*** Applied Machine Learning Fundamentals ***
Bayesian Decision Theory

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Winter term 2020/2021

Find all slides on GitHub (DaWe1992/Applied_ML_Fundamentals)

Lecture Overview

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	**Bayesian Decision Theory**
Unit IV	Probability Density Estimation
Unit V	Regression
Unit VI	Classification I
Unit VII	Evaluation
Unit VIII	Classification II
Unit IX	Clustering
Unit X	Dimensionality Reduction

Agenda for this Unit

1. Bayesian Decision Theory
   - Introduction
   - Class Conditional Probabilities
   - Class Priors
   - Bayes' Theorem
   - Bayes' optimal Classifier
2. Naïve Bayes Classifier
   - Assumptions and Algorithm
   - An Example
   - Laplace Smoothing
3. Risk Minimization
   - Error $\neq$ Risk
   - Loss Functions for Risk Minimization
   - Handling of continuous Data
4. Wrap-Up
   - Summary
   - Self-Test Questions
   - Lecture Outlook
   - Recommended Literature and further Reading
   - Meme of the Day

Section:

Bayesian Decision Theory

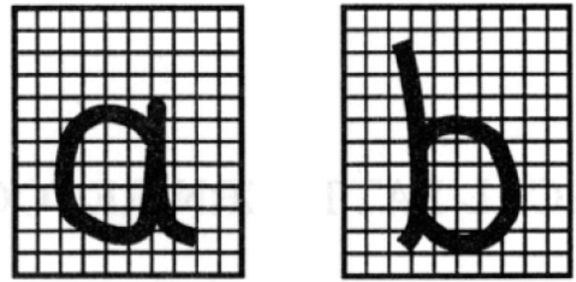
DHBW
Duale Hochschule
Baden-Württemberg

Statistical Methods

- Statistical methods assume that the process that 'generates' the data is governed by the **rules of probability**
- The data is understood to be a set of **random samples** from some underlying **probability distribution**
- This is the reason for the name **statistical machine learning**

The basic assumption about how the data is generated is always there, even if you don't see a single probability distribution!

Running Example: Optical Character Recognition (OCR)



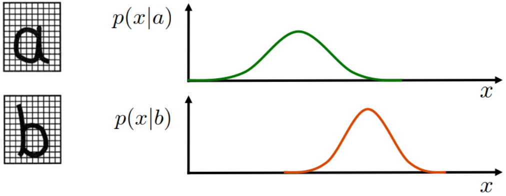
Goal: Classify a new letter so that the probability of a wrong classification is minimized

Class Conditional Probabilities

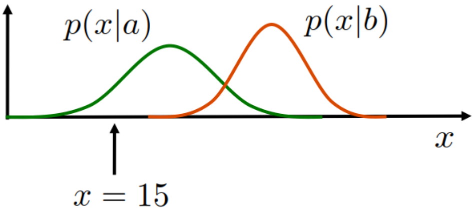
- First concept: **Class conditional probabilities**
- Probability of $\boldsymbol{x}$ given a specific class $\mathcal{C}_k$ is formally written as:

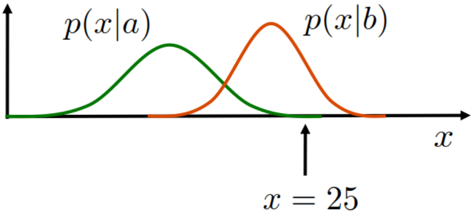
$$p(\boldsymbol{x}|\mathcal{C}_k) \in [0, 1] \tag{1}$$

- $\boldsymbol{x} \in \mathbb{R}^m$ is a feature vector, e.g. # black pixels, height-width ratio, ...



Class Conditional Probabilities (Ctd.)

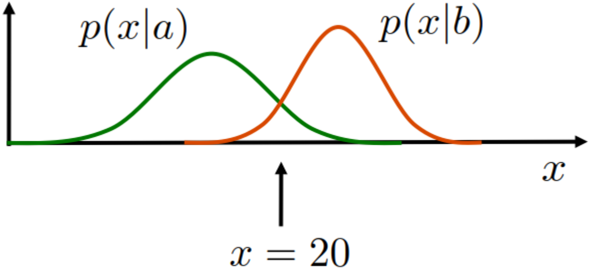




If $x = 15$ we would predict class a, since $p(15|a) > p(15|b)$.
If $x = 25$ we would output class b, since $p(25|b) > p(25|a)$.

Class Conditional Probabilities (Ctd.)

We have a problem!



- **Which class should be chosen now?**
- The conditional probabilities are the same... ☠

Class Prior Probabilities

- Second concept: **Class priors**
- The prior probability of a data point belonging to a particular class $\mathcal{C}_k$

$$\mathcal{C}_1 \equiv a \qquad p(\mathcal{C}_1) = 0.75$$
$$\mathcal{C}_2 \equiv b \qquad p(\mathcal{C}_2) = 0.25$$

- By definition:
  - $0 \leqslant p(\mathcal{C}_k) \leqslant 1, \ \forall k$
  - The sum of all probabilities equals one: $\sum_{k=1}^{|\mathcal{C}|} p(\mathcal{C}_k) = 1$
- **The class prior is equivalent to a prior belief in the class label**

How would you decide now?

How to get the Prior Probabilities?

Count Count's advice:

Simply count the number of instances in each class!

Important

Bayes' Theorem

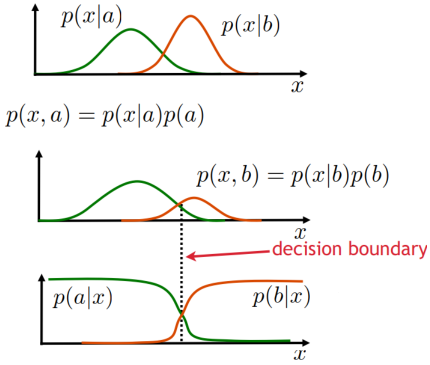
- What we actually want to compute: $p(\mathcal{C}_k|\boldsymbol{x}) \Rightarrow$ **Posterior probability**
- We can compute it by applying **Bayes' theorem**
- This is one of the **most important formulas (!!!)**

$$\overbrace{p(\mathcal{C}_k|\boldsymbol{x})}^{\text{Class posterior}} = \frac{\overbrace{p(\boldsymbol{x}|\mathcal{C}_k)}^{\text{Class cond.}} \cdot \overbrace{p(\mathcal{C}_k)}^{\text{Class prior}}}{\underbrace{p(\boldsymbol{x})}_{\text{Normalization term}}} = \frac{p(\boldsymbol{x}|\mathcal{C}_k) \cdot p(\mathcal{C}_k)}{\sum_{j=1}^{|\mathcal{C}|} p(\boldsymbol{x}|\mathcal{C}_j) \cdot p(\mathcal{C}_j)} \tag{2}$$

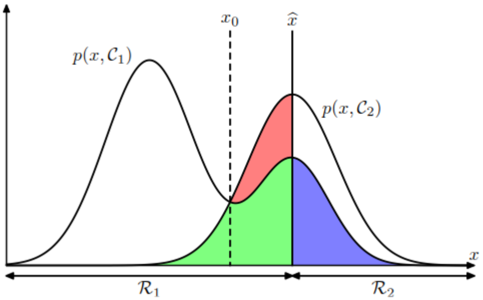
Calculation of the Posterior Probability

- By applying Bayes' theorem we can compute the posterior
- Simply plug ❶ and ❷ into Bayes' theorem
  1. Class prior probabilities
  2. Class conditional probabilities

We get the final **decision boundary**



Error Minimization



$$p(error) = p(x \in \mathcal{R}_1, \mathcal{C}_2) + p(x \in \mathcal{R}_2, \mathcal{C}_1)$$

$$= \overbrace{\int_{\mathcal{R}_1} p(x|\mathcal{C}_2) \cdot p(\mathcal{C}_2)\,\mathrm{d}x}^{\text{red + green area}} + \underbrace{\int_{\mathcal{R}_2} p(x|\mathcal{C}_1) \cdot p(\mathcal{C}_1)\,\mathrm{d}x}_{\text{blue area}}$$

Important

Bayes' optimal Classifier

- Decision rule:
  - Decide $\mathcal{C}_1$, if $p(\mathcal{C}_1|\boldsymbol{x}) > p(\mathcal{C}_2|\boldsymbol{x})$
  - This is equivalent to: *(we don't need the normalization)*

$$p(\boldsymbol{x}|\mathcal{C}_1) \cdot p(\mathcal{C}_1) > p(\boldsymbol{x}|\mathcal{C}_2) \cdot p(\mathcal{C}_2) \tag{3}$$

  - Which is in turn equivalent to:

$$\frac{p(\boldsymbol{x}|\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)} > \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)} \tag{4}$$

- A classifier obeying this rule is called **Bayes' optimal Classifier**

Section:

Naïve Bayes Classifier

DHBW
Duale Hochschule
Baden-Württemberg

A naïve Assumption

- We want to compute $p(\mathcal{C}_k|\boldsymbol{x})$. Recall Bayes' theorem:

Our first classification algorithm!

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k) \cdot p(\mathcal{C}_k)}{p(\boldsymbol{x})} \tag{5}$$

- Assumptions:
  - All features x_j are pairwise conditionally independent ($\Rightarrow$ **naïve**)

$$p(\boldsymbol{x}|\mathcal{C}_k) = p(x_1|\mathcal{C}_k) \cdot p(x_2|\mathcal{C}_k, x_1) \cdot p(x_3|\mathcal{C}_k, x_1, x_2) \cdot \ldots = \prod_{j=1}^{m} p(x_j|\mathcal{C}_k) \tag{6}$$

  - $p(\boldsymbol{x})$ is constant w. r. t. class label $\Rightarrow$ It is omitted

How to get the most probable Class?

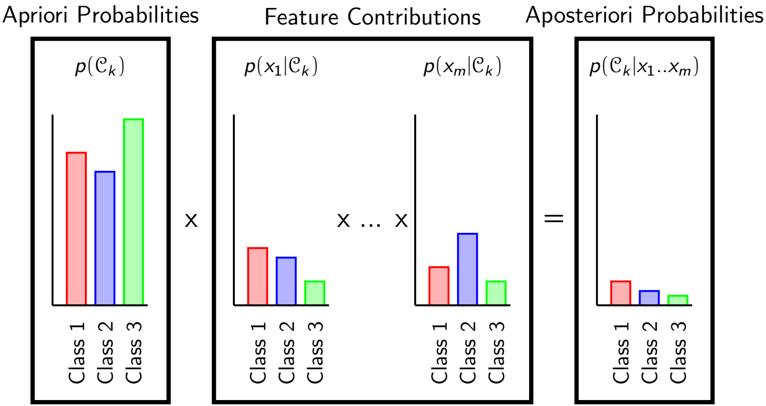
- **Given:**
  - New instance $\boldsymbol{x} = \langle x_1, x_2, \dots, x_m \rangle$ to be classified
  - Finite set of κ classes $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_\kappa\}$
  - Labeled training data ($\Rightarrow$ supervised learning)
- **Wanted:** Most probable class $\mathcal{C}_{MAP}$ (maximum aposteriori) for $\boldsymbol{x}$:

$$\mathcal{C}_{MAP} = \underset{\mathcal{C}_k \in \{\mathcal{C}_1, \dots, \mathcal{C}_\kappa\}}{\arg\max} \; \widehat{p}(\mathcal{C}_k | \boldsymbol{x}) \tag{7}$$

$$= \underset{\mathcal{C}_k \in \{\mathcal{C}_1, \dots, \mathcal{C}_\kappa\}}{\arg\max} \; \widehat{p}(\mathcal{C}_k) \prod_{j=1}^{m} \widehat{p}(x_j | \mathcal{C}_k) \tag{8}$$

$\widehat{p}$ denotes an **approximated** probability

How to get the most probable Class? (Ctd.)



Example Data Set

Outlook	Temperature	Humidity	Wind	PlayGolf
sunny	hot	high	weak	**no**
sunny	hot	high	strong	**no**
overcast	hot	high	weak	**yes**
rainy	mild	high	weak	**yes**
rainy	cool	normal	weak	**yes**
rainy	cool	normal	strong	**no**
overcast	cool	normal	strong	**yes**
sunny	mild	high	weak	**no**
sunny	cool	normal	weak	**yes**
rainy	mild	normal	weak	**yes**
sunny	mild	normal	strong	**yes**
overcast	mild	high	strong	**yes**
overcast	hot	normal	weak	**yes**
rainy	mild	high	strong	**no**
sunny	cool	high	strong	???

How to estimate the Probabilities?

- How to estimate the probabilities $\widehat{p}(\mathcal{C}_k)$ and $\widehat{p}(x_j|\mathcal{C}_k)$?
- **Solution**: Simply count the occurrences

$$\widehat{p}(\mathcal{C}_k) = \frac{\sum_{i=1}^{n} \mathbb{1}\{y^{(i)} = \mathcal{C}_k\}}{n} \tag{9}$$

$$\widehat{p}(x_j = v|\mathcal{C}_k) = \frac{\sum_{i=1}^{n} \mathbb{1}\{x_j^{(i)} = v \wedge y^{(i)} = \mathcal{C}_k\}}{\sum_{i=1}^{n} \mathbb{1}\{y^{(i)} = \mathcal{C}_k\}} \tag{10}$$

- $\mathbb{1}\{bool\}$ is the indicator function

  (returns 1, if *bool* is true, 0 otherwise. E. g.: $\mathbb{1}\{1+1=2\}=1$, $\mathbb{1}\{3=2\}=0$)

Let's compute some Probabilities

- New instance $\boldsymbol{x} = \langle sunny, cool, high, strong \rangle$
- What is its class?
- Let's compute some of the probabilities needed:

$$\begin{aligned}
\widehat{p}(Golf = yes) &= {}^{9}/_{14} = 0.64 \\
\widehat{p}(Golf = no) &= {}^{5}/_{14} = 0.36 \\
\widehat{p}(Outlook = sunny | Golf = yes) &= {}^{2}/_{9} = 0.22 \\
\widehat{p}(Outlook = sunny | Golf = no) &= {}^{3}/_{5} = 0.60 \\
&\ldots
\end{aligned}$$

Class Prediction

$$\widehat{p}(yes|\mathbf{x}) = \overbrace{\widehat{p}(sunny|yes)}^{=0.22} \cdot \overbrace{\widehat{p}(cool|yes) \cdot \widehat{p}(high|yes) \cdot \widehat{p}(strong|yes)}^{\text{calculate probabilities accordingly}} \cdot \overbrace{\widehat{p}(yes)}^{=0.64}$$
$$= 0.0053$$

$$\widehat{p}(no|\mathbf{x}) = \underbrace{\widehat{p}(sunny|no)}_{=0.60} \cdot \underbrace{\widehat{p}(cool|no) \cdot \widehat{p}(high|no) \cdot \widehat{p}(strong|no)}_{\text{calculate probabilities accordingly}} \cdot \underbrace{\widehat{p}(no)}_{=0.36}$$
$$= 0.0206$$

Classification: $\mathcal{C}_{MAP} = no$ (no golf today...)

Scaling the Output

- **But wait!** These probabilities don't sum up to one!?!?
  - This is because we dropped the normalization term $p(\boldsymbol{x})$
  - Scaling can fix this:

$$\widehat{p}(yes|\boldsymbol{x})_{norm} = \frac{0.0053}{0.0053 + 0.0206} = 0.205$$

$$\widehat{p}(no|\boldsymbol{x})_{norm} = \frac{0.0206}{0.0053 + 0.0206} = 0.795$$

- Scaling does **not** change the prediction

Laplace Smoothing

- **Problem:** A feature value $v^\star$ in the test data not seen during training
- $\widehat{p}(v^\star|\mathcal{C}_k) = 0$: The whole product becomes zero...
- **Solution**: Laplace smoothing

$$\widehat{p}(\mathcal{C}_k) = \frac{\sum_{i=1}^{n} \mathbb{1}\{y^{(i)} = \mathcal{C}_k\} + 1}{n + \kappa} \tag{11}$$

$$\widehat{p}(x_j = v|\mathcal{C}_k) = \frac{\sum_{i=1}^{n} \mathbb{1}\{x_j^{(i)} = v \wedge y^{(i)} = \mathcal{C}_k\} + 1}{\sum_{i=1}^{n} \mathbb{1}\{y^{(i)} = \mathcal{C}_k\} + \kappa} \tag{12}$$

Section:

Risk Minimization

DHBW
Duale Hochschule
Baden-Württemberg

Error $\neq$ Risk

- So far, we have tried to minimize the misclassification rate
- Nevertheless, there are cases where not every misclassification is equally bad
- Some classical examples:
  - **Smoke detector**
    - If there is a fire, we must make sure to detect it
    - If there is not, an occasional false alarm may be acceptable
  - **Medical diagnosis**
    - If the patient is sick, we have to detect the disease
    - If they are healthy, it can be okay to classify them as sick (order further tests)
- **Minimizing the error is not necessarily equal to minimizing the risk**

Loss Functions

- **Key idea:** We have to create a loss function which expresses what we want:

$$\text{loss}(\text{decision} = \text{healthy} \mid \text{patient} = \text{sick}) \gg$$
$$\text{loss}(\text{decision} = \text{sick} \mid \text{patient} = \text{healthy})$$

- We can decide for one of the κ possible classes...
- ...and we have a loss function $\ell(\mathcal{C}_i|\mathcal{C}_k)$ which returns the cost for deciding for $\mathcal{C}_i$ given $\mathcal{C}_k$ **(depends on the weighting of false positives and false negatives)**
- Expected loss (risk) of making a decision for class $\mathcal{C}_i$:

$$R(\mathcal{C}_i|\boldsymbol{x}) = \sum_k \ell(\mathcal{C}_i|\mathcal{C}_k)p(\mathcal{C}_k|\boldsymbol{x}) \tag{13}$$

Risk Minimization

- Consider two classes: $\mathcal{C}_1$ and $\mathcal{C}_2$
- Therefore, we have two possibilities: Deciding for class $\mathcal{C}_1$ or class $\mathcal{C}_2$
- Let ℓ_{ik} be a shorthand notation for $\ell(\mathcal{C}_i|\mathcal{C}_k)$
- Risk of both decisions:

$$
\begin{aligned}
R(\mathcal{C}_1|\boldsymbol{x}) &= \ell_{11} p(\mathcal{C}_1|\boldsymbol{x}) + \ell_{12} p(\mathcal{C}_2|\boldsymbol{x}) \\
R(\mathcal{C}_2|\boldsymbol{x}) &= \ell_{21} p(\mathcal{C}_1|\boldsymbol{x}) + \ell_{22} p(\mathcal{C}_2|\boldsymbol{x})
\end{aligned}
$$

- **Goal:** Create a decision rule so that the overall risk is minimized
- Decide for $\mathcal{C}_1$, iff $R(\mathcal{C}_2|\boldsymbol{x}) > R(\mathcal{C}_1|\boldsymbol{x})$

Risk Minimization (Ctd.)

$$R(\mathcal{C}_2|\mathbf{x}) > R(\mathcal{C}_1|\mathbf{x})$$
$$\ell_{21}p(\mathcal{C}_1|\mathbf{x}) + \ell_{22}p(\mathcal{C}_2|\mathbf{x}) > \ell_{11}p(\mathcal{C}_1|\mathbf{x}) + \ell_{12}p(\mathcal{C}_2|\mathbf{x})$$
$$(\ell_{21} - \ell_{11})p(\mathcal{C}_1|\mathbf{x}) > (\ell_{12} - \ell_{22})p(\mathcal{C}_2|\mathbf{x})$$

$$\frac{\ell_{21} - \ell_{11}}{\ell_{12} - \ell_{22}} > \frac{p(\mathcal{C}_2|\mathbf{x})}{p(\mathcal{C}_1|\mathbf{x})} = \frac{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}$$
$$\frac{p(\mathbf{x}|\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)} > \frac{\ell_{12} - \ell_{22}}{\ell_{21} - \ell_{11}} \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}$$

It is reasonable to assume that **the loss of a correct decision is smaller than that of a wrong decision**:

$$\ell_{ik} > \ell_{ii} \quad \forall k \neq i$$

Risk Minimization 0-1 Loss

$$\frac{p(\boldsymbol{x}|\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)} > \frac{\ell_{12} - \ell_{22}}{\ell_{21} - \ell_{11}} \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}$$

- **0-1 loss:** Decide for $\mathcal{C}_1$, if:

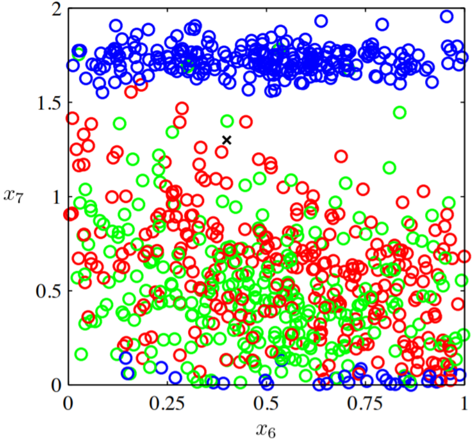
$$\frac{p(\boldsymbol{x}|\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)} > \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)} \qquad \text{with} \qquad \ell(\mathcal{C}_i|\mathcal{C}_k) = \begin{cases} 0 & i = k \\ 1 & i \neq k \end{cases} \tag{14}$$

- **0-1 loss leads to the same decision rule which minimizes the misclassification rate**

Are we done?

- **Question:** Are we done with classification?
  - We have decision rules for simple and general loss functions
  - They are even **Bayes' optimal**
  - We can deal with two or more classes
  - We can deal with high dimensional feature vectors
  - We can incorporate prior knowledge about the class distribution
- We have seen how to get the probabilities for the discrete case (cf. naïve Bayes classifier)
- **But: What about continuous data?**

Continuous Data



Section:

Wrap-Up

DHBW
Duale Hochschule
Baden-Württemberg

Summary

- Statistical methods assume that the process that 'generates' the data is **governed by the rules of probability**
- We need class **conditional probabilities** and **class priors**
- Use **Bayes' theorem** to get the **class posteriors**
- **Bayes' optimal classifier:** Decide for the most probable class
- Naïve Bayes assumes all **features to be pairwise conditionally independent**
- **Error minimization is not equal to risk minimization**

Important

Self-Test Questions

1. What are class conditional probabilities?
2. What does *Bayes optimal* mean?
3. How can we incorporate prior knowledge about the class distribution into the classification?
4. What is the naïve assumption which naïve Bayes makes? When is this a problem?
5. Explain what maximum aposteriori is!
6. What is misclassification and risk? Are they the same?

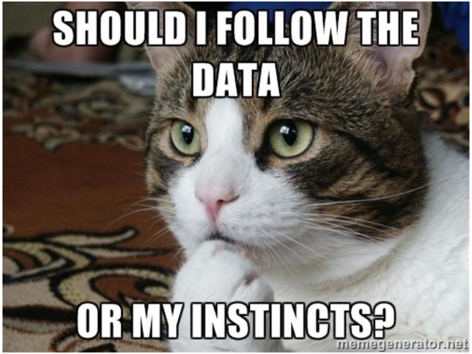
What's next...?

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	Bayesian Decision Theory
Unit IV	**Probability Density Estimation**
Unit V	Regression
Unit VI	Classification I
Unit VII	Evaluation
Unit VIII	Classification II
Unit IX	Clustering
Unit X	Dimensionality Reduction

Recommended Literature and further Reading I

[1] Pattern Recognition and Machine Learning
Christopher Bishop. Springer. 2006.
→ Link, cf. chapter 1.5

Meme of the Day



Thank you very much for the attention!

Topic: *** Applied Machine Learning Fundamentals *** Bayesian Decision Theory
Term: Winter term 2020/2021

Contact:

Daniel Wehner, M.Sc.
SAP SE / DHBW Mannheim
daniel.wehner@sap.com

Do you have any questions?