*** Applied Machine Learning Fundamentals ***
k-Nearest Neighbors

Daniel Wehner

SAP SE

October 30, 2019

DHBW
Duale Hochschule
Baden-Württemberg

Find all slides on GitHub

Agenda October 30, 2019

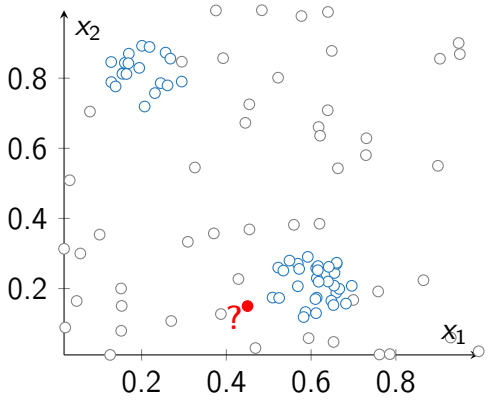
Section:

Introduction

Introduction

- **Basic idea:** Predict the class label based on nearby known examples
- Instance-based learning, a. k. a. lazy learning

We do not learn any model, the data speaks for itself!



Derivation of the Algorithm

- Unconditional density:

$$p(\boldsymbol{x}) = \frac{k}{n \cdot v}$$

- Class priors:

$$p(\mathcal{C}_j) = \frac{n_j}{n}$$

Remember non-parametric density estimation?

Combine using Bayes' theorem:

$$p(\mathcal{C}_j|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_j) \cdot p(\mathcal{C}_j)}{p(\boldsymbol{x})} = \frac{\frac{k_j}{n_j \cdot v} \cdot \frac{n_j}{n}}{\frac{k}{n \cdot v}} = \frac{k_j}{k} \tag{1}$$

Section:

Distance Metrics

Distance Metrics

- How to measure the distance between two data points i and j?
  $\Rightarrow$ **distance metrics**
- Let d be a function $d : (u, v) \mapsto \mathbb{R}^+$ (including 0)
- Function d has the following properties:

1. $d(u, v) = d(v, u)$ **(commutativity)**
2. $d(u, v) = 0 \Rightarrow u = v$
3. $d(u, k) \leqslant d(u, v) + d(v, k)$ **(triangle inequality)**

Distance Metrics (Ctd.)

Minkowski distance:

$$d_p(u, v) = \left(\sum_{j=1}^{m} |x_j^{(u)} - x_j^{(v)}| \right)^{1/p} \tag{2}$$

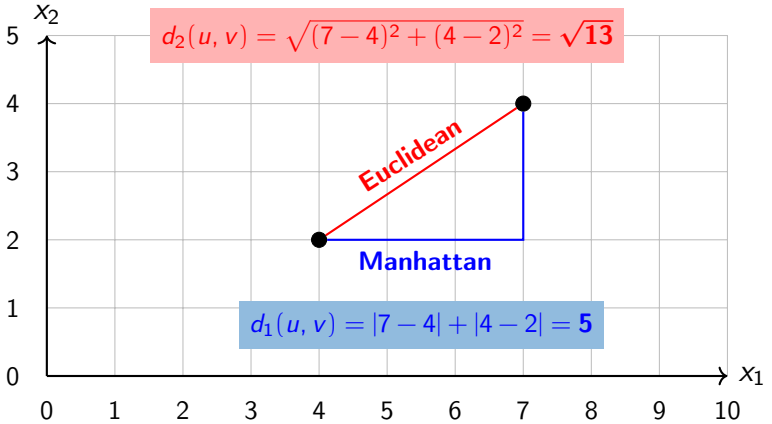
Manhattan distance: $(p = 1)$

$$d_1(u, v) = \sum_{j=1}^{m} |x_j^{(u)} - x_j^{(v)}|$$

Euclidean distance: $(p = 2)$

$$d_2(u, v) = \sqrt{\sum_{j=1}^{m} |x_j^{(u)} - x_j^{(v)}|^2}$$

Distance Metrics (Ctd.)



Cosine Similarity

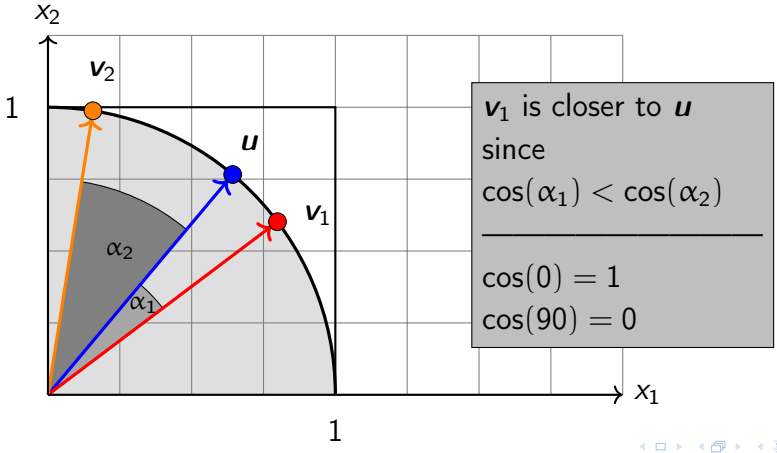
- **Similarity metrics** are an alternative to distance metrics
- **Example: Cosine similarity**
- The cosine similarity of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is the cosine of the angle:

$$\cos \measuredangle(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\|\boldsymbol{a}\|_2 \cdot \|\boldsymbol{b}\|_2} = \frac{\sum_{j=1}^{m} a_j \cdot b_j}{\sqrt{\sum_{j=1}^{m} (a_j)^2} \cdot \sqrt{\sum_{j=1}^{m} (b_j)^2}} \tag{3}$$

- The dot product is defined as:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \|\boldsymbol{a}\|_2 \cdot \|\boldsymbol{b}\|_2 \cdot \cos \measuredangle(\boldsymbol{a}, \boldsymbol{b}) \tag{4}$$

Cosine Similarity (Ctd.)



v_1 is closer to u
since
$\cos(\alpha_1) < \cos(\alpha_2)$

$\cos(0) = 1$
$\cos(90) = 0$

Section:

k-nearest Neighbors Algorithm

Prediction with *k*-Nearest Neighbors (Ctd.)

k-nearest neighbors algorithm:

1. Calculate the **pairwise distances** between the new data point and all data points in the data set
2. Sort the data points by distances **in ascending order**
   (if similarity metrics are used, sort in descending order)
3. Look at the first k examples and **count how often each class occurs**
4. Predict the class with **the maximum score**

❶ Calculation of Distances

v	x_1	x_2	$\mathcal{C}$	$d_2(u, v)$
1	0.66	0.24	1	0.23
2	0.25	0.79	1	0.67
3	0.16	0.81	1	0.73
4	0.57	0.21	1	0.13
5	0.21	0.72	1	0.62
6	0.66	0.27	1	0.24
7	0.27	0.11	0	0.19
8	0.39	0.13	0	0.07
9	0.39	0.86	0	0.71
10	0.44	0.67	0	0.52
11	0.31	0.33	0	0.23
12	0.03	0.51	0	0.55
⋮	⋮	⋮	⋮	⋮

- $\boldsymbol{x}^{(u)} = (0.45, 0.15)$
- Calculate the **Euclidean distance** between $\boldsymbol{x}^{(u)}$ and all other data points $\boldsymbol{x}^{(v)}$

Prediction is expensive!

❷/❸/❹ Prediction of the Class Label

- Let k be set to 10
- Step ❷: Sort data set by distances *(cf. table)*
- Step ❸: Count the classes
  - **Class 0:** 3
  - **Class 1:** 7
- Step ❹: **Predict class 1!**

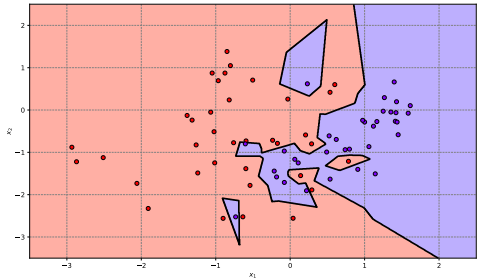
x_1	x_2	$\mathcal{C}$	$d_2(u, v)$
0.51	0.17	1	0.06
0.39	0.13	0	0.07
0.52	0.17	1	0.08
0.43	0.23	0	0.08
0.47	0.03	0	0.12
0.52	0.26	1	0.13
0.57	0.21	1	0.13
0.53	0.25	1	0.13
0.58	0.12	1	0.14
0.59	0.13	1	0.14
⋮	⋮	⋮	⋮

Section:

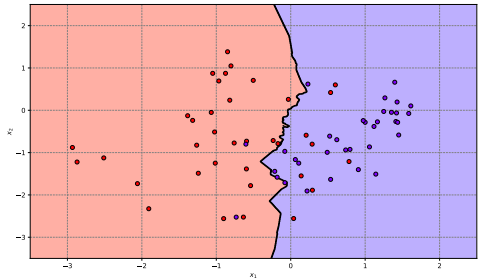
Choice of k

How to choose k?

The choice of k is important:



$k = 1$ (☠ overfitting ☠)



$k = 30$ (about right)

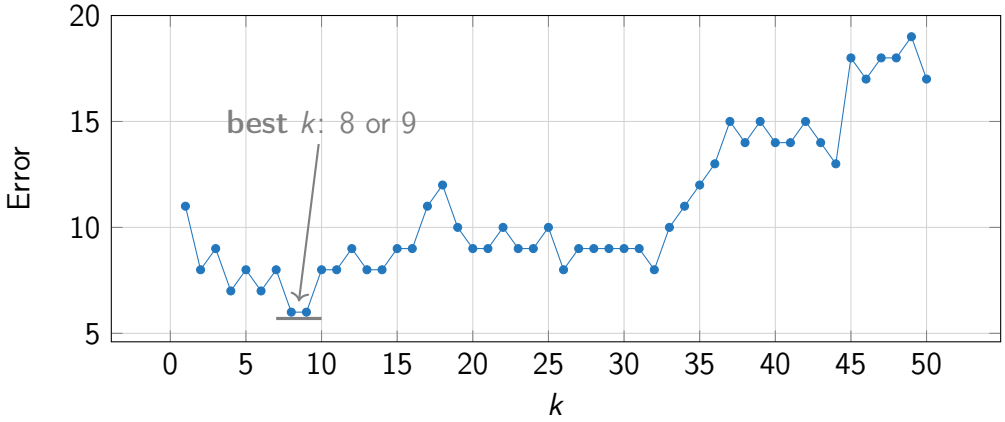
How to choose k? (Ctd.)

- First of all, it is recommended to use **odd values** for k
  (no tie-breaking necessary)
- Compute k depending on the size of the data set $\mathcal{D}$:

$$k = \sqrt{\frac{n}{2}} \qquad \text{or} \qquad k = \sqrt{n} \tag{5}$$

- **Other strategy:** Evaluate different k on a `dev` set and choose the best one

How to choose k? (Ctd.)



Section:

Wrap-Up

Summary

Lecture Overview

Unit I: Machine Learning Introduction

Self-Test Questions

Recommended Literature and further Reading

Thank you very much for the attention!

Topic: *** Applied Machine Learning Fundamentals *** k-Nearest Neighbors
Date: October 30, 2019

Contact:
Daniel Wehner (D062271)
SAP SE
daniel.wehner@sap.com

Do you have any questions?