IntroML - Lecture Notes Week 9

Ruben Schenk, ruben.schenk@inf.ethz.ch

July 17, 2022

1 Unsupervised Learning: Dimension Reduction

1.1 Introduction

The basic challenge is posed as follows: Given a data set $D = \{x_1, ..., x_n\}$ with $x_i \in \mathbb{R}^d$, obtain an **embedding** (low-dimensional representation) $z_1, ..., z_n \in \mathbb{R}^k$ where typically $k << d$.

One might want to do this for several reasons:

- Visualization ($k = 1, 2, 3$)
- Regularization (model selection)
- Unsupervised feature discovery (i.e. determining features from data)
- etc.

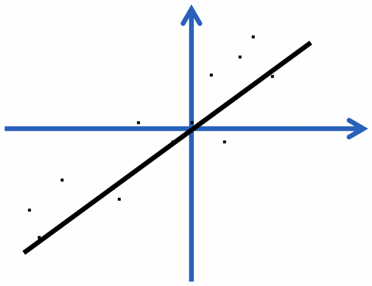
Note: Our focus is on model-based approaches, i.e.:

- Given: Data $D = \{x_1, ..., x_n\} \subseteq \mathbb{R}^d$
- Goal: Obtain a mapping $f : \mathbb{R}^d \to \mathbb{R}^k$ where usually $k << d$
- We can distinguish:
  - Linear dimension reduction $f(x) = Ax$
  - Nonlinear dimension reduction (parametric or non-parametric)

1.2 Dimension Reduction

Linear dimension redcution can be seen as compression. The motivation behind this process is that low-dimensional representation should allow to compress the original data and allow for a accurate reconstruction.

Let us consider a simple example for $k = 1$. Given is a data set $D = \{x_1, ..., x_n\} \subseteq \mathbb{R}^d$, assumed to be centered, i.e. $\mu = \frac{1}{n}\sum_i x_i = 0$. We want to represent the data as points on a line $x_i \approx z_i w$ with coefficients $w \in \mathbb{R}^d$.



In other words, we want $x_i \approx z_i w$ minimizing $||z_i w - x_i||_2^2$. To ensure the uniqueness of the solution, we normalize w, i.e. $||w||_2 = 1$. We want to optimize jointly over w, z_1, $z_2, ...$:

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

In our $k = 1$ case, the optimal z is given by:

$$z_i^* = w^T x_i$$

Thus, we effectively solve a *regression* problem, interpreting x as features and z as labels. Since for any fixed $||w||_2 = 1$, it holds that $z_i^* = w^T x_i$. Therefore, we only need:

$$w^* = \arg\min_{||w||_2=1} \sum_{i=1}^n ||ww^T x_i - x_i||_2^2,$$

which is equivalent to

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

which is furthermore equivalent to

$$w^* = \arg\max_{||w||_2=1} w^T \Sigma w,$$

where $\Sigma = \frac{1}{n}\sum_{i=1}^n x_i x_i^T$ is the *empirical covariance,* assuming the data is centered (i.e. $\mu = \frac{1}{n}\sum_i x_i = 0$).

Finally, the optimal solution to $w^* = \arg\max_{||w||_2=1} w^T\Sigma w$ is given by the **principal eigenvector** of Σ, i.e. $w = v_1$ where, for $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$,

$$\Sigma = \sum_{i=1}^d \lambda_i v_i v_i^T.$$

But what if $k > 1$? Suppose we wish to project more than one dimension. Thus we want:

$$(W,\, z_1, ...,\, z_n) = \arg\min_{W^T W = I_k,\, z} \sum_{i=1}^n ||W z_i - x_i||_2^2,$$

where $W \in \mathbb{R}^{d\times k}$ is *orthogonal,* and $z_1, ...,\, z_n \in \mathbb{R}^k$. This is called the *principal component analysis* problem and its solution can be obtained in closed form even for $k > 1$.

1.3 Principle Component Analysis (PCA)

The **Principal Component Analysis (PCA)** problem is as follows:

Given centered data $D = \{x_1, ..., x_n\} \subseteq \mathbb{R}^d$ with $\mu = \frac{1}{n}\sum_i x_i = 0$ and $\Sigma = \frac{1}{n}\sum_{i=1}^n x_i x_i^T$, the solution to the PCA problem

$$(W,\, z_1, ...,\, z_n) = \arg\min_{W^T W = I_k,\, z} \sum_{i=1}^n ||W z_i - x_i||_2^2,$$

where $1 \leq k \leq d$, $W \in \mathbb{R}^{d\times k}$ is orthogonal and $z_1, ...,\, z_n \in \mathbb{R}^k$, is given by

$$W = (v_1 \mid \cdots \mid v_k) \text{ and } z_i = W^T x_i.$$

Hereby: $\Sigma = \sum_{i=1}^d \lambda_i v_i v_i^T$ and $\lambda_1 \geq \cdots \geq \lambda_d$.

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 the Euclidean norm).

One might remember that we can obtain PCA through the *Singular-Value Decomposition (SVD).* We recall that any $X \in \mathbb{R}^{n\times d}$ can be represented 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 (w.l.o.g. in decreasing order). Its entries are called *singular values.* The top k principal components are exactly the first k columns of V:

$$n\Sigma = X^T X = VS^T U^T USV^T = VS^T SV^T = VDV^T.$$

Finally, we can compare PCA and k-means:

PCA-Problem:

$$(W,\, z_1, ...,\, z_n) = \arg \min_{W^T W = I_k,\, z} \sum_{i=1}^{n} ||Wz_i - x_i||_2^2,$$

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

k-means problem (equivalent formulation):

$$(W,\, z_1, ...,\, z_n) = \arg \min_{W,\, z} \sum_{i=1}^{n} ||Wz_i - x_i||_2^2,$$

where $W \in \mathbb{R}^{d \times k}$ is *arbitrary,* and $z_1, ...,\, z_n \in E_k$ for $E_k = \{e_1, ...,\, e_k\}$ are all unit vectors.

In summary:

- We can think of PCA and k-means as options 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.

1.4 Kernel PCA

1.5 Neural Network Autoencoders