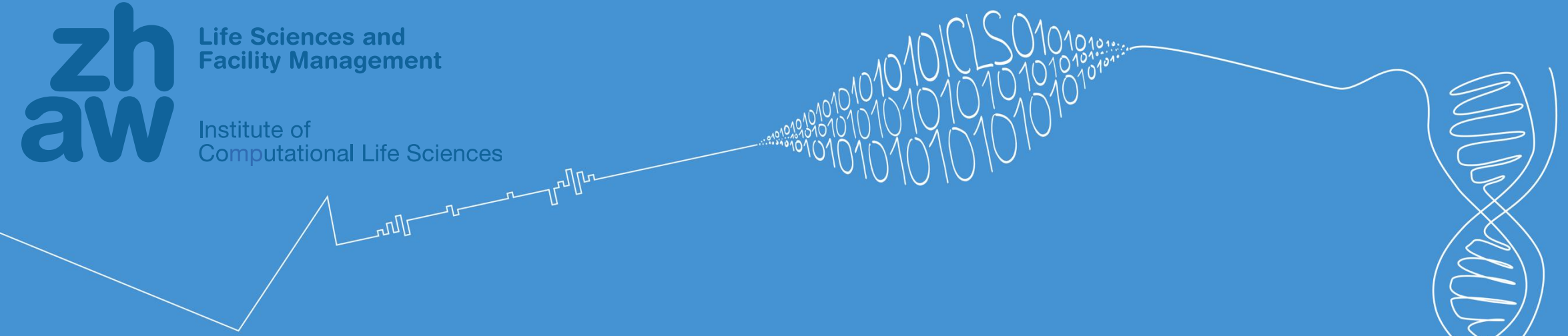
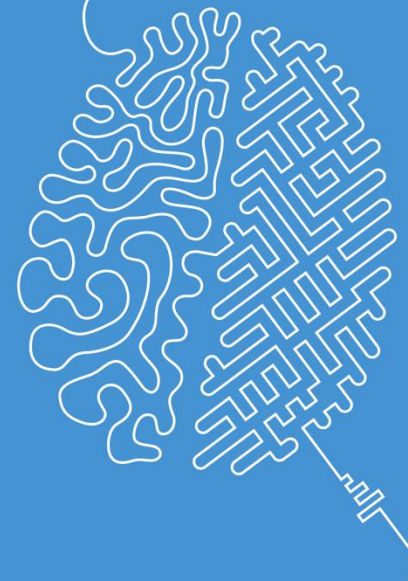
Life Sciences and
Facility Management

Institute of
Computational Life Sciences





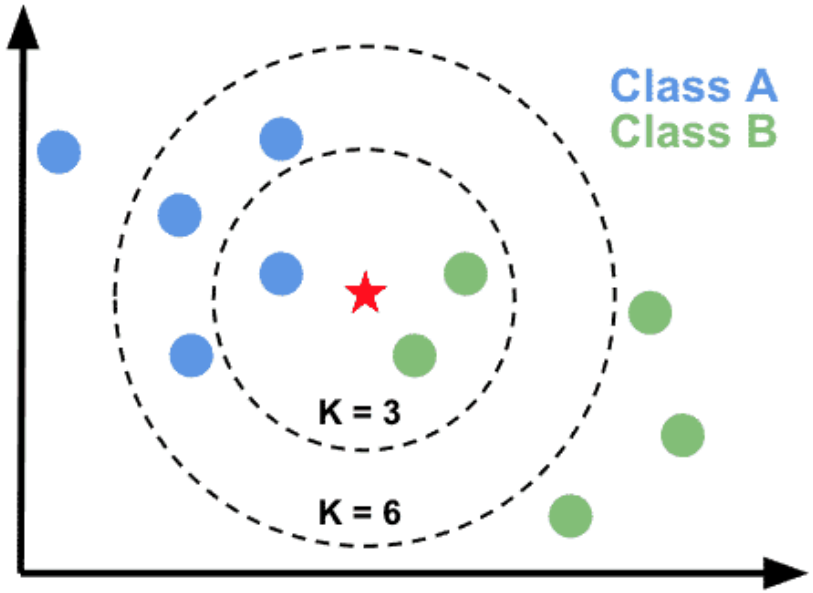
k-Nearest Neighbors (kNN)

Machine Learning

Norman Juchler

Outline

- The kNN method is an intuitive method to assign a class to a new data point based on the majority class of its k nearest neighbors.
- It can be used for classification (supervised) and clustering (unsupervised) and even regression.



Note: Don't confuse k-nearest neighbors with k-means clustering!

Instance-based learning

- Model-based or parametric learning:
  - Most machine learning methods aim to abstract a model from the data: $y = f(x|\theta)$
- Instance-based learning:
  - Predictions are made directly using specific instances from the training data
  - Memory-based: store the training data and use them directly during prediction
  - Local decisions: Predictions are made based on the local neighborhood of the input data
  - There's typically no training phase to develop an instance-based model

- The kNN algorithm is an example of instance-based learning.

kNN algorithm

- kNN uses local information from *nearby* training examples to predict new labels.

Steps

- Choose k
- Calculate distance between an **inference point** and all (relevant) points in the training data
- Identify the k nearest neighbors
- Aggregate neighbor information
  - Classification: Majority vote (or weighted majority vote)
  - Regression: Average (or weighted average) of neighbors

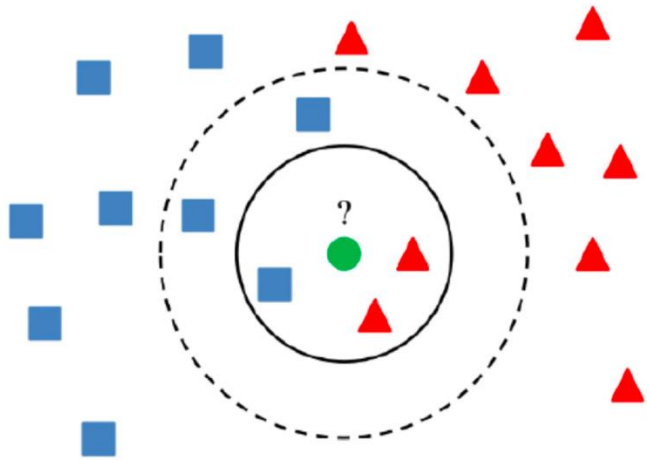


Illustration of kNN for supervised learning: Search for the k nearest neighbors of the **inference point** for:
— k = 3
- - k = 5

kNN algorithm

- kNN uses local information from *nearby* training examples to predict new labels.

Notes:

- It is common to weight neighbors with the inverse of their distance, such that closer points have greater influence:

$$\text{weight} = \frac{1}{\text{distance}}$$

- Weights should also be applied in case of imbalanced data.
- Since feature-space distances combine different units, normalizing the training data is key!

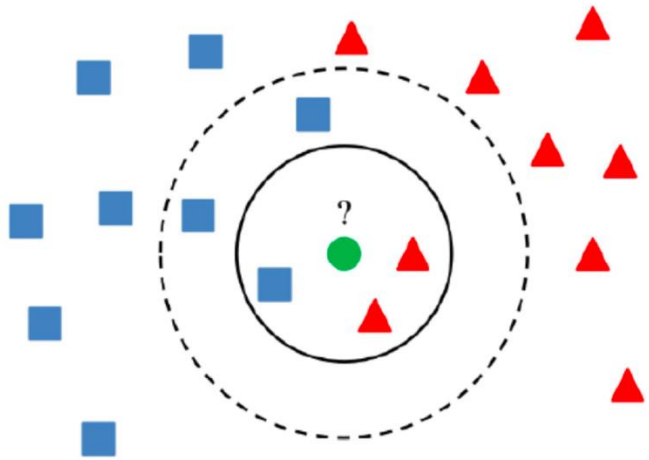
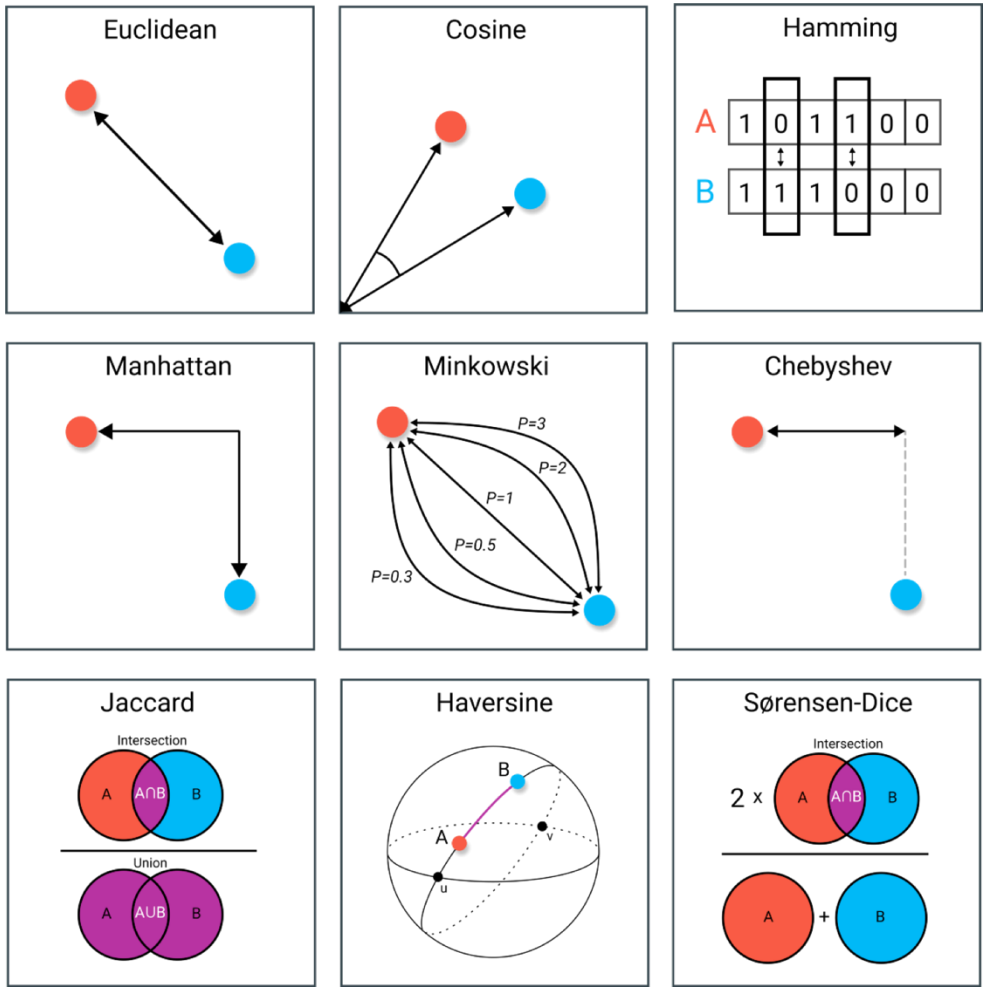


Illustration of kNN for supervised learning: Search for the k nearest neighbors of the **inference point** for:
— k = 3
- - k = 5

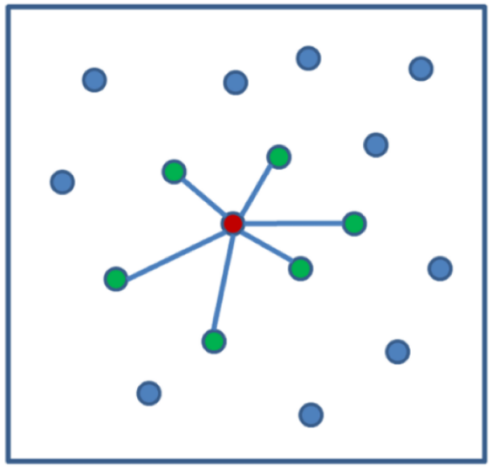
Similarity / distance measures

- Euclidean $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Manhattan $\sum_{i=1}^{n}|p_i - q_i|$
- Minkowski $\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}}$
- Cosine $\cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n}A_iB_i}{\sqrt{\sum_{i=1}^{n}A_i^2}\sqrt{\sum_{i=1}^{n}B_i^2}}$
- Hamming
- ...
- and any application specific distances...

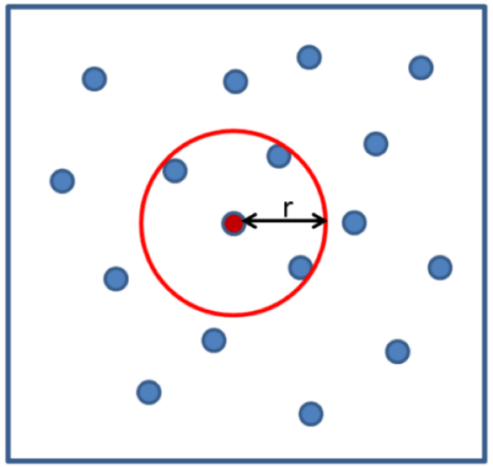


Variant: Radius neighbors

- A fixed number of neighbors will correspond to small distances in the center and large distances in the tails of the distributions.
- Solution: Use radius-neighbors
- Using a fixed distance
  - helps to respect the measurement uncertainty
  - improves extrapolation ability of the input variables
- Fast implementations exist (using range searching)



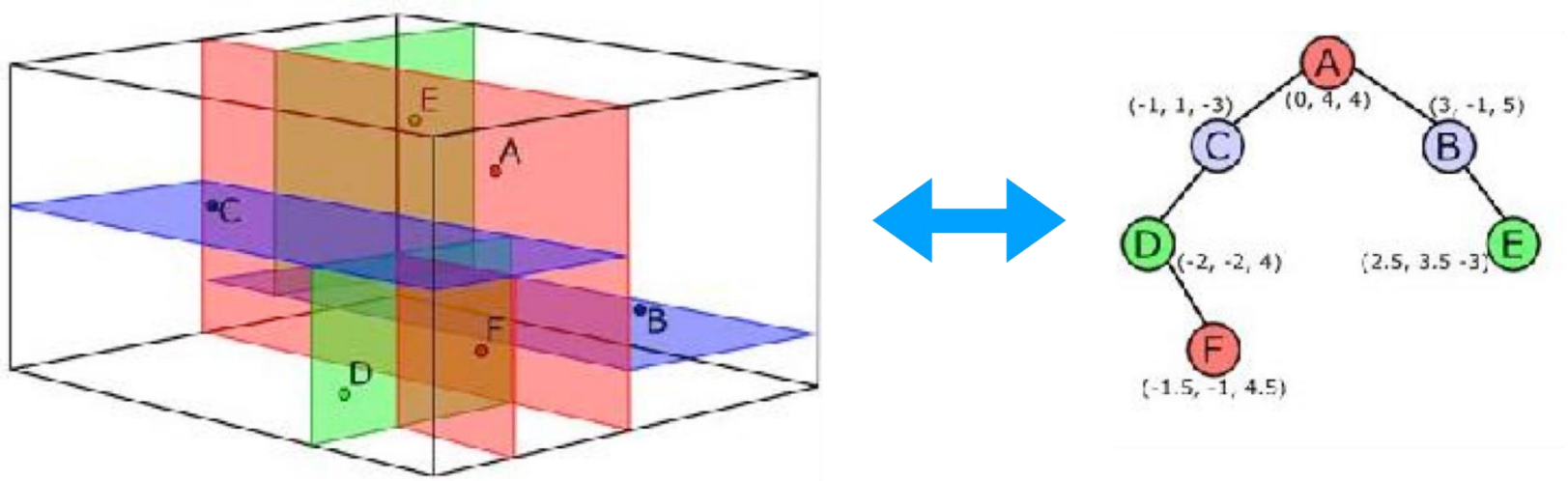
K-NN Search



Radius Search

How is it done? Fast neighbor search

- Range search in binary trees
  - **Binary trees** can store all points in an *n-dimensional feature* space (KD-tree)
  - At each node's decision the search-space is halted → O(log n) search time
  - **Range search** can find all nodes with a box by just traversing the edges



- Locality-Sensitive Hashing (LSH)
  - A simple and scalable method for fast indexing and retrieval in high dimensions.

Question

What do you think of the generalization capability of a kNN model?

Further ~~reading~~ watching

- StatQuest: K-nearest neighbors (5 min)
- The part about heatmap requires this video:
  StatQuest: Hierarchical clustering (11min)

- Don’t confuse kNN with k-means clustering!