POZNAN UNIVERSITY OF TECHNOLOGY
EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)
pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

COURSE DESCRIPTION CARD - SYLLABUS

Course name
Information Theory and Data Compression Methods

Course

Field of study
Computing

Year/semester
1/2

Area of study (specialization)
ITI

Profile of study
general academic

Level of study
Second-cycle studies

Course offered in
Polish

Form of study
full-time

Requirements
elective

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
30	30	
Tutorials	Projects/seminars	

Number of credit points

5

Lecturers

Responsible for the course/lecturer:
Robert Susmaga
e-mail: robert.susmaga@cs.put.poznan.pl
tel.: 61 6652934
Faculty of Computing and Telecommunications
Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

Basic knowledge regarding:
a) data structures (one- and two-dimensional arrays, lists, trees),
b) probability theory (probability, including conditional probability, probability distributions, expected value),
c) calculus (logarithmic function).
Basic skills regarding designing, creating and testing computer programs (in a programming language of one’s choice) that implement simple processing of static (vectors and matrices) and dynamic (lists, trees) data structures.
(recommended) A fair amount of cognitive curiosity and not less perseverance in pursuing the goals of personal development.

Course objective

The objective of the course is to present a selection of aspects of the Information Theory, one of the most fundamental theories underlying theoretical Computer Science of modern-day. The Information Theory deals with representing, storing and communicating information expressed in the form of symbols. Owing to the fact that many important applications of this theory reach far beyond the core of Computing Science, the presented selection of aspects will be confined to the most fundamental ones, mainly those related to such domains of the Computer Science as Data Compression. The fruits of rapid development of notions in the Information Theory, initially disputable and professedly unsolvable, have soon turned out to be incredibly useful and to have a great deal of practical value. In the modern-day these solutions show up in virtually all imaginable computer systems in existence, ones that could hardly survive nowadays without the ubiquitous multimedia content, the popularity and versatility of which has been consistently and unwaveringly influenced by the accessibility of data compression methods.
Detailed objectives of the course include sharing skills and knowledge sufficient for:
a) understanding the fundamental ideas underlying Information Theory and Data Compression,
b) identifying, formulating and solving problems in Information Theory and Data Compression, also using theoretical notions and justifications,
c) designing and creating computer programs that successfully implement the presented methods.

Course-related learning outcomes

Knowledge

The student:
-- has advanced and in-depth knowledge of widely understood information systems, theoretical foundations of their construction and methods, tools and programming environments used to implement them (K2st_W1)
-- has a structured and theoretically founded general knowledge related to key issues in the field of computer science, especially in Information Theory and its applications to Data Compression (K2st_W2)
-- has advanced detailed knowledge regarding selected IT issues related to Information Theory and Data Compression (K2st_W3)
-- has advanced and detailed knowledge of the processes occurring in the life cycle of hardware or software information systems (K2st_W5)
-- knows advanced methods, techniques and tools used to solve complex engineering tasks and conduct research in Information Theory and Data Compression (K2st_W6)

Skills

The student:
-- is able to obtain information from literature, databases and other sources, integrate them, interpret and critically evaluate them, draw conclusions and formulate and fully justify opinions (K2st_U1)
-- can use analytical, simulation and experimental methods to formulate and solve engineering problems and simple research problems arising in Information Theory and Data Compression (K2st_U4)
-- can -- when formulating and solving engineering tasks -- integrate knowledge from different areas of Information Theory and Data Compression (and if necessary also knowledge from other scientific disciplines) and apply a systemic approach, also taking into account non-technical aspects (K2st_U5)
-- is able to assess the suitability and the possibility of using new achievements (methods and tools) and

new Data Compression products (K2st_U6)
-- can carry out a critical analysis of existing technical solutions and propose their improvements (K2st_U8)
-- is able to assess the usefulness of methods and tools for solving an engineering task, consisting in the construction or evaluation of a Data Compression system or its components, including the limitations of these methods and tools (K2st_U9)
-- can determine the directions of further learning and implement the process of self-education, including other people (K2st_U16)
Social competencies.
The student:
-- understands that in the field of IT the knowledge and skills quickly become obsolete (K2st_K1)
-- understands the importance of using the latest knowledge in the field of computer science in solving research and practical problems (K2st_K2)
-- understands the importance of popularization activities concerning the latest achievements in the field of computer science (K2st_K3)

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:
Formative assessment (laboratory classes): evaluation of the solutions to the assigned programming problems (as they arise).
Final assessment:
-- (laboratory classes): evaluation of the solutions to the assigned programming problems,
-- (lectures): evaluation of the results of a written test (45--60 min) with both multiple choice, short answer and (small) computational questions (mostly: micro-problems to be solved in writing).

Programme content

The course includes, but is not limited to, the following.
Fundamentals of:
-- linear algebra (linear / convex combinations of vectors),
-- calculus (fundamental properties and graphs of log(x) and x*log(x)),
-- probability theory (discrete random variables): probability and conditional probability, probability distributions.
Information and an information measure. Basic properties of the information measure. The measure of Hartley. The measure of Shannon (entropy): basic mathematical properties (graphs and extrema in two- and three-dimensional cases). The entropy as the measure of information. Applications of the measures of Hatley and Shannon to text, audio and video transmissions.
Multi-dimensional aspects of entropy (discrete data): joint entropy; conditional entropy and mutual information; dependencies between the entropy and the mutual information; their applications in a data analysis contexts.

Three types of data processing: encoding/decoding, encrypting/decrypting and hashing. Compression (discrete data): idea and objectives. Compression as a form of coding. The two basic types of compression (lossless and lossy). The two basic techniques of compression (code-based and dictionary-based).
Encoding/decoding (discrete data): the idea and objectives. Codes: definitions and examples, tree representations of codes, prefix codes. Kraft's inequality. Data encoding aimed at compressing.
Code-based compression (discrete data): Shannon-Fano encoding and Huffman encoding: the idea, code trees, optimal codes, algorithms, examples, properties. Entropy in code-based compression.
Dictionary-based lossless compression: Lempel-Ziv-Welch method: the idea, representations, dictionaries, algorithms, examples, properties.
Modern day compression systems. The future of compression.
(Optionally) Extended application of entropy-related notions in data analysis (relative entropy by Kullback-Leibler and its application); adaptive Huffman coding; arithmetic coding: the idea, probability intervals, algorithms, examples; further dictionary based methods (LZ77, LZ78), Burrows-Wheeler transform: the idea, encoding/decoding.
(Laboratory classes)
Introduction to the Python programming language and a selection of its libraries: NumPy and Matplotlib.
Simple programs operating on scalar and vector arguments. Probability distributions, probabilistic approximations to languages.
Selected multidimensional aspects of entropy in data analysis (feature selection, document queries).
Applications of selected code-based methods in compression: Shannon-Fano coding and Huffman coding.
Applications of selected dictionary-based methods in compression: LZW.
(Optional) Adaptive Huffman coding. Arithmetic coding. LZ77/LZ78 methods.

Teaching methods

Lectures: extensive slide show presentations (theoretical elements, explanations, practical examples, exercises), exemplary computations and visualizations.
Laboratory classes: instructions: short slide show presentations (examples, exercises); practice: designing and creating (in a programming language of one's choice) programs to solve the assigned problems (which illustrate the ideas and notions presented during the lectures).

Bibliography

Basic

1. A. Drozdek: "Wprowadzenie do kompresji danych", WNT, Warszawa, 1999.
2. A. Przelaskowski: "Kompresja danych. Podstawy, metody bezstratne, kodery obrazów", BTC, Legionowo, 2005.

Additional

1. Teaching materials (slide show presentations).
2. T.M. Cover, J.A. Thomas, "Elements of Information Theory", 2nd Edition, Wiley & Sons, Hoboken, New Jersey, 1991.
3. D.J.C. MacKay: "Information Theory, Inference, and Learning Algorithms", Cambridge University Press, Cambridge, UK, 2003.
4. K. Sayood (red.): "Lossless Compression Handbook", Academic Press, Elsevier Science, San Diego, California, 2003.
5. K. Sayood: "Introduction to Data Compression", 3rd Ed., Morgan Kaufmann Publishers, San Francisco, California, 2006.

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,0
Classes requiring direct contact with the teacher	60	2,4
Student's own work (literature studies, preparation for laboratory classes/tutorials, creating/testing/experimenting with programs, preparation for tests/exams)	65	2,6