Learning and Intelligent Systems

Unsupervised Learning: Dimension Reduction

Prof. Gunnar Rätsch
Biomedical Informatics (bmi.inf.ethz.ch)
Material: Andreas Krause
We will

- Introduce basic dimension reduction algorithms
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Neural network autoencoders

- Much more details in
  - Computational Intelligence Lab
Basic challenge

Given data set \( D = \{ x_1, \ldots, x_n \} \)

obtain „embedding“ (low-dimensional representation)

\[ z_1, \ldots, z_n \in \mathbb{R}^k \]

Motivation

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

\[ 3 \]
Example: Embedding of faces
[Saul & Roweis]
Typical approaches

- Assume \( D = \{ x_1, \ldots, x_n \} \subseteq \mathbb{R}^d \)

- Obtain mapping \( f : \mathbb{R}^d \rightarrow \mathbb{R}^k \) where \( k \ll d \)

- Can distinguish
  - Linear dimension reduction: \( f(x) = Ax \)
  - 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 $D = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$
- Want to represent data as points on a line $w \in \mathbb{R}^d$ with coefficients $z_1, \ldots, z_n$
- I.e., want $z_i w \approx x_i$, assuming $\mu = \frac{1}{n} \sum_i x_i = 0$

\[
\begin{align*}
\text{assume mean 0} \\
\text{this is v.i.o.g.} \\
(\text{just subtract mean})
\end{align*}
\]
Linear dim. reduction for reconstruction

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

\[ \mathbf{2w} = \mathbf{2}^t \mathbf{w}^t \quad \text{for} \quad \mathbf{2}^t = \alpha \mathbf{2}, \quad \mathbf{w}^t = \frac{1}{\alpha} \mathbf{w} \quad \text{for} \quad \alpha \neq 0 \]

- Optimize over $\mathbf{w}, \mathbf{z}_1, \ldots, \mathbf{z}_n$ jointly:

\[ (\mathbf{w}^*, \mathbf{z}_1^*, \ldots, \mathbf{z}_n^*) = \arg\min_{\mathbf{w}, \mathbf{z}_1, \ldots, \mathbf{z}_n} \sum_{i=1}^n \| \mathbf{z}_i \mathbf{w} - \mathbf{x}_i \|_2^2 \]
Linear dim. reduction for reconstruction

- Given data set \( D = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d \)
- Want \( z_i w \approx x_i \), e.g., minimizing \( \|z_i w - x_i\|^2 \)
- To ensure uniqueness, normalize: \( \|w\|_2 = 1 \)

Optimize over \( w, z_1, \ldots, z_n \) jointly:

\[
(w^*, z^*) = \arg \min_{\|w\|_2 = 1, z} \sum_{i=1}^{n} \|z_i w - x_i\|^2
\]
Solving for $z$ given $w$

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

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

$\mathbf{z} = w^\top \mathbf{x}$
Suppose we consider some vector \( \mathbf{w} \). What can we say about the optimal \( z \)?

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

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

Thus, we effectively solve a *regression* problem, interpreting \( \mathbf{x} \) as features and \( z \) as labels!
Solving for $z$ given $w$

- Want to solve
  \[(w^*, z^*) = \arg\min_{||w||_2 = 1, z} \sum_{i=1}^{n} ||z_i w - x_i||_2^2\]

- **Note:** For any fixed $||w||_2 = 1$ it holds that $z_i^* = w^T x_i$
  therefore, only need
  \[w^* = \arg\min_{||w||_2 = 1} \sum_{i=1}^{n} ||ww^T x_i - x_i||_2^2\]

\[\sum_{i=1}^{n} (ww^T x_i - x_i)^T (ww^T x_i - x_i)\]
\[= \sum_{i=1}^{n} (x_i^T \overbrace{ww^T x_i} + x_i^T x_i - 2x_i^T w^T x_i)\]
\[= \sum_{i=1}^{n} x_i^T (ww^T x_i - 2w^T x_i + x_i^T x_i)\]
\[= \sum_{i=1}^{n} x_i^T (x_i^T x_i - 2w^T x_i)\]
\[= \sum_{i=1}^{n} x_i^T (x_i^T x_i) - \sum_{i=1}^{n} 2x_i^T w^T x_i\]
\[= \sum_{i=1}^{n} x_i^T x_i - \sum_{i=1}^{n} 2x_i^T w^T x_i = \sum_{i=1}^{n} ||x_i||^2 - 2 \sum_{i=1}^{n} (w^T x_i)^2\]
Solving for $\mathbf{w}$

The objective is equivalent to

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

is equivalent to

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

\[\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \text{ is empirical covariance of data (assuming } \mathbf{m}=0)\]
Solving for $\mathbf{w}$

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 \( \mathbf{w} \)

The optimal solution to

\[
\begin{align*}
\mathbf{w}^* &= \arg \max_{\|\mathbf{w}\|_2=1} \mathbf{w}^T \Sigma \mathbf{w} \\
\mu &= \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T
\end{align*}
\]

is given by the principal eigenvector of \( \Sigma \)

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

\[
\Sigma = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^T, \quad \lambda_1 \geq \cdots \geq \lambda_d \geq 0
\]

Proof: Take any \( \mathbf{v} = \sum_{i=1}^{d} \alpha_i \mathbf{v}_i \); Nontrivial \( \mathbf{w}^T \mathbf{v} = \sum_{i=1}^{d} \alpha_i^2 \) unless \( \mathbf{w} = \mathbf{v} \) (if \( \mathbf{w} = \mathbf{v} \) or \( \mathbf{w} \) is parallel to \( \mathbf{v} \), then \( \mathbf{w}^T \mathbf{v} = \sum_{i=1}^{d} \alpha_i^2 \) is maximized).

\[
\mathbf{w}^T \Sigma \mathbf{v} = \left( \sum_{i=1}^{d} \alpha_i \mathbf{v}_i \right)^T \Sigma \left( \sum_{j=1}^{d} \alpha_j \mathbf{v}_j \right) = \sum_{i=1}^{d} \alpha_i^2 \mathbf{v}_i^T \Sigma \mathbf{v}_i = \sum_{i=1}^{d} \alpha_i^2 \lambda_i
\]

\[
= \sum_{i=1}^{d} \alpha_i^2 \lambda_i
\]

Maximized for \( \alpha_1 \neq 0, \alpha_2 \ldots \alpha_d = 0 \).
How about $k>1$?

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

$$ (W, z_1, \ldots, z_n) = \arg \min \sum_{i=1}^{n} ||Wz_i - x_i||^2 $$

where $W \in \mathbb{R}^{d \times k}$ is orthogonal, $z_1, \ldots, z_n \in \mathbb{R}^k$

$$ W^T W = I_k = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{k \times k} $$

- 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 \( D = \{ x_1, \ldots, x_n \} \subseteq \mathbb{R}^d, 1 \leq k \leq d \)

\[
\Sigma = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \quad \mu = \frac{1}{n} \sum_{i} x_i = 0
\]

- The solution to the PCA problem

\[
(W, z_1, \ldots, z_n) = \arg \min \sum_{i=1}^{n} \| Wz_i - x_i \|_2^2
\]

where \( W \in \mathbb{R}^{d \times k} \) is orthogonal, \( z_1, \ldots, z_n \in \mathbb{R}^k \)

is given by \( W = (v_1 \mid \ldots \mid v_k) \) and \( z_i = W^T x_i \)

where

\[
\Sigma = \sum_{i=1}^{d} \lambda_i v_i v_i^T \quad \lambda_1 \geq \cdots \geq \lambda_d \geq 0
\]
PCA is a projection

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

This projection is chosen to minimize the reconstruction error (measured in Euclidean norm)
PCA Illustration
Connection to SVD

- Can obtain PCA through **Singular-Value Decomposition**

*Recall*: Can represent any \( \mathbf{X} \in \mathbb{R}^{n \times d} \) as \( \mathbf{X} = \mathbf{USV}^T \)

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**

\[
\mathbf{X} = \mathbf{USV}^T \quad \mathbf{U}^\top \mathbf{U} = \mathbf{I}_n \quad \mathbf{V}^\top \mathbf{V} = \mathbf{I}_d \quad \mathbf{S} = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_d \end{bmatrix}
\]
PCA via SVD

- **Recall**: Can represent any $X \in \mathbb{R}^{n \times d}$ as $X = USV^T$ where $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{d \times d}$ are orthogonal, and $S \in \mathbb{R}^{n \times d}$ is diagonal (wlog in decreasing order).

- The top $k$ principal components are exactly the first $k$ columns of $V$

Recall that PCs $W$ are top $k$ eigenvectors of $\Sigma$

$\Sigma = \frac{1}{n} X^TX$

$\Sigma = \frac{1}{n} V S^T U^T S V^T = \frac{1}{n} V S^T S V^T$, $S S^T = \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 \end{pmatrix}$

hence $V$ diagonalizes $\Sigma$
Common PCA Usecases

- Visualization \((k=1, 2, 3)\)
- Feature induction
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:

\[(W, z_1, \ldots, z_n) = \arg \min_{W} \sum_{i=1}^{n} \|Wz_i - x_i\|^2_2\]

where \(W \in \mathbb{R}^{d \times k}\) is orthogonal, \(z_1, \ldots, z_n \in \mathbb{R}^k\)

k-Means problem: (equivalent formulation)

\[(W, z_1, \ldots, z_n) = \arg \min_{W} \sum_{i=1}^{n} \|Wz_i - x_i\|^2_2\]

\[W = (\mu_1, \ldots, \mu_k)\]

where \(W \in \mathbb{R}^{d \times k}\) arbitrary and \(z_1, \ldots, z_n \in E_k\)

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!
Recall PCA for $k=1$

- Optimal solution to PCA problem solves, for $\Sigma = X^T X$

$$\arg \max_{||z||_2=1} z^T X^T X z = \arg \max_{||z||_2=1} \sum_{i=1}^{n} (z^T x_i)^2$$
Recall PCA for k=1

- Optimal solution to PCA problem solves, for $\Sigma = X^T X$

$$\arg \max_{||z||_2=1} z^T X^T X z = \arg \max_{||z||_2=1} \sum_{i=1}^{n} (z^T x_i)^2$$

- Applying feature maps, using $z = \sum_{j=1}^{n} \alpha_j \phi(x_i)$ and observing $||z||_2^2 = \alpha^T K \alpha$

$$\arg \max_{||z||_2=1} \sum_{i=1}^{n} \left( z^T \phi(x_i) \right)^2 = \arg \max_{\alpha^T K \alpha=1} \sum_{i=1}^{n} \left( \sum_{j} \alpha_j \phi(x_j)^T \phi(x_i) \right)^2$$

$$= \arg \max_{\alpha^T K \alpha=1} \sum_{i=1}^{n} \left( \sum_{j} \alpha_j k(x_j, x_i) \right)^2 = \arg \max_{\alpha^T K \alpha=1} \sum_{i=1}^{n} \left( \alpha^T K_i \right)^2$$

$$= \arg \max_{\alpha^T K \alpha=1} \alpha^T K^T K \alpha$$
Kernel PCA (k=1)

- The Kernel-PCA problem (k=1) requires solving

\[ \alpha^* = \arg \max_{\alpha^T K \alpha = 1} \alpha^T K^T K \alpha \]

- The optimal solution is obtained in closed form from the eigendecomposition of \( K \):

\[ \alpha^* = \frac{1}{\sqrt{\lambda_1}} v_1 \]

\[ K = \sum_{i=1}^{n} \lambda_i v_i v_i^T \quad \lambda_1 \geq \cdots \geq \lambda_d \geq 0 \]
For general $k > 1$, the Kernel Principal Components are given by $\alpha^{(1)}, \ldots, \alpha^{(k)} \in \mathbb{R}^n$

where

$$\alpha^{(i)} = \frac{1}{\sqrt{\lambda_i}} v_i$$

is obtained from:

$$K = \sum_{i=1}^{n} \lambda_i v_i v_i^T \quad \lambda_1 \geq \cdots \geq \lambda_d \geq 0$$

Given this, a new point $x$ is projected as $z \in \mathbb{R}^k$
For general \( k > 1 \), the Kernel Principal Components are given by \( \alpha^{(1)}, \ldots, \alpha^{(k)} \in \mathbb{R}^n \) where
\[
\alpha^{(i)} = \frac{1}{\sqrt{\lambda_i}} v_i
\]
is obtained from:
\[
K = \sum_{i=1}^{n} \lambda_i v_i v_i^T \quad \lambda_1 \geq \cdots \geq \lambda_d \geq 0
\]
Given this, a new point \( x \) is projected as \( z \in \mathbb{R}^k \)
\[
z_i = \sum_{j=1}^{n} \alpha_j^{(i)} k(x, x_j)
\]
**Side note**: Applying k-means on kernel-principal components is sometimes called **Kernel-k-means** or **Spectral Clustering**