



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Machine learning with graphs

Course

Field of study

Artificial intelligence

Area of study (specialization)

Level of study

Second-cycle studies

Form of study

full-time

Year/Semester

2/3

Profile of study

general academic

Course offered in

English

Requirements

elective

Number of hours

Lecture

15

Tutorials

Laboratory classes

15

Projects/seminars

Other (e.g. online)

Number of credit points

2

Lecturers

Responsible for the course/lecturer:

Paweł Misiorek, PhD

email: pawel.misiorek@put.poznan.pl

tel. 61 665-3958

Institute of Computing Science, Faculty of

Computing and Telecommunications

ul. Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

A student starting this course should have basic knowledge in mathematics, including basics of linear algebra, probability and graph theory. Moreover, she/he should know the fundamental concepts and rules of machine learning, statistics and data analysis as well as should have programming skills. The student should also be capable of knowledge acquisition from selected sources, as well as express the readiness for collaborating as the part of a team.

Course objective

The course is focused on graph-based techniques of mining complex data and their applications to machine learning problems. In particular, the course introduces the concept of hypergraphs as a generalization of graphs in which hyperedges may contain any number of nodes, enabling more accurate modeling of real-world networks and data structures. The course includes the following topics:



basics of graph and hypergraph modeling, random graph/hypergraph models and their applications, network mining techniques such as centrality measures or finding most important nodes, community detection algorithms, modularity, graph and hypergraph embeddings and their applications.

Course-related learning outcomes

Knowledge

K2st_W3: has advanced detailed knowledge regarding selected issues in artificial intelligence and related fields

K2st_W4: has knowledge about development trends and the most important cutting edge achievements in computer science, artificial intelligence and other selected and related scientific disciplines

Skills

K2st_U1: is able to obtain information from literature, databases and other sources (both in Polish and English), integrate them, interpret and critically evaluate them, draw conclusions and formulate and fully justify opinions

K2st_U3: is able to plan and carry out experiments, including computer measurements and simulations, interpret the obtained results and draw conclusions and formulate and verify hypotheses related to complex engineering problems and simple research problems

K2st_U5: can - when formulating and solving engineering tasks - integrate knowledge from different areas of computer science and artificial intelligence (and if necessary also knowledge from other scientific disciplines) and apply a systemic approach, also taking into account non-technical aspects

K2st_U16: can determine the directions of further learning and implement the process of self-education, including other people

Social competences

K2st_K2: understands the importance of using the latest knowledge in the field of computer science and artificial intelligence in solving research and practical problems

K2st_K4: is aware of the need to develop professional achievements and comply with the rules of professional ethics

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lectures: The grade based on the assessment test conducted at the last lecture. The test will include several practical tasks and a few theoretical questions concerning the scope of lectures (the list of issues used to state the questions and tasks will be sent to students two weeks before the test). Based on points achieved, a standard scale is used to derive the final mark. The passing threshold: 50%.

Laboratory classes: Students solve practical or programming assignments and report their solutions within two weeks. Each task is evaluated using a scale from 0.0 (lack of the solution) to 5.0 (full solution). The final grade is computed as an average from the individual grades.



Programme content

- 1) Basics of graph and hypergraph modeling. Open-source libraries for graph/hypergraph building, processing and visualization (including iGraph, hypernetx, networkx)
- 2) Complex network modeling methods. Random graph models. Random hypergraph models. Building synthetic and null models based on random graphs.
- 3) Network mining techniques such as indicating node centrality, finding most important nodes or investigating degree correlations.
- 4) Community detection algorithms including the problem of overlapping communities. Graph and hypergraph modularity.
- 5) Graph embeddings. Measures for embedding quality evaluation.
- 6) Basics of Graph Neural Networks.
- 7) Summary of graph applications in machine learning for data cleaning, feature importance, feature design, link prediction, node classification, graph classification.

Teaching methods

Lectures: multimedia presentations, demonstration of examples with solutions, code analysis

Laboratory classes: practical coding exercises (using Python scripts and open-source libraries), discussion on exemplary solutions, teamwork

Bibliography

Basic

Bogumił Kamiński, Paweł Prałat, François Théberge: Mining Complex Networks, Chapman and Hall/CRC, 2021.

William L. Hamilton, Graph Representation Learning, Morgan & Claypool Publishers, 2020.
(https://www.cs.mcgill.ca/~wlh/grl_book/)

Peter Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012.

Additional

Albert-László Barabási, Network Science, Cambridge University Press, 2017.

M. Newman. Networks. Oxford University Press, 2018

A.N. Langville, C.D. Meyer. Google's PageRank and beyond: The science of search engine rankings. Princeton University Press, 2011

F. Chung, L. Lu, Complex Graphs and Networks, American Mathematical Society, 2006.



Breakdown of average student's workload

	Hours	ECTS
Total workload	50	2,0
Classes requiring direct contact with the teacher	30	1,0
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) ¹	20	1,0

¹ delete or add other activities as appropriate