ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Introduction to Machine Learning

Unsupervised Learning: Dimension Reduction

Prof. Andreas Krause
Learning and Adaptive Systems (las.ethz.ch)

We will

- Introduce basic dimension reduction algorithms
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Neural network autoencoders
- Much more details in
  - Computational Intelligence Lab
  - Deep Learning

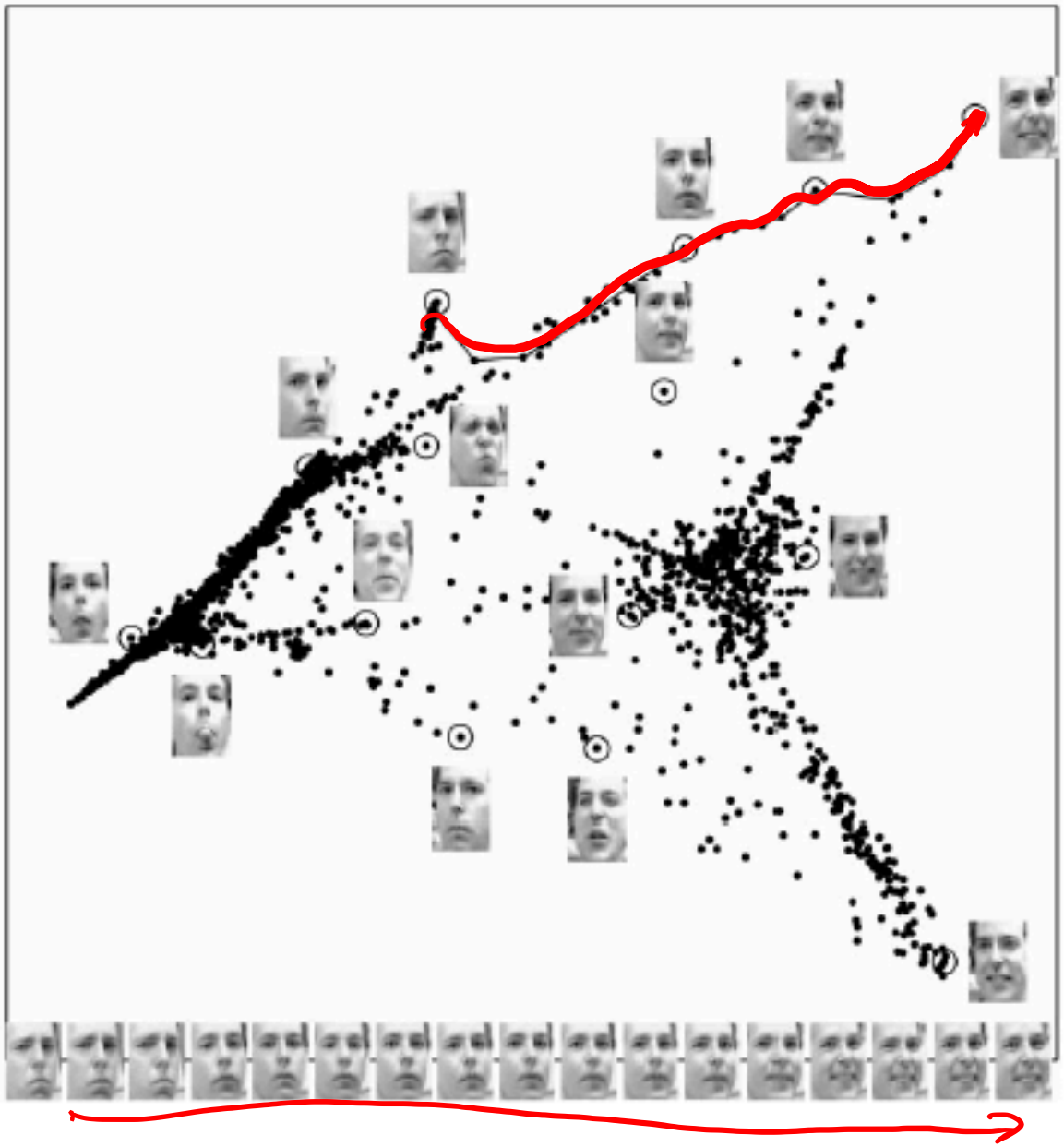
Basic challenge

- Given data set $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ $x_i \in \mathbb{R}^d$
  obtain „embedding“ (low-dimensional representation)
  $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^k$ $k < d$

- **Motivation**
  - Visualization (k=1,2,3)
  - Regularization (model selection)
  - Unsupervised feature discovery
    (i.e., determine features from data!)
  - ...

Example: Embedding of faces

[Saul & Roweis]

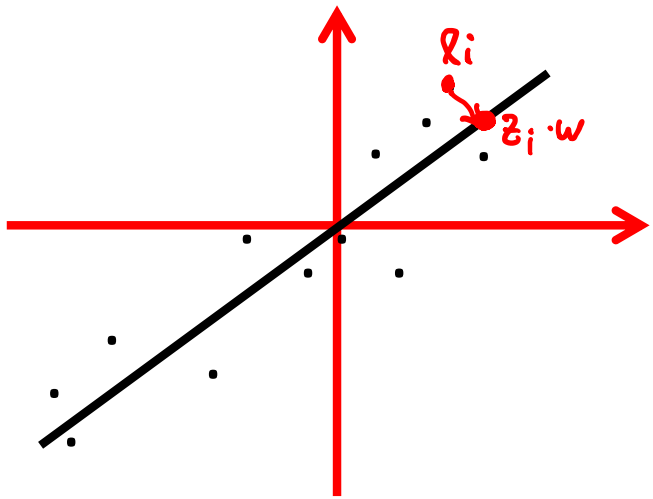


Typical approaches

- Assume $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$
- Obtain mapping $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^k$ where $k \ll d$
- Can distinguish
  - Linear dimension reduction: $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ $A \in \mathbb{R}^{k \times d}$
  - Nonlinear dimension reduction (parametric or non-parametric)
- **Key question**: Which mappings should we prefer?

Linear dim. reduction as *compression*

- **Motivation**: Low-dimensional representation should allow to *compress* original data (accurate reconstruction)
- **Example**: k=1
- Given data set $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$
- Want to represent data as points on a *line* $\mathbf{w} \in \mathbb{R}^d$ with coefficients $z_1, \ldots, z_n$
- I.e., want $z_i \mathbf{w} \approx \mathbf{x}_i$, assuming $\mu = \frac{1}{n} \sum_i \mathbf{x}_i = 0$



Linear dim. reduction for reconstruction

- Given data set $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$
- Want $z_i \mathbf{w} \approx \mathbf{x}_i$, e.g., minimizing $||z_i \mathbf{w} - \mathbf{x}_i||_2^2$
- To ensure uniqueness, normalize: $||\mathbf{w}||_2 = 1$

$(\alpha z_i)\left(\frac{1}{\alpha} w\right) = z_i w$

- Optimize over $\mathbf{w}, z_1, \ldots, z_n$ jointly:

$$(w^*, z_{1:n}^*) = \underset{w, z_{1:n}}{\operatorname{argmin}} \sum_{i=1}^{n} \| x_i - z_i \cdot w \|_2^2$$

Linear dim. reduction for reconstruction

- Given data set $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$
- Want $z_i \mathbf{w} \approx \mathbf{x}_i$, e.g., minimizing $||z_i \mathbf{w} - \mathbf{x}_i||_2^2$
- To ensure uniqueness, normalize: $||\mathbf{w}||_2 = 1$

- Optimize over $\mathbf{w}, z_1, \ldots, z_n$ jointly:

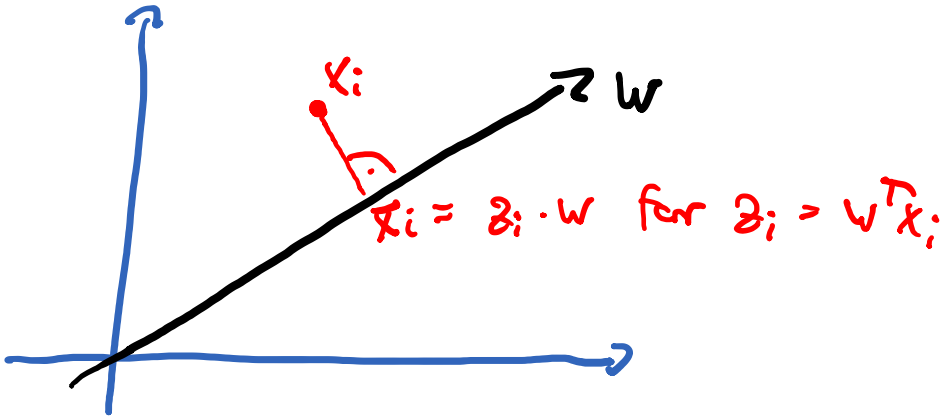
$$(\mathbf{w}^*, \mathbf{z}^*) = \arg \min_{||\mathbf{w}||_2=1, \mathbf{z}} \sum_{i=1}^{n} ||z_i \mathbf{w} - \mathbf{x}_i||_2^2$$

$\hat{R}(w, z_{1:n})$

Solving for *z* given ***w***

$$(\mathbf{w}^*, \mathbf{z}^*) = \arg \min_{||\mathbf{w}||_2=1, \mathbf{z}} \sum_{i=1}^{n} ||z_i \mathbf{w} - \mathbf{x}_i||_2^2$$

- Suppose we consider some vector ***w***. What can we say about the optimal *z*?



Solving for *z* given ***w***

$$(\mathbf{w}^*, \mathbf{z}^*) = \arg \min_{||\mathbf{w}||_2=1, \mathbf{z}} \sum_{i=1}^{n} ||z_i \mathbf{w} - \mathbf{x}_i||_2^2$$

- Suppose we consider some vector ***w***. What can we say about the optimal *z*?

$$z_i^* = \mathbf{w}^T \mathbf{x}_i$$

- Thus, we effectively solve a *regression* problem, interpreting ***x*** as features and *z* as labels!

Solving for *z* given ***w***

- Want to solve

$$(\mathbf{w}^*, \mathbf{z}^*) = \arg \min_{||\mathbf{w}||_2=1, \mathbf{z}} \sum_{i=1}^{n} ||z_i \mathbf{w} - \mathbf{x}_i||_2^2$$

- **Note**: For any fixed $||\mathbf{w}||_2 = 1$ it holds that $z_i^* = \mathbf{w}^T \mathbf{x}_i$ therefore, only need

$$\mathbf{w}^* = \arg \min_{||\mathbf{w}||_2=1} \sum_{i=1}^{n} ||\mathbf{w}\mathbf{w}^T \mathbf{x}_i - \mathbf{x}_i||_2^2$$

$\hat{R}(w)$

$\hat{R}(w) = \sum_{i=1}^{n} (ww^T x_i - x_i)^T (ww^T x_i - x_i) = \sum_{i=1}^{n} \left(x_i^T w \underbrace{w^T w}_{1} w^T x_i - 2 x_i w w^T x_i + x_i^T x_i \right)$

$-(x_i^T w)(w^T x_i) = -(w^T x_i)^2$ $\quad$ $\|x_i\|_2^2$

$= \underbrace{\sum_{i=1}^{n} \|x_i\|_2^2}_{\text{Constant wrt } w} - \sum_{i=1}^{n} (w^T x_i)^2$

$\Rightarrow \operatorname{argmin}_w \hat{R}(w) = \operatorname{argmax}_w \sum_{i=1}^{n} (w^T x_i)^2$

Solving for *w*

- The objective

$$\mathbf{w}^* = \arg \min_{||\mathbf{w}||_2=1} \sum_{i=1}^{n} ||\mathbf{w}\mathbf{w}^T\mathbf{x}_i - \mathbf{x}_i||_2^2$$

is equivalent to $$\mathbf{w}^* = \arg \max_{||\mathbf{w}||_2=1} \sum_{i=1}^{n} (\mathbf{w}^T\mathbf{x}_i)^2$$

(*)

$(*) = \sum_{i=1}^{n} w^T x_i x_i^T w = w^T \left(\sum_{i=1}^{n} x_i x_i^T \right) w$

$= n \cdot \Sigma$

↑ empirical covariance of data (assuming mean 0)

Solving for $\boldsymbol{w}$

- Further:

$$\mathbf{w}^* = \arg \max_{||\mathbf{w}||_2=1} \sum_{i=1}^{n} (\mathbf{w}^T \mathbf{x}_i)^2$$

is equivalent to: $\mathbf{w}^* = \arg \max_{||\mathbf{w}||_2=1} \mathbf{w}^T \Sigma \mathbf{w}$

where $\Sigma = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T$ is the *empirical covariance*

assuming the data is centered: $\mu = \frac{1}{n} \sum_i \mathbf{x}_i = 0$

Solving for $\boldsymbol{w}$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T$$

- The optimal solution to

$$\mathbf{w}^* = \arg \max_{||\mathbf{w}||_2=1} \mathbf{w}^T \Sigma \mathbf{w} \qquad \mu = \frac{1}{n}\sum_i \mathbf{x}_i = 0$$

(#) (##)

is given by the principal eigenvector of Σ

i.e., $\mathbf{w}^* = \mathbf{v}_1$ where

$$\Sigma = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$$

(###)

$$\lambda_1 \geq \cdots \geq \lambda_d \geq 0$$

Why? $w = \sum_{i=1}^{d} \alpha_i v_i$ Thus, objective (#) $= \left(\sum_i \alpha_i v_i\right)^T \left(\sum_{j=1}^{d} \lambda_j v_j v_j^T\right)\left(\sum_k \alpha_k v_k\right)$

$= \sum_{i,j,k} \alpha_i \alpha_k \lambda_j (v_i^T v_j)(v_j^T v_k) = \sum_{i=1}^{d} \alpha_i^2 \lambda_i$

$(v_i^T v_j) = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otw} \end{cases}$ $(v_j^T v_k) = \begin{cases} 1 & \text{if } j=k \\ 0 & \text{otw} \end{cases}$

Constraint (##) $= \sum_{i=1}^{d} \alpha_i^2 = 1$

Thus: from (###): $\alpha_1 = 1$, $\alpha_i = 0$ $\forall i > 1$

How about k>1?

- Suppose we wish to project to more than one dimension. Thus we want:

$$(\mathbf{W}, \mathbf{z}_1, \ldots, \mathbf{z}_n) = \arg\min \sum_{i=1}^{n} ||\mathbf{W}\mathbf{z}_i - \mathbf{x}_i||_2^2$$

where $\mathbf{W} \in \mathbb{R}^{d \times k}$ is *orthogonal*, $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^k$

$$W = \left(v_1 \middle| \cdots \middle| v_k \right) \in \mathbb{R}^{d \times k}$$

- This is called the Principal Component Analysis problem
- Its solution can be obtained in closed form even for k>1

Principal component analysis (PCA)

- Given centered data $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$, $1 \leq k \leq d$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \qquad \mu = \frac{1}{n}\sum_{i} \mathbf{x}_i = 0$$

- The solution to the PCA problem

$$(\mathbf{W}, \mathbf{z}_1, \ldots, \mathbf{z}_n) = \arg\min \sum_{i=1}^{n} ||\mathbf{W}\mathbf{z}_i - \mathbf{x}_i||_2^2$$

where $\mathbf{W} \in \mathbb{R}^{d \times k}$ is orthogonal, $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^k$

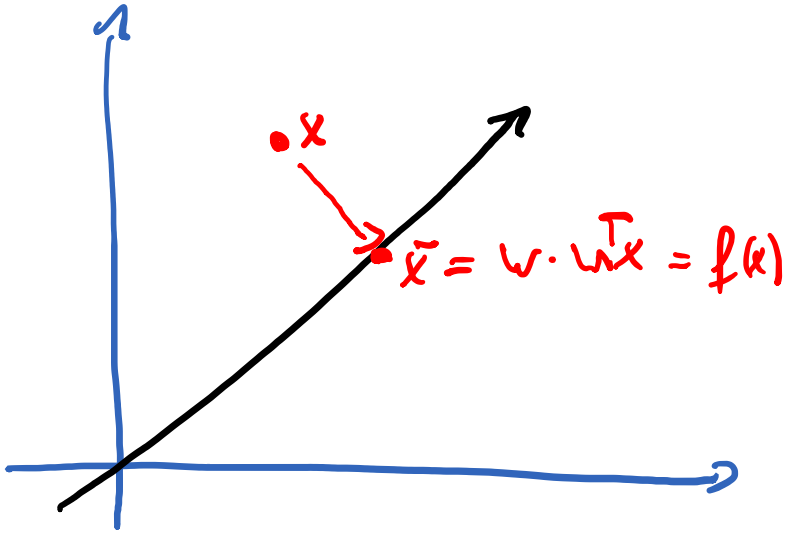
is given by $\mathbf{W} = (\mathbf{v}_1 | \ldots | \mathbf{v}_k)$ and $\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i$

where

$$\Sigma = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^T \qquad \lambda_1 \geq \cdots \geq \lambda_d \geq 0$$

PCA is a projection

- The linear mapping $\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$ obtained from PCA projects vectors $\mathbf{x} \in \mathbb{R}^d$ into a *k*-dimensional subspace

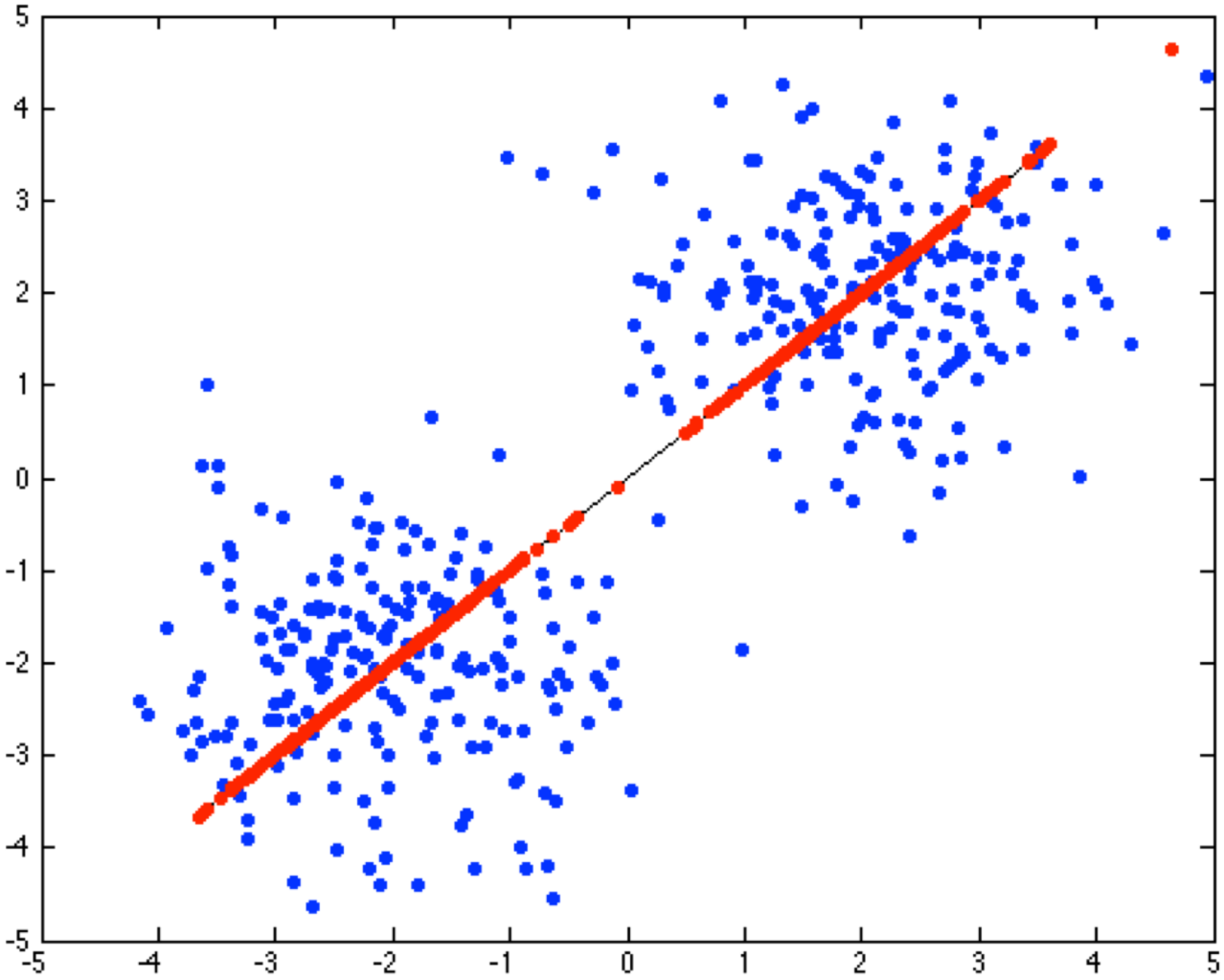


It holds that $f(f(x)) = f(x)$

$\Rightarrow f$ is a projection

- This projection is chosen to minimize the reconstruction error (measured in Euclidean norm)

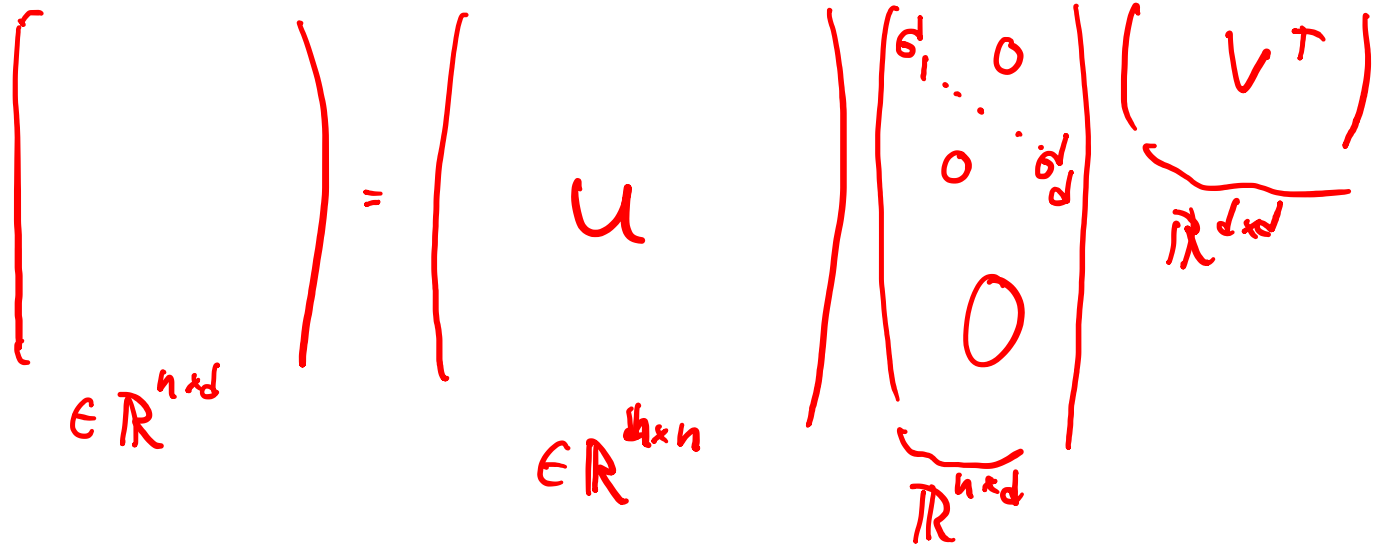
PCA Illustration



PCA Demo

Connection to SVD

- Can obtain PCA through Singular-Value Decomposition
- *Recall*: Can represent any $\mathbf{X} \in \mathbb{R}^{n \times d}$ as $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ where $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{d \times d}$ are orthogonal, and $\mathbf{S} \in \mathbb{R}^{n \times d}$ is diagonal (wlog in decreasing order). Its entries are called singular values



PCA via SVD

- *Recall*: Can represent any $\mathbf{X} \in \mathbb{R}^{n \times d}$ as $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ where $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{d \times d}$ are orthogonal, and $\mathbf{S} \in \mathbb{R}^{n \times d}$ is diagonal (wlog in decreasing order).
- The top *k* principal components are exactly the first *k* columns of ***V***

$$n\Sigma = X^T X = V S^T \underbrace{U^T U}_{I} S V^T = V \underbrace{S^T S}_{D} V^T$$

Common PCA Usecases

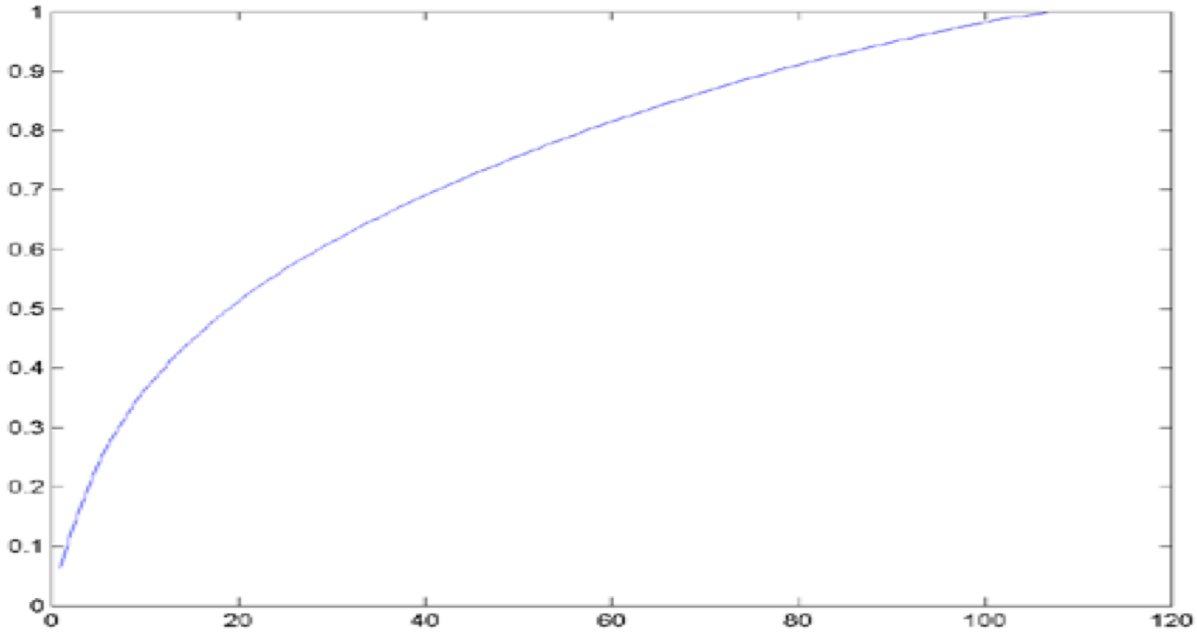
- Visualization (k=1, 2, 3)
- Feature learning
- Compression

Example: Eigenfaces [de Coro]

Choosing *k*

- For visualization: by inspection ☺
- For feature induction: by cross-validation
- Otherwise: Pick *k* so that most of the variance is explained (similar to the choice in *k*-means)

Reconstruction performance



Now we will

- Discuss relationship of PCA to other methods
- Introduce nonlinear dimension reduction techniques

Side note: PCA vs. K-Means

PCA Problem:

$$(\mathbf{W}, \mathbf{z}_1, \ldots, \mathbf{z}_n) = \arg\min \sum_{i=1}^{n} ||\mathbf{W}\mathbf{z}_i - \mathbf{x}_i||_2^2$$

where $\mathbf{W} \in \mathbb{R}^{d \times k}$ is orthogonal, $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^k$

k-Means problem: (equivalent formulation)

$$(\mathbf{W}, \mathbf{z}_1, \ldots, \mathbf{z}_n) = \arg\min \sum_{i=1}^{n} ||\mathbf{W}\mathbf{z}_i - \mathbf{x}_i||_2^2$$

where $\mathbf{W} \in \mathbb{R}^{d \times k}$ *arbitrary* and $\mathbf{z}_1, \ldots, \mathbf{z}_n \in E_k$

hereby, $E_k = \{[1, 0, \ldots, 0], \ldots, [0, \ldots, 0, 1]\}$
is the set of unit vectors in $\mathbb{R}^k$

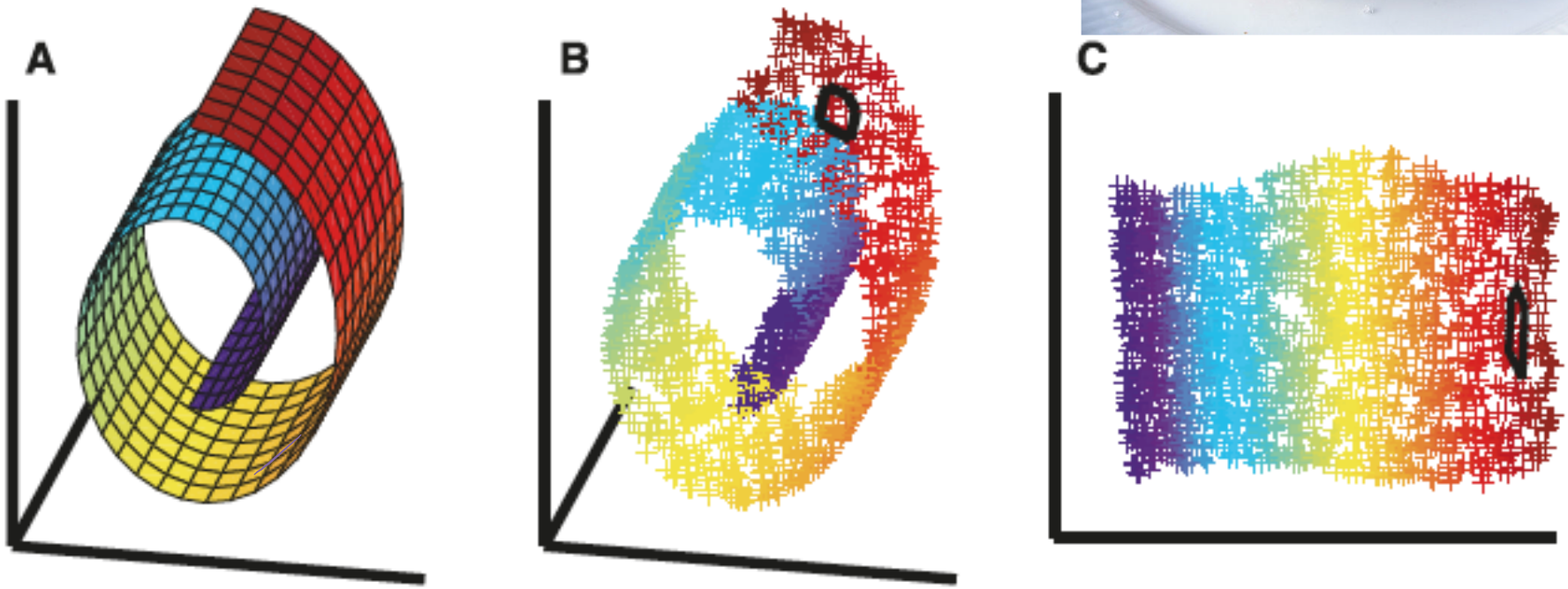
PCA vs. k-Means

- Can think of PCA and k-Means to solve a similar unsupervised learning problem, with different constraints
- Both aim to compress the data with maximum fidelity under constraints on the model complexity
- This insight gives rise to a much broader class of techniques!

  ➔ Matrix factorization, see Computational Intelligence Lab

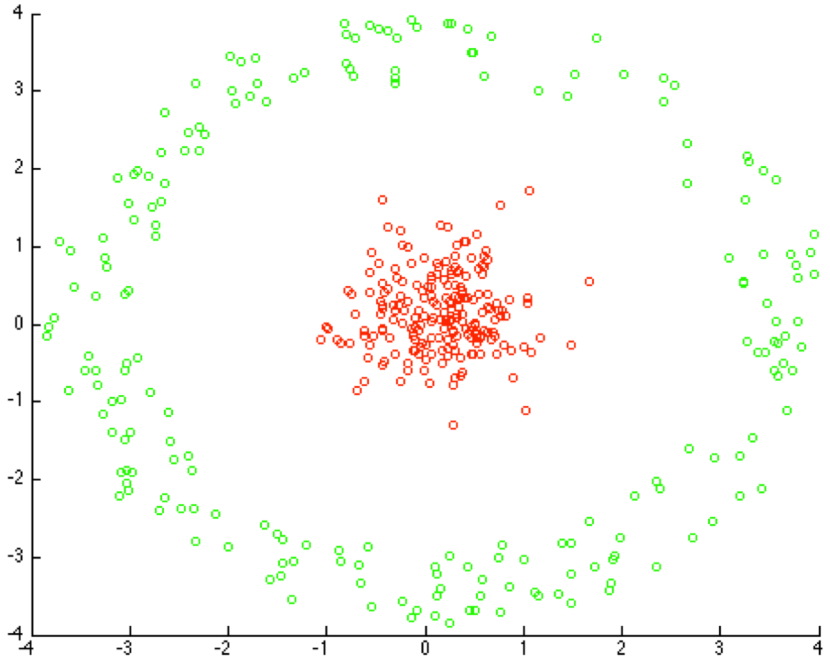
Nonlinear Dimension Reduction

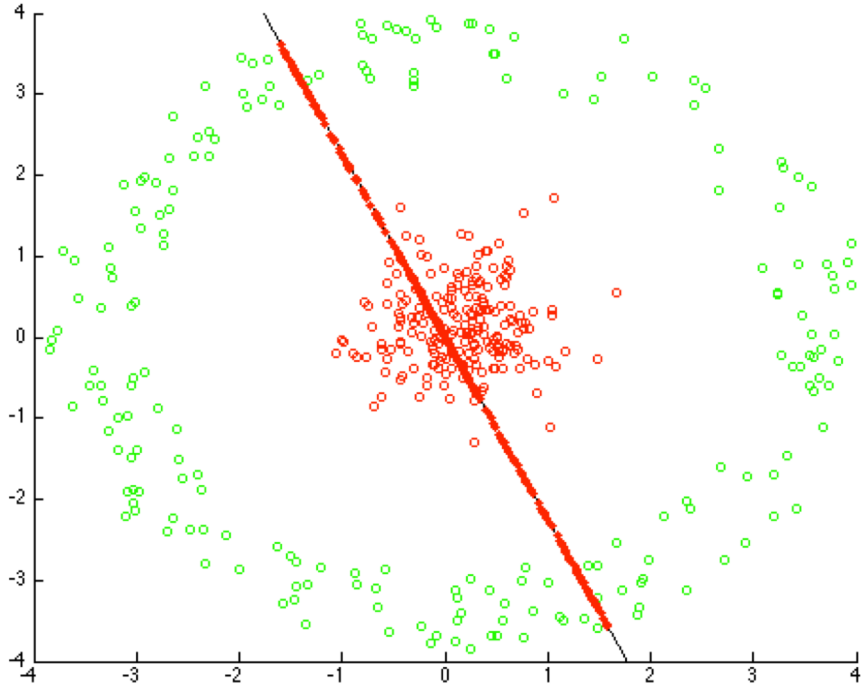
- Motivating Example: „Swiss Roll“



- What is the result of a linear projection?

Another example





Use Kernels!

- *Recall*: In supervised learning, kernels allowed us to solve non-linear problems by reducing them to linear ones in high-dimensional (implicitly represented) spaces
- Can take the same approach for unsupervised learning!

Ansatz: $w = \sum_{i=1}^{n} \alpha_i x_i$

Why: for PCA, w^* is (only) eigenvector of $X^T X$

$\Rightarrow X^T X w = \lambda w \Rightarrow w = \frac{1}{\lambda} \sum_i \beta_i x_i$

($\beta = Xw$, $\sum_i \beta_i x_i = X^T \beta$)

$\Rightarrow \alpha_i = \frac{\beta_i}{\lambda}$