Institute for Application-Oriented Knowledge Processing

Student Topics in Computational Data Analytics

JOHANNES KEPLER UNIVERSITÄT LINZ

https://www.jku.at/en/faw/

Our Faculty

Data – Information – Knowledge

Johannes Fürnkranz

- Computational Data Analytics
- Data Mining and Knowledge Discovery
- Rule Learning and Interpretability
- Machine Learning in Games
- Preference Learning and Multi-Label Classification

Josef Küng

- Knowledge-based Systems and Knowledge Representation
- Security and Trust in Information Systems
- Process-aware Information Systems
- Similarity Queries

Birgit Pröll

- Information Retrieval & Extraction
- Natural Language Processing
- Web Search and Mining
- Web Engineering und Web Science

Wolfram Wöß

- Information Integration (Semantic-based, Ontologies)
- Data Modeling
- Knowledge Representation and Knowledge Graphs
- Data Quality, Data Profiling, Data Mining

Student Topics

- We offer seminar/project/thesis topics in Computational Data Analysis
- In the following, you can find sample topics in a few areas
  - Machine Learning in Games
  - Interpretability and Inductive Rule Learning
  - Multi-label Classification
- If you are interested in similar problems, you can also propose your own topic
- Prerequisites
  - Some basic knowledge (and ideally practical experience) in machine learning and data mining is assumed

Seminar / Project / Thesis

There are two possible paths

- You compile seminar / project / thesis (or two of the three) into a single package, typically
  - start with giving a presentation (seminar)
  - implement or work with state-of-the-art techniques (project)
  - investigate a new interesting question (thesis)
- You do all of them separately
  - availability depends on the amount of interest
  - seminar: several talks by different students on an over-arching topic
  - project: group work on some problem (often a competition)

You can switch between the two models inbetween

1. Game Playing

- We are generally interested in using AI technology for game playing
    - typically conventional board or card games, but dynamic video games are also possible
- Topics could involve questions such as
    - design and implement a strong player for a new game
    - learn a player from human game playing databases or from self play
    - analyze human decisions in game databases
    - gain knowledge about the game by analyzing game databases
    - ….
- Some example projects are on the following slides
    - If you have an interesting game project, feel free to talk to us

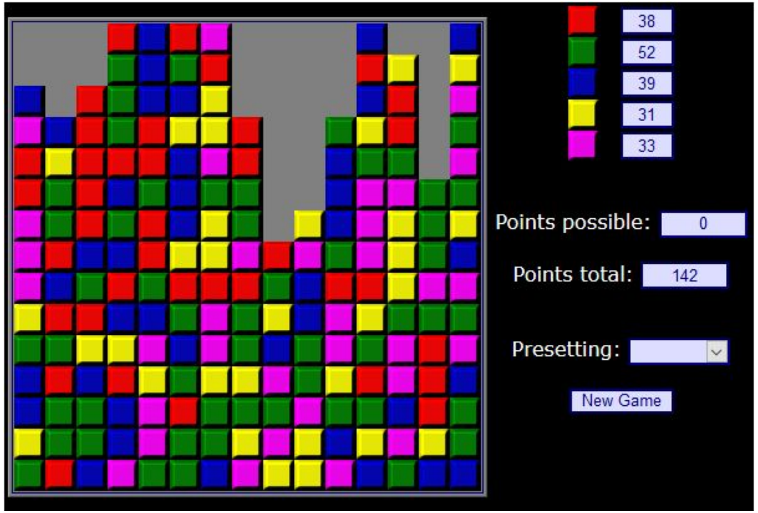
Innovative Player for Same Game

Tasks:

- Research and Present SOA Players based on MCTS
- Implement and compare some of them
- Develop a new solution
- **Optional:** Beat the high score :-)

TAKEN

http://www.js-games.de/eng/games/samegame



🟥	38
🟩	52
🟦	39
🟨	31
🟪	33

Points possible: 0

Points total: 142

Presetting:

Preference-Based GNRPA

- (General) Nested Rollout Policy Adaptation is a general learning algorithm similar to Monte-Carlo Tree Search
    - the idea is to perform several roll-outs and to repeatedly adjust the weights in the roll-out policy towards the maximum policy
    - used in many puzzle games, such as SameGame
- the current implementation makes use of numerical scores
- can it also work with only preference-based feedback?
    - -> also SameGame thesis

http://www.js-games.de/eng/games/samegame

Discovering Interpretable Strategies for Chess Endgames

- Chess endgame databases provide tables that Contain optimal moves for certain endgames
- Human playes, on the other hand, follow simple strategies

Tasks:

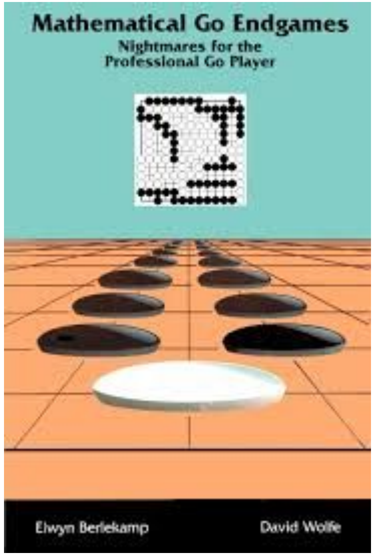
- Take a simple database like KRK (or later the more interesting KQKR)
- Try to work out ways for learning a human-like (i.e., simple, but possibly suboptimal) policy from the databases

Evaluation of Computer Endgame Play in NN-based Go Programs

- Computer Go engines such as AlphaZero or its public variant Leela Zero make mistakes
- Endgame databases play perfectly

- Possible tasks:
  - Investigate and quantify the amount of mistakes these progams make
  - Try to characterize the type of mistakes they make
  - Compare mistakes in the evaluation vs. search

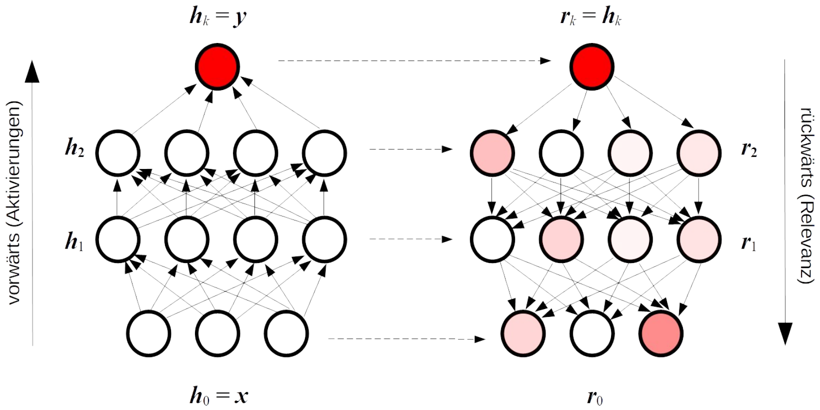
- Together with Martin Müller, an expert in Go & Computer Go



http://zero.sjeng.org/

Interpretation of NN-based Go or Chess Programs

- There are publicly available neural networks for Leela Go and others
- There are several recent techniques for making deep learning interpretable
  - LIME, SHAP, LRP



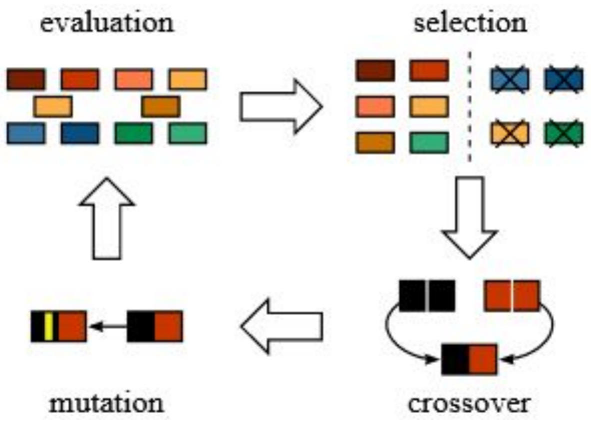
- Tasks:
  - Try these methods on trained Go or chess networks
  - See whether they help to interpret the learned knowledge

- Possibly together with Martin Müller, an expert in Go & Computer Go

http://zero.sjeng.org/

MCTS as a General Optimization Technique

- Monte-Carlo Tree Search (MCTS) is designed to find a sequence of actions to reach a goal
- In many problems, not the path is of interest, but the final result
  - E.g., in many puzzles or hard problems
- Classical solutions methods for such problems include local search algorithms such as genetic algorithms
- Can (single-agent) MCTS be an alternative?
  - There are various variants that operate on sets instead of sequences
  - These need to be researched, evaluated, compared, improved.



FAW

Bertram, Fürnkranz

Deep Reinforcement Learning for Reconnaissance Blind Chess

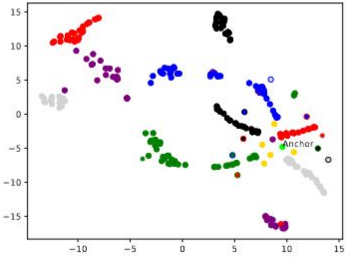
- RBC is a chess-variant in which you are not aware of the moves of your opponent (https://rbc.jhuapl.edu/)
- Each turn you are allowed to look at a 3x3 square
- Game playing algorithms are very good at solving perfect information games but still struggle in scenarios with imperfect information
- Tasks:
  - Research RL-algorithms for imperfect information games like Neural Fictitious Self Play
  - Implement your own agent
  - Test your agent against baselines and on the leaderboard

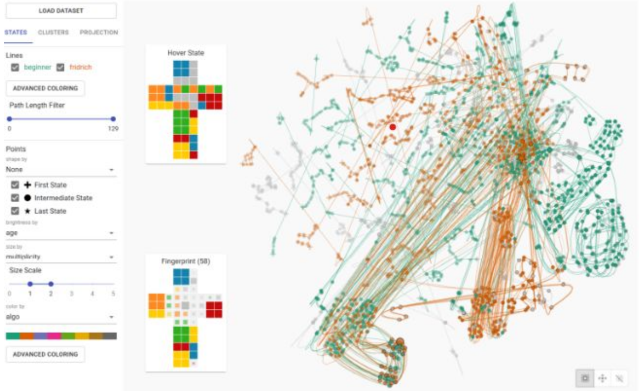
Predicting Win-rates of Hearthstone decks

- In the ‘18 AAIA competition, participants were tasked to predict the win-rates of Hearthstone decks (https://knowledgepit.ai/predicting-winrates-of-hearthstone-decks/)
- Training and testing files are fully available, so algorithms can still be constructed informally for this competition
- Can you improve upon the existing algorithms?
- Tasks:
    - Review the submissions for the competition
    - Develop your own strategy
    - Test and evaluate your algorithm against the leaderboard

Interpretation of “Magic: The Gathering” Card Drafts

- we have developed a method that is able to predict human card selection in the collectible card game *Magic: The Gathering*
- the method is based on a learned embedding of the cards in a high-dimensional space
- we want to analyze and interpret this card embedding and the dynamics of human card selection
- Approach:
  - Combine our embeddings with the ProjectionPathExplorer of the Visual Data Science Lab (Marc Streit)
  - jointly supervised by them





Lines

☑ beginner ☑ fridrich

Path Length Filter: 0 – 129

Points

shape by: None

☑ ✚ First State

☑ ● Intermediate State

☑ ★ Last State

brightness by: age

size by: multiplicity

Size Scale: 1 – 2 (0 1 2 3 4 5)

color by: algo

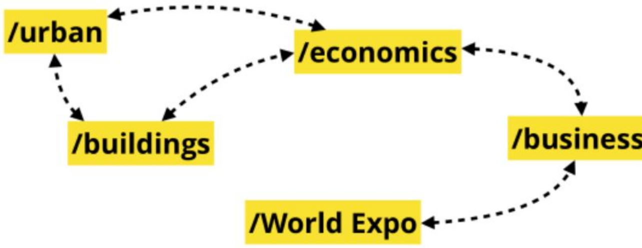
2. Multi-Label Classification and Preference Learning

- **Multi-class Classification** is the task of learning a function that is able to assign a **single label** to a given example
- **Multi-label Classification** is the task of learning a function that is able to assign a **set of labels** to a given example
  - *Example task*: Assign a subset from a set of pre-defined keywords to a document
- **Preference Learning** ist the task of learning to **order a set of labels**
  - *(Hypothetical) Example task*: Given the characteristics of persons, learn to rank political parties in the person's order of preference
- We have been working on several aspects of the problems, and have some thesis topic that deal with these types of problems

Multi-Label Prediction Corrector

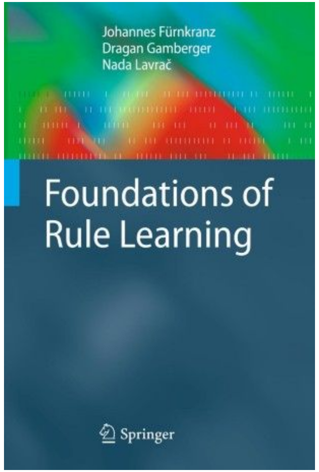
- Multi-label classification is the task of predicting not a single but a set of labels
  - e.g., assigning keywords to a document
- Task:
  - define a post-processor for correcting predictions of arbitrary multi-label classifiers so that they respect some data-driven constraints such as
    - only predict previously observed label sets (correct to the closest if not)
    - respect label dependencies previously observed in the data (e.g., using an association rule learner)
    - possibly minimize a loss function in the neighborhood
    - ….
  - the approach should be implemented (best in Java) and tested with several multi-label classifiers on a variety of publicly available datasets

Studying Multi-Label Dependencies



- one of the challenges in multi-label classification is that labels depend on each other
  - e.g., if an image contains the sun, it will probably also contain blue sky
- nevertheless, many of the common benchmark datasets do not seem to have a strong dependency structure
  - some of them are even, essentially, single-label
- Task:
  - survey the literature for studies on how to measure label dependencies
  - investigate existing benchmark datasets for this quality
  - generate artificial datasets with known dependency structures
    - building upon a known dataset generator by Dembczynski et al.

3. Inductive Rule Learning


Johannes Fürnkranz
Dragan Gamberger
Nada Lavrač

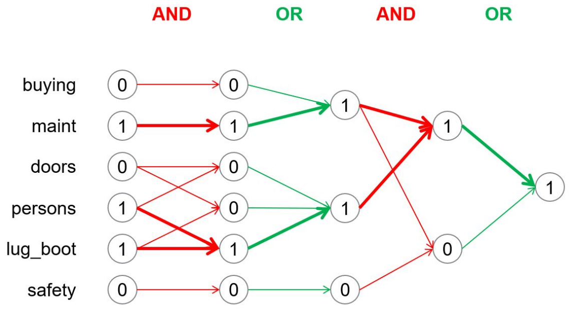
Foundations of Rule Learning

Springer

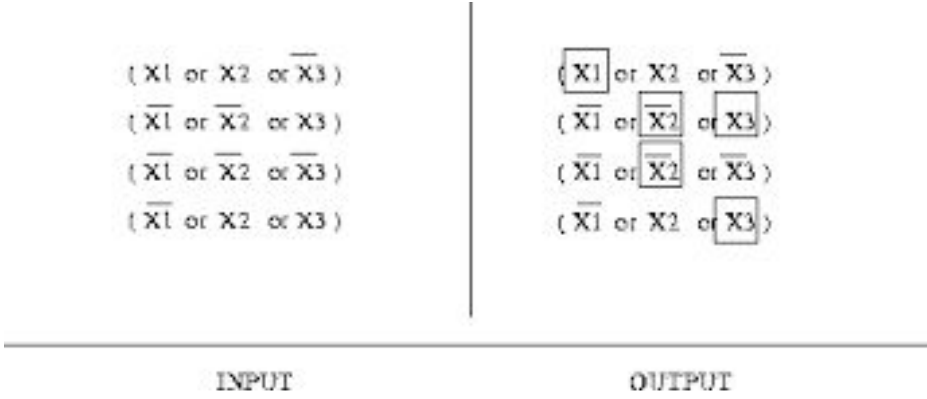
- Logical IF-Then rules are commonly used in many applications
  - Are very interpretable but often not very accurate
- Many topics that can be worked on, such as
  - **Combining Rule Predictions**
    - Many approaches have been proposed on how to combine the predictions of multiple rules (e.g., in an ensemble) -> compare them!
  - **Interpreting Multi-Label Predictions**
    - Current works on interpretable machine learning focus on single-label predictions tasks -> methods for explaining joint predictions are needed
  - **Comparison of Extracted Explanations vs. Learned Models**
    - Do local rules extracted from a deep network outperform rules that are directly learned from the data?

Improve Rule Networks

- Rule networks are neural networks with Boolean weights and activations
  - Each node can be interpreted as a rule
- Basic scikit-learn implementation developed in our group is provided
- Task:
  - Survey the literature on binary/ternary neural networks
    - How to adapt these approaches to the Boolean case?
  - Improve the existing implementation
    - Initialization of rule networks
    - Weight flipping algorithm
    - Multi-class classification
    - … (open for suggestions)



SAT for Inductive Rule Learning


$(X1 \text{ or } X2 \text{ or } \overline{X3})$
$(\overline{X1} \text{ or } \overline{X2} \text{ or } X3)$
$(\overline{X1} \text{ or } \overline{X2} \text{ or } \overline{X3})$
$(\overline{X1} \text{ or } X2 \text{ or } X3)$
$(\boxed{X1} \text{ or } X2 \text{ or } \overline{X3})$
$(\overline{X1} \text{ or } \boxed{\overline{X2}} \text{ or } \boxed{X3})$
$(\overline{X1} \text{ or } \boxed{\overline{X2}} \text{ or } \overline{X3})$
$(\overline{X1} \text{ or } X2 \text{ or } \boxed{X3})$
INPUT
OUTPUT

- Several recent works have investigated the use of satisfiability solvers for decision tree learning and rule learning
- Task:
  - survey the field
  - implement and compare different approaches to each other and to SOA machine learning algorithms
  - investigate potential for noise handling
  - possibly develop a new improved approach
  - possibly also for alternative representations, such as decision graphs

Learning Internal Disjunctions

- Internal disjunctions are sets of values for a condition
  - e.g., `weather = rainy || cloudy`
- Only a few learners can **explicitly** learn such rules
  - e.g., AQ or Bexa
- However, most rule learner can implicitly learn equivalent conditions as conjunctions of negations
  - e.g., `weather != sunny & weather != snowfall`
- Task:
  - Evaluate how rule learners work when *only* internal negated conditions are allowed
  - Possibly compare this to extension that explicitly handle internal disjunctions

Evaluate Closed Classification Rules

- Classification rule learners typically strive for learning simple discriminative rules
  - “If you see an animal with a trunk, it is an elephant.”
- From an interpretability point of view (and maybe also for robustness) it may be preferable to learn more complex characteristic rules
  - “An Elephant is a very large and heavy animal that has a trunk, tusks, grey skin, big ears and four thick legs.”
- Task:
  - survey the literature on characteristic rule learning
  - implement an algorithm that takes discriminative rules as input and outputs the closure
  - devise a method for “approximate” closures, possibly also with a bias toward interesting conditions
  - evaluate it on various benchmark data w.r.t. accuracy, robustness, and interpretability

Improve JRip

- Ripper is a classic rule learning algorithm that is still very hard to beat in terms of efficiency as well as compactness of the learned rules
- the currently best implementation available is **JRIP**
  - available in the Weka data mining environment (implemented in Java, https://www.cs.waikato.ac.nz/ml/weka/)
  - code of the original C++ implementation is also available
- However JRip could be made for flexible in various ways, such as
  - allow the use of different search heuristics
  - allow for different class orders, both static as well as dynamically selected
  - adjust for the use of mini-batches instead of train/test splits
  - analyze the rule optimization phase and possible find a cheaper alternative
  - interface with Python/scikit-learn
- excellent Java programming skills required

Local Optimal Rule Learning

- Local optimal rule learning behaves well on single-label classification and execution performance but the output rule set is often much larger than that by other approaches.
- Tasks: at least one of the following two tasks should be done
  - Quest for a technique to reduce the output rule set while keeping the classification performance not worse.
  - Improve classification performance: from the set of covering rules for a new example, develop a method(s) to select a rule for classifying the new example.

Rule Set Ensemble Learning

- Random Forest, an ensemble of decision trees, have been well known for its high classification accuracy. There are also available rule learners which basically generate rule set ensemble from decision trees.
- Tasks: Develop a method for learning an ensemble of rule sets directly from data while keeping in mind three following criteria.
  - As simple rule sets as possible
  - Comparative classification performance
  - Efficiency so that it is applicable for big data

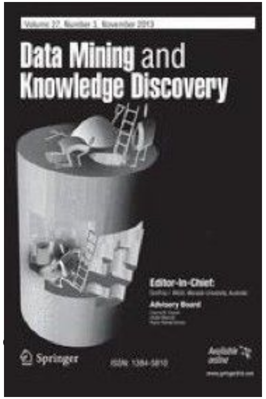
Distributed Rule Learning

- Dealing with big and very big data sets is a crucial problem (running time and/or consumed memory) to the state-of-the-art rule learners.
- Tasks: Develop a parallel rule learning method for handling well very large-scale data sets on distributed environments e.g. clusters, local networks while keeping in mind the following criteria.
  - Remain or keep as same as possible with the rule set by the corresponding serial version.
  - Maintain high speed up when the number of computational nodes becomes larger.

4. Other Topics

- We are also open to suggestions for other topics
  - typically they have to do with machine learning and/or knowledge discovery in databases
- Sometimes we also have other topics

Review Matcher for DAMI



- **Data Mining and Knowledge Discovery** is a peer-reviewed international journal
  - This means that articles that are submitted for publications by rese will be reviewed by other researchers
- **Editorial Board:**
  - a finite set of researchers that do many of these reviews
- **Problem:**
  - submitted paper need to be matched to competent reviewers
- **Task:**
  - implement a system that is able to match a short text (e.g. the abstract of a paper) to a reviewer (who can be defined with a set of keywords)

Citation Normalizer

- In (badly formatted) scientific papers, one often finds incomplete or abbreviated references, such as

  [1] Fürnkranz et al: On cognitive preferences and the plausibility of rule-based models, MLJ, 2020

- Devise a system that takes such incomplete (and possibly incorrect) references, and returns a complete citation and a full BibTex Entry (using DBLP as a source)

  [1] Johannes Fürnkranz, Tomás Kliegr, Heiko Paulheim: On cognitive preferences and the plausibility of rule-based models. Machine Learning 109(4): 853-898 (2020)

```
@article{FurnkranzKP20,
  author    = {Johannes F{\"{u}}rnkranz and
               Tom{\'{a}}s Kliegr and
               Heiko Paulheim},
  title     = {On cognitive preferences and the plausibility of
rule-based models},
  journal   = {Machine Learning},
  volume    = {109},
  number    = {4},
  pages     = {853--898},
  year      = {2020},
  url       = {https://doi.org/10.1007/s10994-019-05856-5},
  doi       = {10.1007/s10994-019-05856-5}
}
```

SS Project Only: Data Mining Cup

- The Data Mining Cup is an annual international Data Mining Competition for students
  - Students get a dataset
  - and a test set without labels
  - need to make predictions for the test set
  - possibly minimizing some cost function
- Project:
  - March: Trial period (analyzing a past dataset)
  - mid April: focusing on the new dataset
  - end June: predictions are due
  - report on the way the submission was derived
- Group work possible

DATA MINING CUP
International Student Competition

https://www.data-mining-cup.com/

Distributed Frequent Itemset Mining

- Frequent Itemset Mining (FIM) is a fundamental mining technique in Data Mining. It can be employed as a key calculation phase in other mining models such as Association Rules, Inductive rules, Classifications, Text Mining, etc.
- In the current era of Big Data, the distributed FIM algorithms have been proposed to dealing with very-large scale data sets. But the effort to improve the execution performance is always necessary because a distributed FIM algorithm likely suffers a kind(s) of adverse data sets.
- Tasks:
  - Quest for a distributed FIM algorithm that can perform stablly on diversity kinds of data sets and higher efficiency.

Optimized Implementation for FUSINTER Discretization

- FUSINTER is an effective method to convert numeric data to discrete one which are then consumed by many machine learning techniques.
- Tasks:
  - Implement FUSINTER method by Python in an optimized way for better efficiency.
  - Think of modifications from the basic technique to improve the execution performance while keeping as high as possible the prediction accuracy of a consumer, e.g. target rule learner.

Data Quality Topics

- Trust in AI-based decisions requires (training) **data of high quality**
- Bias in data needs to be detected and avoided
- We developed **DQ-MeeRKat**: a tool for automated data quality measurement: https://github.com/lisehr/dq-meerkat
- Possible master thesis topics:
  - Extend DQ-MeeRKat with multivariate outlier detection models
  - Extend DQ-MeeRKat with ML models for duplicate detection
  - Measuring the Quality of Knowledge Graphs
  - … (suggest own topic on data quality)

Contact
Lisa Ehrlinger
Science Park 3 - Stockwerk 3 - Raum 338
+43 732 2468 4195
lisa.ehrlinger@jku.at

https://www.jku.at/en/faw/teaching/bachelor-masters-theses/