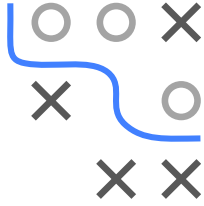
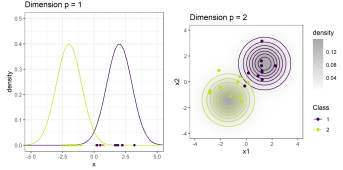
Introduction to Machine Learning



Curse of Dimensionality
Curse of Dimensionality - Examples
Learning Algorithms



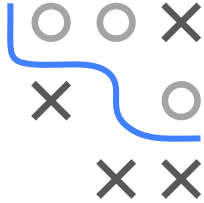
Learning goals

- See how the performance of k-NN and the linear model deteriorates in high-dimensional spaces

EXAMPLE: K-NN

Let us look at the performance of algorithms for increasing dimensionality. First, we consider the k-NN algorithm:

- In a high dimensional space, data points are spread across a huge space.
- The distance to the **next neighbor** $d_{NN1}(\mathbf{x})$ becomes extremely large.
- The distance might even get so large that all points are **equally far** away - we cannot really determine the nearest neighbor anymore.

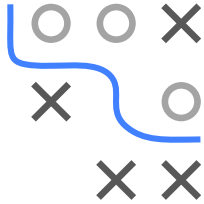


EXAMPLE: K-NN

The consequences for the k-nearest neighbors approach can be summarized as follows:

- At constant sample size n and growing p, the distance between the observations increases
  $\rightarrow$ the coverage of the p-dimensional space decreases,
  $\rightarrow$ every point becomes isolated / far way from all other points.
- The size of the neighborhood $N_k(x)$ also "increases" (at constant k)
  $\rightarrow$ it is no longer a "local" method.
- Reducing k dramatically does not help much either, since the fewer observations we average, the higher the variance of our fit.

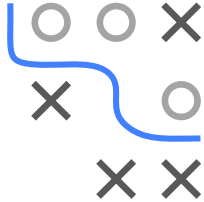
$\rightarrow$ k-NN estimates get more inaccurate with increasing dimensionality of the data.



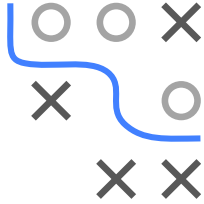
EXAMPLE: LINEAR MODEL

We also investigate how the linear model behaves in high dimensional spaces.

- We take the Boston Housing data set, where the value of houses in the area around Boston is predicted based on 13 features describing the region (e.g., crime rate, status of the population, etc.).
- We train a linear model on the data consisting of 506 observations.
- We artificially create a high-dimensional dataset by adding $100, 200, 300, \ldots$ noise variables (containing no information at all) and look at the performance of a linear model trained on this modified data (10 times repeated 10-fold CV).



COD: WAYS OUT



Many methods besides k-NN struggle with the curse of dimensionality. A large part of ML is concerned with dealing with this problem and finding ways around it.

Possible approaches are:

- Increasing the space coverage by gathering more observations (not always viable in practice!)
- Reducing the number of dimensions before training (e.g. by using domain knowledge, PCA or feature selection)
- Regularization