

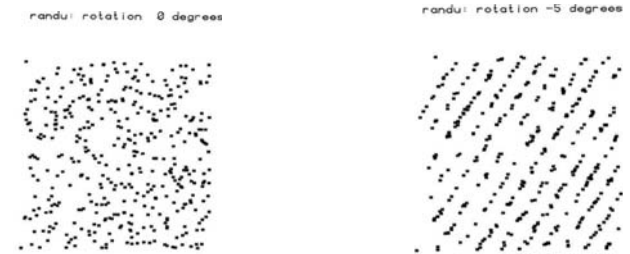
## Machine Learning: Topic Chart

- Core problems of pattern recognition
- Bayesian decision theory
- Perceptrons and Support vector machines
- Data clustering
- *Dimension reduction*

## What is Dimensionality Reduction ?

**Goal:** Automatically find “interesting” projections from high-dimensional feature space to low-dimensional space.

**Example** for structure in high-dimensional spaces: IBM random number generator RANDU(early FORTRAN lib.), triplets  $(x_{n+2}, x_{n+1}, x_n)$  lie on 15 parallel planes.



## Reasons for Dimensionality Reduction

select *most interesting* dimensions in preprocessing step:

- data compression
- feature selection
- complexity reduction
- Example: face recognition,  $m \times n$  grey-scale image lives in  $mn$ -dimensional space.

**visualization of data:** project to 1, 2 or 3 dimensional space.

## When does Dimensionality Reduction work?

**“Noise dimensions”:** many variables may have very small variation, and may hence be ignored

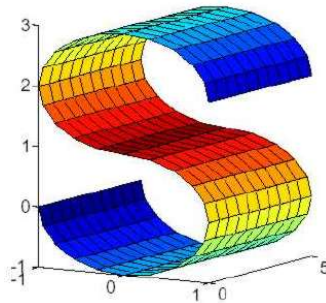
**Decoupling:** many variables may be correlated / dependent, hence a new set of independent variables is preferable.

**Problem:** projection is “smoothing”: (high-dim.) structure is obscured, but never enhanced.

**Goal:** find sharpest / most interesting projections

## linear vs. non-linear projections

example:



What is the result of linear vs. non-linear projections?

## Overview

### Linear Projections:

- Principal Component Analysis (PCA)
- Exploratory Projection Pursuit

### Non-Linear Projections:

- locally linear embedding (LLE)
- **more methods in “Machine Learning II”**

## Linear Projection

from high-dim. space  $\mathbb{R}^d$  to low-dim. space  $\mathbb{R}^m$ :

$$z = Wx$$

where

$$x \in \mathbb{R}^d$$

$$z \in \mathbb{R}^m$$

$W$  is a linear map (matrix):

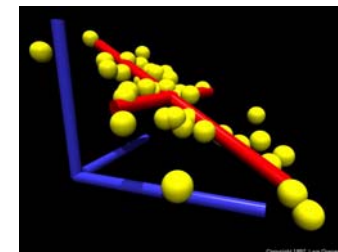
- orthogonal projection: row vectors of  $W$  are orthonormal
- if  $m = 1$ :  $W$  reduces to a row vector  $w^\top$

**Note:** while the projection is linear, the objective function (see below) may be non-linear!

## Principal Component Analysis (PCA)

Idea:

- Shift the coordinate system in the center of mass of the given data points
- and rotate it to align coordinate axes with principal axes
- to capture as much *interesting signal* as possible: maximum variance of data.



## PCA: formal setup

**Given** are data points  $x^s \in \mathbb{R}^d$ ,  $s = 1, \dots, n$ .

**New Rotated Coordinate System:** Define a new set of  $d$  orthonormal basis vectors  $\phi_i \in \mathbb{R}^d$ , i.e.,

$$\phi_i^\top \phi_j = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{otherwise} \end{cases}$$

**data point in new coordinate system:**  $x^s = \sum_{i=1}^d y_i^s \phi_i$

**Approximation of data points**  $x^s$  : use only  $m \leq d$  coordinates to optimally approximate  $x^s$ . Replace coordinates  $m < i \leq d$  by preselected, optimized constants  $b_i$ :

$$\hat{x}^s(m) = \sum_{i \leq m} y_i^s \phi_i + \sum_{m < i \leq d} b_i \phi_i$$

**Note:** the  $b_i$  do not depend on index  $s$ , i.e., cannot be adapted to the individual data points  $x^s$  ( $\rightarrow$  shift to center of mass).

**Approximation Error for data point**  $x^s$ :

$$\begin{aligned} \Delta x^s &= x^s - \hat{x}^s(m) = x^s - \sum_{i \leq m} y_i^s \phi_i - \sum_{m < i \leq d} b_i \phi_i \\ &= \sum_{m < i \leq d} (y_i^s - b_i) \phi_i \end{aligned}$$

## A Quality-Measure of the Projection: Mean Squared Error

$$\mathbb{E}\{\|\Delta x^s(m)\|^2\} = \sum_{m < i \leq d} \mathbb{E}\{(y_i^s - b_i)^2\}$$

**“Interestingness” criterion in PCA:** Choose the representation with minimal  $\mathbb{E}\{\|\Delta x^s(m)\|^2\}$ , i.e., optimize the  $b_i, \phi_i$  to minimize  $\mathbb{E}\{\|\Delta x^s(m)\|^2\}$ .

**Remark:** An equivalent criterion is to maximize mutual information between original data points and their projections (assumption: Gaussian distribution of data).

**Necessary condition for minimum:**

$$\begin{aligned} \frac{\partial}{\partial b_i} \mathbb{E}\{(y_i^s - b_i)^2\} &= -2(\mathbb{E}\{y_i^s\} - b_i) = 0 \\ \Rightarrow b_i &= \mathbb{E}\{y_i^s\} = \phi_i^\top \mathbb{E}\{x^s\} \end{aligned}$$

## Inserting into the error criterion:

$$\begin{aligned} \mathbb{E}\{\|\Delta x^s\|^2\} &= \sum_{m < i \leq d} \mathbb{E}\{(y_i^s - \mathbb{E}\{y_i^s\})^2\} \\ &= \sum_{m < i \leq d} \phi_i^\top \underbrace{\mathbb{E}\{(x^s - \mathbb{E}\{x^s\})(x^s - \mathbb{E}\{x^s\})^\top\}}_{=: \Sigma_X} \phi_i \end{aligned}$$

**Optimal Choice of Basis Vectors:** Choose the eigenvectors of the covariance matrix  $\Sigma_X$ , i.e.,

$$\Sigma_X \phi_i = \lambda_i \phi_i$$

**Costs of PCA:**

$$\mathbb{E}\{\|\Delta x^{s,\text{opt}}(m)\|^2\} = \sum_{m < i \leq d} \lambda_i$$

**Proof Idea:** Choose an arbitrary orthonormal basis

$$\psi_i = \sum_j a_{ij} \phi_j, \text{ i.e., } a_i^\top a_k = \delta_{ik}.$$

$$\Rightarrow \mathbf{E}\{\|\Delta X(m)\|^2\} = \sum_{i=m+1}^d a_i^\top \Lambda a_i$$

where  $\Lambda$  ... diagonal matrix with  $\lambda_i$  on diagonal.

Minimize this functional under the constraint that the vectors  $a_i$  are orthonormal, and use the fact that, for  $i > m$ ,  $\delta_i$  are the smallest eigenvalues.

$$\Rightarrow a_i = (0, \dots, 0, \underbrace{1}_i, 0, \dots, 0) \text{ is a solution,}$$

but any rotation in the subspace of the  $d - m$  eigenvectors with the smallest  $d - m$  eigenvalues also minimizes the criterion.

$\Rightarrow$  The eigenvectors  $\phi_i$  minimize the error criterion.

## PCA: Summary

compute sample mean  $\mathbf{E}\{x^s\}$  and covariance matrix  $\Sigma_X = \mathbf{E}\{(x^s - \mathbf{E}\{x^s\})(x^s - \mathbf{E}\{x^s\})^\top\}$

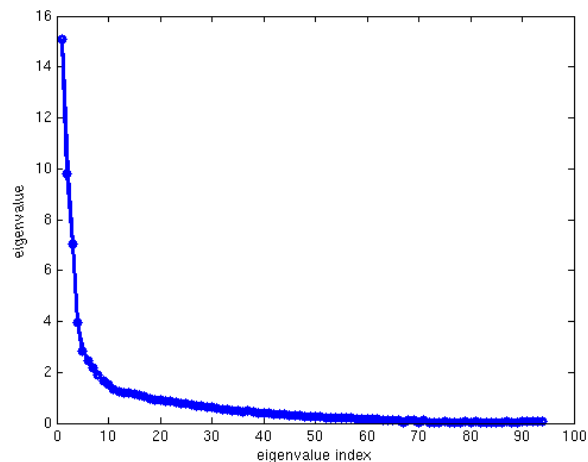
compute spectral decomposition  $\Sigma_X = \Phi \Lambda \Phi^\top$

transformed data points:  $y^s = \Phi^\top (x^s - \mathbf{E}\{x^s\})$

projection: for each  $y^s$ , retain only those components  $i$  where  $\lambda_i$  is among the largest  $m$  eigenvalues.



Figure 4: Standard Eigenfaces.



## Factor Analysis

**Data:**  $n$  data vectors  $X = (X_1, \dots, X_d)$ ;  $n \times d$  data matrix  $\mathbf{X}$

**Singular Value Decomposition:**  $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$  with orthogonal matrices  $\mathbf{U}$ ,  $\mathbf{V}$  and singular values in the diagonal matrix  $\mathbf{D}$ .

**Latent Variables:** Let  $\mathbf{S} = \sqrt{n}\mathbf{U}$  and  $\mathbf{A}^\top = \mathbf{D}\mathbf{V}^\top/\sqrt{n}$

**Interpret**  $X = \mathbf{A}\mathbf{S}$  as a latent variable model.

Problem: The decomposition of  $X$  is not unique since  $X = \mathbf{A}\mathbf{S} = \mathbf{A}\mathbf{R}^\top\mathbf{R}\mathbf{S} =: \mathbf{A}^*\mathbf{S}^*$  for any orthogonal matrix  $\mathbf{R}$ .

**Factor Analysis:** Assume  $X = \mathbf{A}\mathbf{S} + \epsilon$ ;

$\mathbf{S}$  is a vector of  $q < d$  underlying latent variables.

**Goal:** Determine components enforcing additional constraints.

## Independent Component Analysis

**Find components** which are statistically independent.

**Measure of Dependence: Mutual Information**

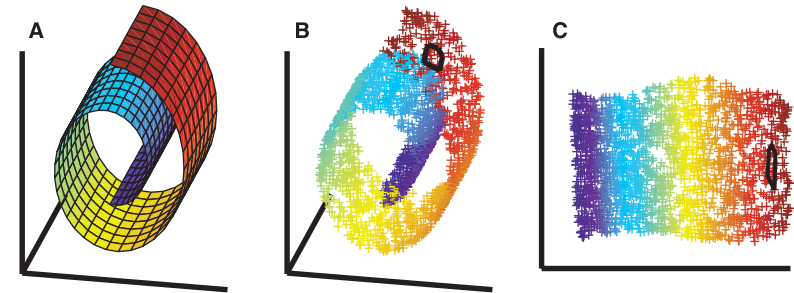
$$\mathcal{I}(Y) = \sum_{j \leq d} H(Y_j) - H(Y).$$

**Strategy:** find a decomposition  $X = \mathbf{A}S$  which minimizes  $\mathcal{I}(Y) = \mathcal{I}(\mathbf{A}^T X)$

**Procedure:** perform a factor analysis and rotate the components to make them mutually independent.

## Non-Linear Projection Methods

**example:** unfolding the locally linear, but globally highly non-linear structure:



What is the result of a linear projection?

## Locally Linear Embedding (LLE)

Saul & Roweis: *Nonlinear Dimensionality Reduction by Locally Linear Embedding*, Science 290, 2323(2000)

**non-linear** projection method

**Basic Idea:** use local patches

- each data point is related to a small number  $k$  of its neighbors
- relation within a patch is modeled in a linear way
- $k$  is the only free parameter

## LLE Algorithm

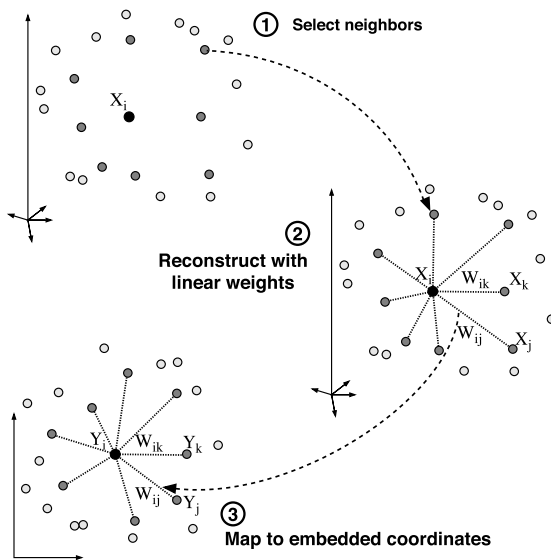
- 1) compute neighbors of each data point  $x_s, s = 1, \dots, n$ .
- 2) approximate each data point  $x_s \in \mathbb{R}^p$  by  $\hat{x}^s = \sum_t W_{st} x_t$ , where the  $x_t$ 's are the neighbors of  $x_s$  (**linear approximation**): find weights  $W_{st}$  that minimize

$$\text{cost}(W) = \sum_s \|x_s - \hat{x}^s\|^2 = \sum_s \|x_s - \sum_t W_{st} x_t\|^2$$

- 3) project to low-dimensional space: **assume that weights  $W_{st}$  capture local geometry also in low-dim. space**. Given the weights  $W_{st}$  from 2), find projected points  $y^s$  by minimizing

$$\text{cost}(y) = \sum_s \|y_s - \sum_t W_{st} y_t\|^2 \quad y_s \in \mathbb{R}^d, d \ll p$$

**Fig. 2.** Steps of locally linear embedding: (1) Assign neighbors to each data point  $X_i$  (for example by using the  $K$  nearest neighbors). (2) Compute the weights  $W_{ij}$  that best linearly reconstruct  $X_i$  from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors  $Y_i$  best reconstructed by  $W_{ij}$  minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights  $W_{ij}$  and vectors  $Y_i$  are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.



## Remarks on LLE

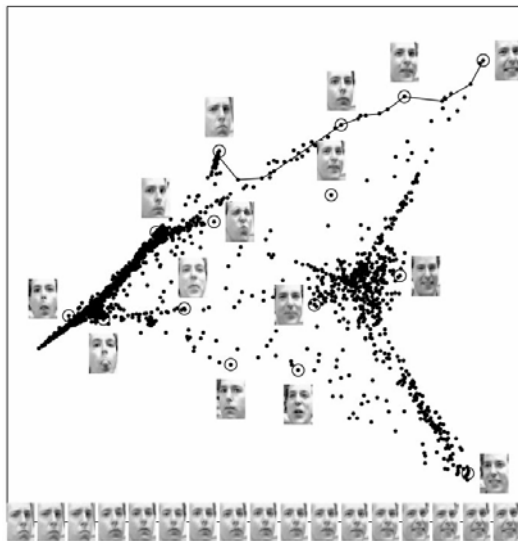
### constraints on weights:

- $W_{st} = 0$  unless  $x_s$  and  $x_t$  are neighbors.
- normalization: for all  $s$ :  $\sum_t W_{st} = 1$ .

**Reason for constraints:** this ensures invariance to rotation, rescaling, translation of data points.

### Optimization:

- step 2): solve least squares problem
- step 3): solve  $n \times n$  eigenvector problem
- no local minima
- computational complexity is quadratic in  $n$



**Fig. 3.** Images of faces (17) mapped into the embedding space described by the first two coordinates of LLE. Representative faces are shown next to circled points in different parts of the space. The bottom images correspond to points along the top-right path (linked by solid line), illustrating one particular mode of variability in pose and expression.