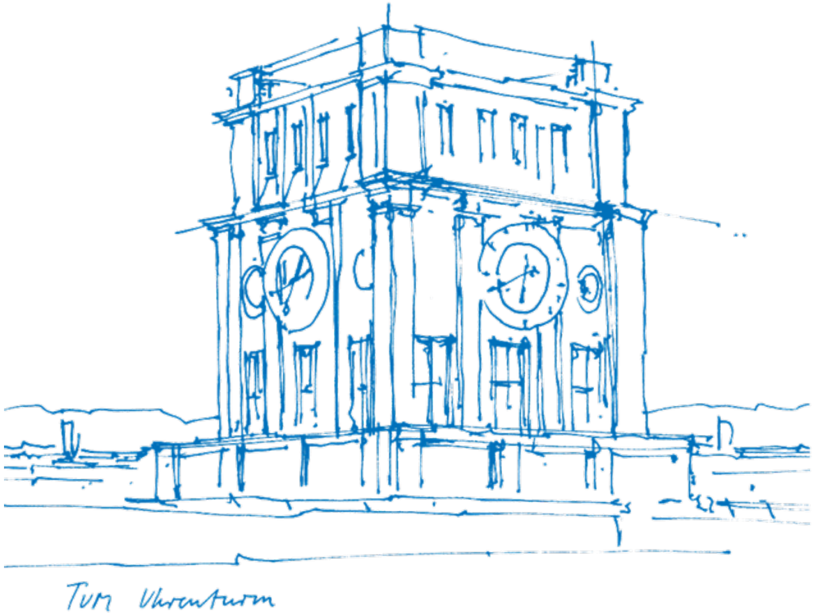
Machine Learning in Crowd Modeling & Simulation Lecture 1 - Modeling crowd dynamics

Felix Dietrich

2023-10-19 (organizational part)



Instructor

Machine Learning in Crowd Modeling & Simulation

Dr. Felix Dietrich

starting 2022	research group leader “Harmonic AI” @TUM
2019-2022	postdoctoral fellow @TUM
2017-2019	postdoctoral fellow @PU/JHU
2014-2017	PhD @TUM

Research interests:

- Data science for complex dynamical systems
- Algorithms for high-dimensional data
- Approximation of linear operators on manifolds
- Bridging scientific computing and machine learning

Doctoral Thesis: *Data-driven surrogate models for dynamical systems*
Website: `www.fd-research.com/`

Hybrid mode

The entire course will be held in hybrid mode: full online participation is possible, with recorded, live-streamed, on-site QnA sessions. I recommend to attend on-site.

All recorded videos are linked on the Moodle page:

`https://www.moodle.tum.de/course/view.php?id=90636`

The chat can be reached here:

`https://zulip.in.tum.de/#narrow/stream/1239-MLCMS-winter23.2F24`

It is a private stream, so let me know your zulip user name: `felix.dietrich@tum.de` or PM on zulip (User ID 3316, “Felix Dietrich”) to add you.

Outline

This video: Lab course organization

- Topics covered in the course
- Software and course requirements
- Organizational issues (exercise sheets, reports, ...)

Content video: Modeling crowds (exercise due date: 2023-11-02)

- General overview
- Modeling approaches
- In detail: Cellular automata

Lab course: ML in Crowd M&S

Lecture 1: Modeling crowd dynamics

- Modeling approaches, verification and validation

Lecture 2: Simulation software

- Introduction to the Vadere software, SIR models

Lecture 3: Representation of data

- Principal Component Analysis, Diffusion Maps, neural networks

Lecture 4: Dynamical systems and bifurcation theory

- Introduction to the theory and examples

Lecture 5: Extracting dynamical systems from data

- Function approximation, vector fields, time-delay embedding

Lecture 6: Future directions of machine learning

- Challenges in data science, master's thesis topics, final projects

Lab course: ML in Crowd M&S

What will I know after this course?

1. State of the art in mathematical modeling of crowds, validation of models to data
2. Implementation of simulation software, visualization of results
3. Machine learning techniques for the numerical analysis of complex dynamical systems:
   3.1 Bifurcation analysis
   3.2 Representation learning (manifold learning)
   3.3 Extraction of predictive dynamical systems from observation data

Requirements

1. Familiarity in either Java or Python (C++ is also possible but not used in examples)
2. Basic concepts of linear algebra (matrix/vector computations, eigendecomposition)
3. Basic knowledge of dynamical systems (ordinary differential equations)
4. Being able to work together in a small group

Lab course: ML in Crowd M&S

Software and programming

For software (visualization, version control, python environment, etc.), I recommend you take a look at the list of resources to learn programming on my website [1].

Good ideas:

- Use an IDE (IntelliJ, eclipse, Visual Studio Code, PyCharm, ...)
- Use Jupyter notebooks to visualize your results, write the actual code in separate code files.
- Document your code where necessary [2], structure your repository and files into general methods, visualization, tests, and data.
- Do not overdo it—one notebook and one file with methods may already be enough for most exercises.
- Watch the coding video on Moodle for “Exercise 0”!

---

1 https://www.fd-research.com

2 https://towardsdatascience.com/5-incredibly-useful-tips-for-writing-pristine-documentation-803155ae4f45

Lab course: ML in Crowd M&S

Software and programming

You can get a lot of points for proper documentation, code clarity and modularity.
Do not underestimate the value of good code!

```
import numpy as np

from skimage.transform import resize
import scipy.misc
import matplotlib.pyplot as plt

values = np.loadtxt(open("pca_dataset.txt", "rb"), delimiter=" ", skiprows=0)


x = values[:,0]
y = values[:,1]

rows = len(values)
columns = len(values[0])


M = np.zeros([rows,columns])
M[:,0] = values[:,0] - values[:,0].mean()
M[:,1] = values[:,1] - values[:,1].mean()
```

NO!

Figure: Bad python code: poorly documented, no separation into methods, classes, etc.—DO NOT DO THIS

Lab course: ML in Crowd M&S

Software and programming

You can get a lot of points for proper documentation, code clarity and modularity.
Do not underestimate the value of good code!

```
# -*- coding: utf-8 -*-
"""
Created on [redacted]

@author: [redacted]
"""

import matplotlib.pyplot as plt
from tkinter import *
import numpy as np
import tkinter.font as tkFont
import tkinter as tk
import time

def randomly_initialize(n, pedestrian_number,obstacle_number) :

    """ Initialize the position of the target, the pedestrians and the obstacles
    RANDOMLY
    env is the matrix of non-moving objects, X the current location of pedestrianc
     Code : (for env) 0 = empty cell, 2 = obstacle, 3 = Target """

    #grid
    grid_side = int(np.sqrt(n))

    ###################non moving objects
    env = np.zeros((grid_side,grid_side)) #obstacles and target : in a matrix
```

YES!

Figure: Good python code: properly documented, docstrings, separation into methods, etc.

Lab course: ML in Crowd M&S

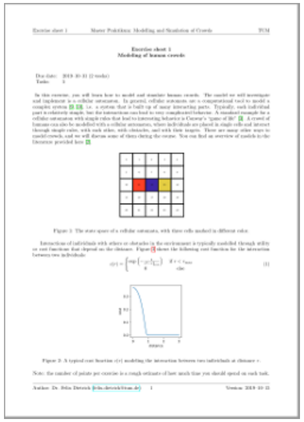
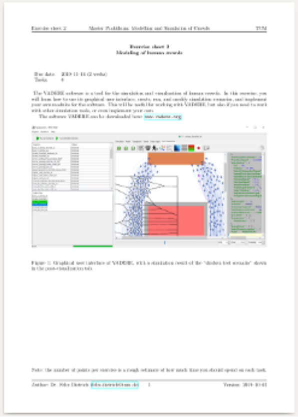
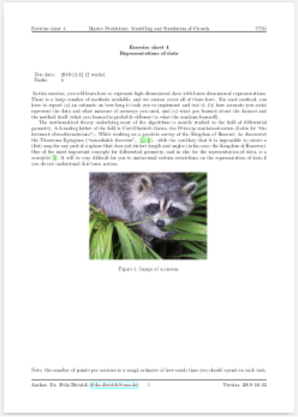
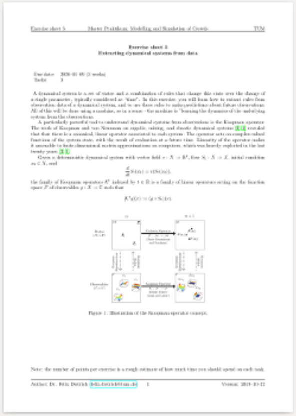
Organization

1. 10 ECTS course
2. Language of instruction: English (exercises, slides)
3. 5 exercise sheets (implementing a model, visualization, numerical analysis, surrogate models, ...)
4. 1 final project (propose or choose a topic in the field of machine learning for crowd data)
5. Lectures and new exercises every 2 weeks (see schedule in TUMOnline and Moodle)
6. The grade for each exercise sheet is an averaged grade from the code submission (50%) and the discussion of the results (written reports and in class, 50%). The first 5 sheets will have the same weight factor of 15% each, the final project (sheet # 6) 25%
7. Continuous participation in the lectures is strongly advised. There is only one lecture per exercise, and I will introduce new concepts only once.
8. All exercises and final projects in groups of five (use “group selection” in Moodle!)

Lab course: exercises

Overview

1. About five tasks in each exercise
2. Five exercises in total, plus one final project

Modeling	Software	Bifurcations	Representation	Dynamics
				
Due: 2023-11-02	2023-11-16	2023-11-30	2023-12-21	2024-01-18

+ final project (25% of all points, due 2023-02-08)

Lab course: exercises

Grading of exercises

1. Max. 100 points per exercise, you can fill up lost points with bonus points
2. Every group member gets individual points, based on the “division of labor” you write in the reports
3. The number of points per group member is multiplied by the number of group members for the individual score (up to a maximum of 100 points!)
4. **New this semester:** you must specify who is the “project leader” in each report. They get 20 individual points in addition to the report+code. Everyone can be a project leader only once.

Grading example

The work on tasks was divided in the following way:

FirstnameA LastnameA (1234567890)	Task 1	100%
	Task 2	100%
	Task 3	50%
	Task 4	50%
	Task 5	0%
FirstnameB LastnameB (1234567891) **Project lead**	Task 1	0%
	Task 2	0%
	Task 3	50%
	Task 4	50%
	Task 5	100%

Assume you get these points for tasks 1-5: 20/20, 10/20, 20/20, 10/20, 10/20
A: 20*100% + 10*100% + 20*50%+10*50%+10*0% = 20+10+10+5+0 = 45; Then: 45*2 (# members)=90.

B: 20*0% + 10*0% + 20*50%+10*50%+10*100% = 0+0+10+5+10 = 25; Then: 25*2 (# members) +20 (team lead)=70.

Lab course: exercises

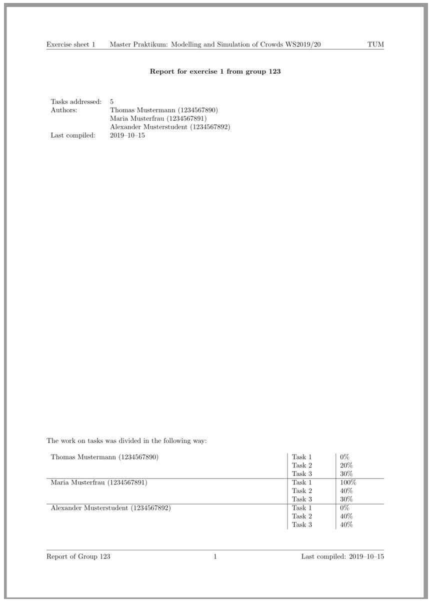
Handing in your reports

1. Use the LaTeX template on Moodle for your report (10-15 pages, including everything but code)
2. Include who contributed how much on each task (in percentage points for the task)
3. Add a link to the repository of your code.
    3.1 Make the repository accessible to me. My usernames are on the Moodle page.
    3.2 **Only use one repository for the entire course.**
    3.3 **Do not upload (large) data or reports! No 1GB+ repositories!**
    3.4 For collaboration with data, use LRZ sync & share (for example) and link to it. Even better: Git LFS (https://docs.gitlab.com/ee/topics/git/lfs/)
4. Before the start of the next lecture, upload the report to Moodle (about two weeks for each report)
5. If you have never worked with LaTeX, start now:
    5.1 https://www.latex-tutorial.com/tutorials/ (English)
    5.2 https://latex.tugraz.at/latex/tutorial (German)
6. Tip: overleaf.com is an easy way to collaborate on a LaTeX document
7. **Important:** do not upload (a) papers/books, or (b) your reports to a public repository! You will get (a) sued for copyright infringement and (b) your identity will get stolen (because the reports contain matriculation numbers). Just make the repository private.

Lab course: exercises

Example

An (incomplete) example report for the first exercise sheet
You can find the template and the example on Moodle


Exercise sheet 1 Master Praktikum: Modelling and Simulation of Crowds WS2019/20 TUM

Report for exercise 1 from group 123

Tasks addressed:	5
Authors:	Thomas Mustermann (1234567890)
	Maria Musterfrau (1234567891)
	Alexander Musterstudent (1234567892)
Last compiled:	2019–10–15

The work on tasks was divided in the following way:

Thomas Mustermann (1234567890)	Task 1	0%
	Task 2	20%
	Task 3	30%
Maria Musterfrau (1234567891)	Task 1	100%
	Task 2	40%
	Task 3	30%
Alexander Musterstudent (1234567892)	Task 1	0%
	Task 2	40%
	Task 3	40%

Report of Group 123 1 Last compiled: 2019–10–15

Lab course: exercise reports

Handing in your reports

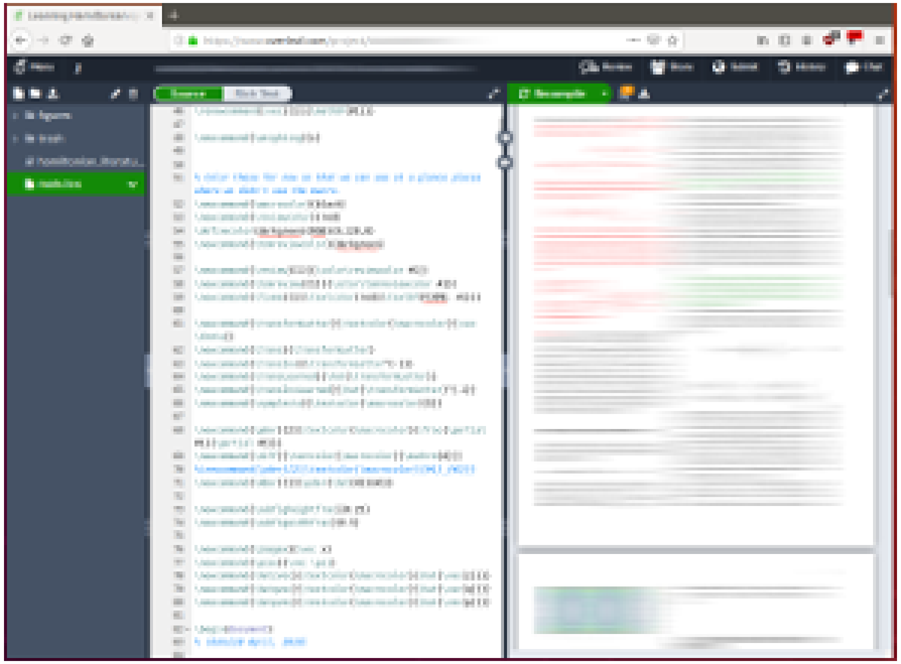


Figure: Interface of a report on overleaf.com. You do not need to install anything, it all works in the browser.

Lab course: exercise reports

Guidelines for good reporting + coding

1. Report only on tasks that you worked on.
2. Include the workload per team member.
3. Include all parameters and test results in the report, tables are ideal for this.
4. You must write a **minimum of three pages per group member in each report.**
5. Every missing page reduces the number of total points for the exercise by how much is missing (i.e., for a group of five: by $1/15 \approx 7$ points).
6. Code will be judged on clarity, modularity, and documentation.
7. Hint for python: Use Jupyter notebooks with markdown to display results and figures, put classes and methods in separate files.
8. Hint for python: Document your code where necessary, use `sklearn` or `scipy` interfaces where appropriate (see for example the RBF class [1]).

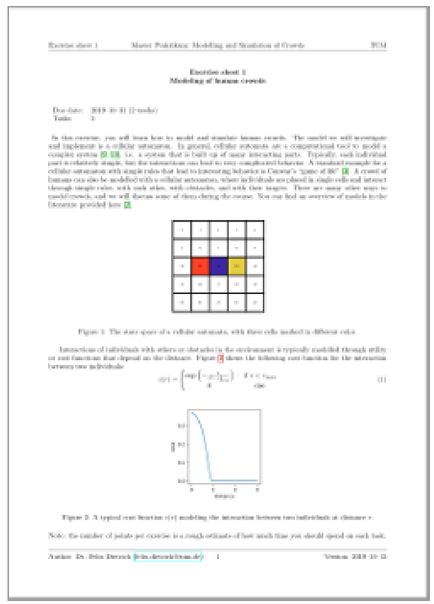
1 `https://github.com/scipy/scipy/blob/v1.4.1/scipy/interpolate/rbf.py#L59-L290`

Modeling crowds

First exercise: modeling a crowd in a cellular automaton

The first exercise sheet

You can find the complete exercise sheet on Moodle



Highlights

Exercise 1

- Many different simulators, languages, GUIs, ...
- Nobody implemented pedestrians walking the same speed in different directions
- The Dijkstra algorithm was a challenge for many groups (diagonal cells have a different distance!)

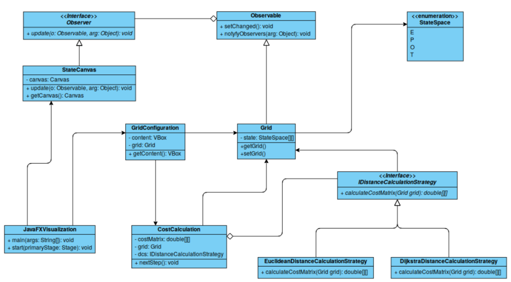
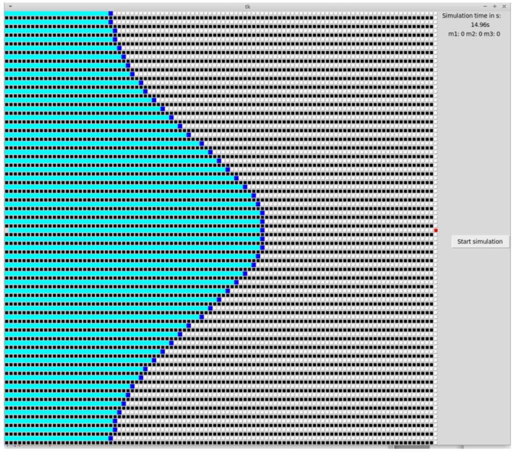


Figure 1: UML diagram of the project.



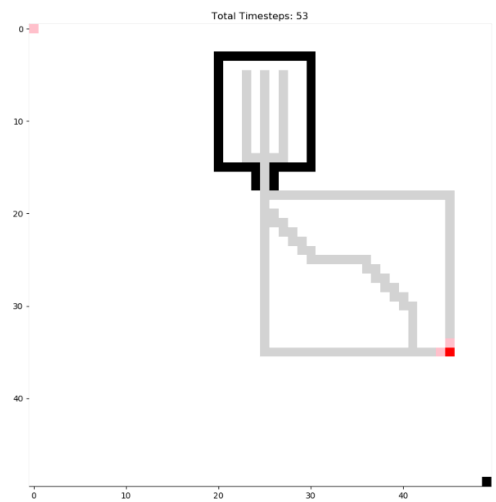
```
Enter number of targets: 1

Enter size of target 1: 0
Enter target 1 location x: 2
Enter target 1 location y: 3
- - - - -
- 1 - - -
- - - T -
- - - - -
- - O - -
```

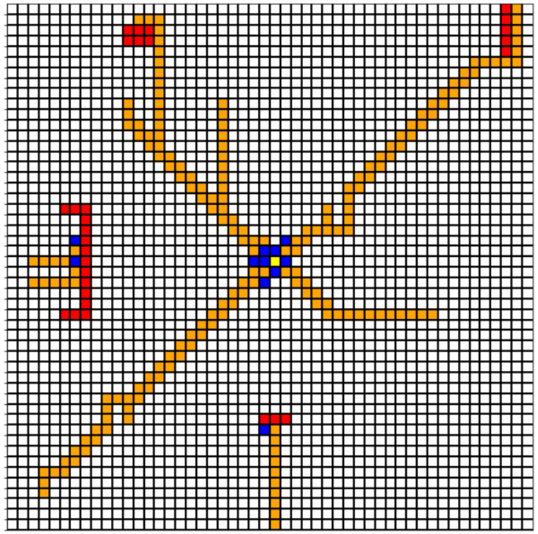
Figure 4: Grid with one target

Highlights

Exercise 1

- Many different simulators, languages, GUIs, ...
- Nobody implemented pedestrians walking the same speed in different directions
- The Dijkstra algorithm was a challenge for many groups (diagonal cells have a different distance!)



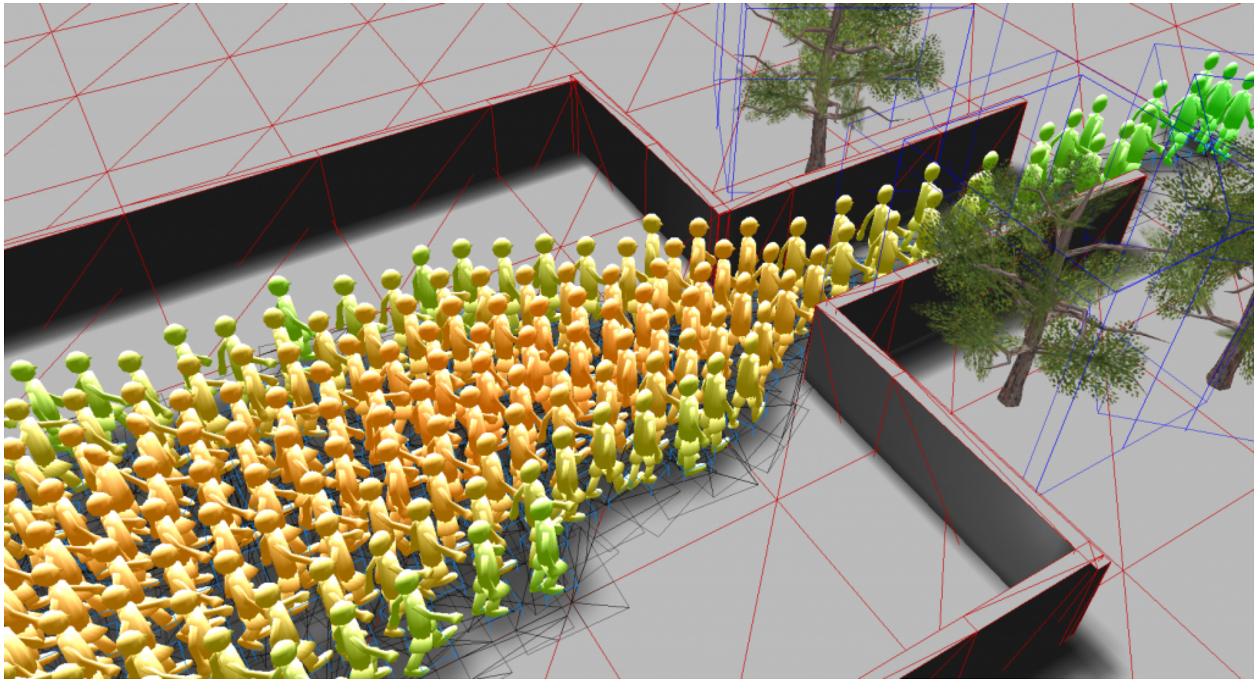


Organizational issues

Next lecture, reports

1. The next QnA is next week, same time **(on-site + zoom)**. Always look at the schedule on Moodle.
2. Hand in your reports on Moodle before 2023-11-02
3. The next lecture is on 2023-11-02
4. After you implemented a small simulation software in the first exercise, we will take a look at the issues arising when a large simulation software is implemented.
5. Specifically, we will take a look at the software Vadere (see `www.vadere.org`).

Questions?



Homework A: send me a PM or email to get access to the course chat.
Homework B: find group, finish first exercise & upload report until 2023-11-02.
For questions / appointments: please ask via email, `felix.dietrich@tum.de`.