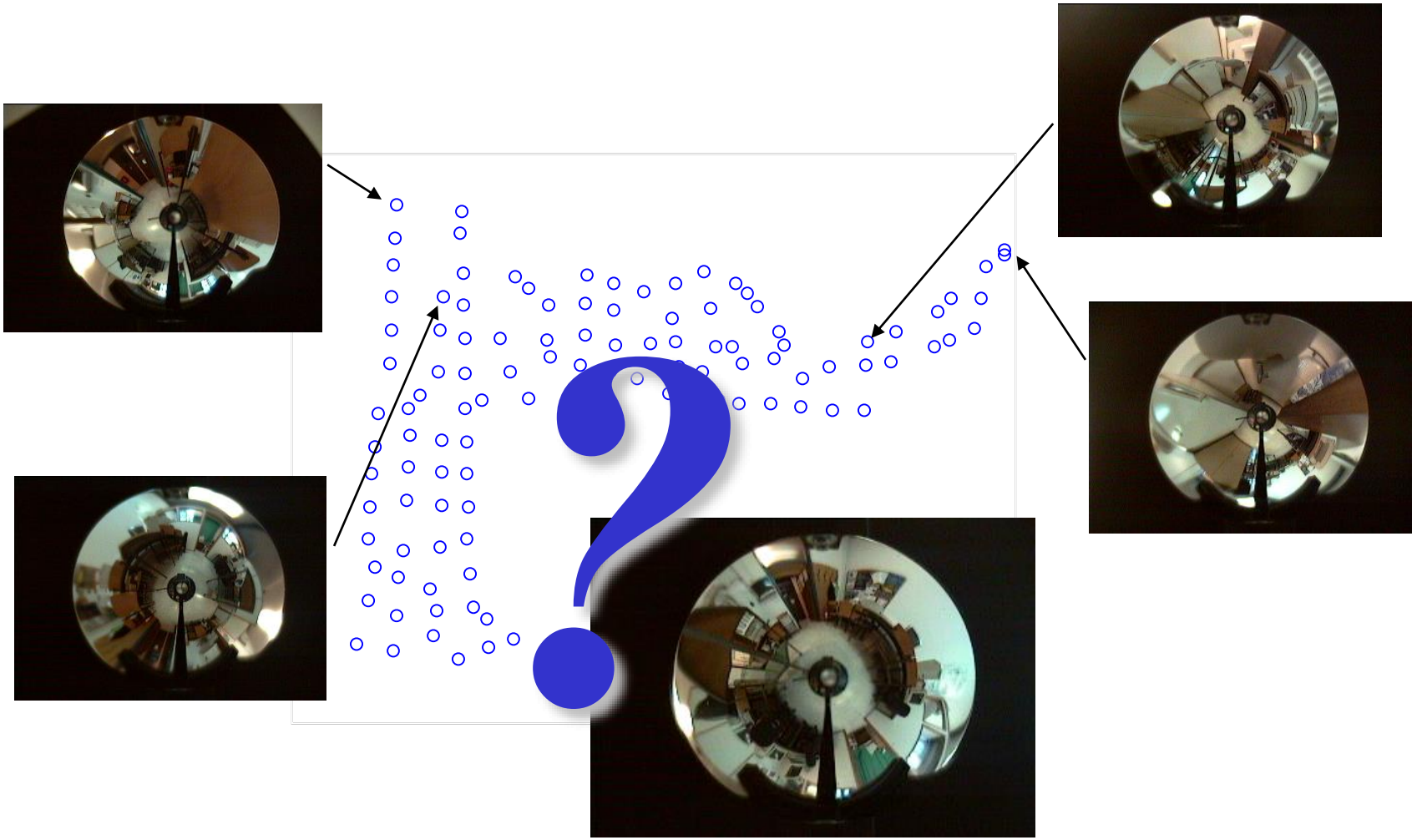
Localisation of a mobile robot



Localisation of a mobile robot



Localisation of a mobile robot



Literature

- Danijel Skočaj*, Robust subspace approaches to visual learning and recognition*, PhD thesis, University of Ljubljana, 2003
  - http://vicos.fri.uni-lj.si/data/publications/skocajPhD03.pdf

- C. M. Bishop, *Pattern recognition and machine learning*, Springer, 2006