



UNIVERSITÄT
DES
SAARLANDES

FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

MODULHANDBUCH

Data Science and Artificial Intelligence BSc

1. Oktober 2026

Liste der Modulbereiche und Module

1	Ringvorlesungen	3
1.1	Perspektiven der Informatik	4
2	Grundlagen der Mathematik	5
2.1	Mathematik für Informatiker 1	6
2.2	Mathematik für Informatiker 2	8
2.3	Mathematik für Informatiker 3	10
3	Grundlagen der Informatik	12
3.1	Big Data Engineering	13
3.2	Grundzüge der Theoretischen Informatik	16
3.3	Grundzüge von Algorithmen und Datenstrukturen	18
3.4	Programming 1	19
3.5	Programming 2	21
4	Spezialisierter Bereich DSAI	23
4.1	Elements of Data Science and Artificial Intelligence	24
4.2	Elements of Machine Learning	25
4.3	Statistics Lab	27
5	Stammvorlesungen DSAI	29
5.1	Artificial Intelligence	30
5.2	Automated Reasoning	32
5.3	Database Systems	33
5.4	Image Processing and Computer Vision	35
5.5	Information Retrieval and Data Mining	37
5.6	Machine Learning	38
5.7	Neural Networks: Theory and Implementation	39
6	Vertiefungsvorlesungen DSAI	40
6.1	Mathematische Statistik	41

6.2	Security	42
6.3	Stochastik I	43
6.4	Stochastik II	44
6.5	Trusted AI Planning	45
7	Seminare DSAI	47
7.1	Seminar Informatik	48
8	Projektseminare DSAI	50
8.1	Projektseminar DSAI	51
9	Bachelor-Seminar und Arbeit	52
9.1	Bachelor-Seminar	53
9.2	Bachelor-Arbeit	54

Modulbereich 1

Ringvorlesungen

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1	6	jedes Wintersemester	1 Semester	2	2

Modulverantwortliche/r Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

Dozent/inn/en Dozent/inn/en der Fachrichtung

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Nachweis über das inhaltliche Verständnis von mindestens drei Vorträgen, z.B. durch schriftliche Ausarbeitung oder Test.

Lehrveranstaltungen / SWS 2 SWS Vorlesung

Arbeitsaufwand 30 h Präsenzstudium
+ 30 h Eigenstudium
= 60 h (= 2 ECTS)

Modulnote Das Modul ist insgesamt bestanden, wenn die Prüfungsleistung bestanden wurde (unbenotet).

Sprache Deutsch / English

Lernziele / Kompetenzen

Frühzeitige Motivierung und Überblick über die zentralen wissenschaftlichen Fragestellungen der Informatik, sowie über die Kompetenzen der Saarbrücker Informatik.

Inhalt

Vorträge durch wöchentlich wechselnde Dozent/inn/en bieten einen Querschnitt durch die Forschungsthemen der Saarbrücker Informatik. Die Themen spannen einen attraktiven Bogen von aktuellster Forschung zu anspruchsvollen Problemen der industriellen Praxis.

Literaturhinweise

Material wird passend zu den zu den einzelnen Vorträgen bereitgestellt.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Perspectives in Computer Science*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1	6	jedes Wintersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Joachim Weickert

Dozent/inn/en Prof. Dr. Joachim Weickert
 Prof. Dr. Mark Groves
 Prof. Dr. Henryk Zähle
 Prof. Dr. Christian Bender

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

- Teilnahme an den Übungen und Bearbeitung der wöchentlichen Übungsaufgaben (50 Prozent der Übungspunkte werden zur Klausurteilnahme benötigt)
- Bestehen der Abschlussklausur oder der Nachklausur

Lehrveranstaltungen / SWS 4 SWS Vorlesung
 + 2 SWS Übung
 = 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
 + 180 h Eigenstudium
 = 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Deutsch und Englisch

Lernziele / Kompetenzen

- Erarbeitung von mathematischem Grundlagenwissen, das im Rahmen eines Informatik- bzw. Bioinformatikstudiums benötigt wird
- Fähigkeit zur Formalisierung und Abstraktion
- Befähigung zur Aneignung weiteren mathematischen Wissens mit Hilfe von Lehrbüchern

Inhalt

Die Zahlen geben die Gesamtzahl der Doppelstunden an.

DISKRETE MATHEMATIK UND EINDIMENSIONALE ANALYSIS

- A. Grundlagen der diskreten Mathematik (8)
1. Mengen (1)
 2. Logik (1)
 3. Beweisprinzipien, incl. vollst. Induktion (1)
 4. Relationen (1)
 5. Abbildungen (2)
 - injektiv, surjektiv, bijektiv
 - Mächtigkeit, Abzählbarkeit
 - Schubfachprinzip
 6. Primzahlen und Teiler (1)
 7. Modulare Arithmetik (1)

B. Eindimensionale Analysis (22)

B.1 Zahlen, Folgen und Reihen (8)

8. Axiomatik der reellen Zahlen, $\sup$, $\inf$ (1)
9. Komplexe Zahlen (1)
10. Folgen (1 1/2)
11. Landau'sche Symbole (1/2)
12. Reihen: Konvergenzkriterien, absolute Kgz. (2)
13. Potenzreihen (1/2)
14. Zahlendarstellungen (1/2)
15. Binomialkoeffizienten und Binomialreihe (1)

B.2 Eindimensionale Differentialrechnung (8)

16. Stetigkeit (1)
17. Elementare Funktionen (1)
18. Differenzierbarkeit (1 1/2)
19. Mittelwertsätze und L'Hospital (1/2)
20. Satz von Taylor (1)
21. Lokale Extrema, Konvexität, Kurvendiskussion (2)
22. Numerische Differentiation (1)

B.3 Eindimensionale Integralrechnung (6)

23. Das bestimmte Integral (2)
24. Das unbestimmte Integral und die Stammfunktion (1)
25. Uneigentliche Integrale (1)
26. Numerische Verfahren zur Integration (1)
27. Kurven und Bogenlänge (1)

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Mathematics for Computer Scientists 1*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
2	6	jedes Sommersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Joachim Weickert

Dozent/inn/en Prof. Dr. Joachim Weickert
 Prof. Dr. Mark Groves
 Prof. Dr. Henryk Zähle
 Prof. Dr. Christian Bender

Zulassungsvoraussetzungen Mathematik für Informatiker 1 (empfohlen)

Leistungskontrollen / Prüfungen

- Teilnahme an den Übungen und Bearbeitung der wöchentlichen Übungsaufgaben (50 Prozent der Übungspunkte werden zur Klausurteilnahme benötigt)
- Bestehen der Abschlussklausur oder der Nachklausur

Lehrveranstaltungen / SWS 4 SWS Vorlesung
 + 2 SWS Übung
 = 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
 + 180 h Eigenstudium
 = 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Deutsch und Englisch

Lernziele / Kompetenzen

- Erarbeitung von mathematischem Grundlagenwissen, das im Rahmen eines Informatik- bzw. Bioinformatikstudiums benötigt wird
- Fähigkeit zur Formalisierung und Abstraktion
- Befähigung zur Aneignung weiteren mathematischen Wissens mit Hilfe von Lehrbüchern

Inhalt

Die Zahlen geben die Gesamtzahl der Doppelstunden an.

LINEARE ALGEBRA

C. Algebraische Strukturen (5)

- 29. Gruppen (2)
- 30. Ringe und Körper (1)
- 31. Polynomringe über allgemeinen Körpern (1/2)
- 32. Boole'sche Algebren (1/2)

D. Lineare Algebra (21)

- 33. Vektorräume (2)
 - Def., Bsp.,
 - lineare Abb.
 - Unterraum,
 - Erzeugnis, lineare Abhängigkeit, Basis, Austauschsatz

- 34. Lineare Abb. (Bild, Kern) (1)
- 35. Matrixschreibweise für lineare Abbildungen (1 1/2)
 - Interpretation als lineare Abbildungen
 - Multiplikation durch Hintereinanderausführung
 - Ringstruktur
 - Inverses
- 36. Rang einer Matrix (1/2)
- 37. Gauss-Algorithmus für lineare Gleichungssysteme: (2)
 - Gausselimination (1)
 - Lösungstheorie (1)
- 38. Iterative Verfahren für lineare Gleichungssysteme (1)
- 39. Determinanten (1)
- 40. Euklidische Vektorräume, Skalarprodukt (1)
- 41. Funktionalanalytische Verallgemeinerungen (1)
- 42. Orthogonalität (2)
- 43. Fourierreihen (1)
- 44. Orthogonale Matrizen (1)
- 45. Eigenwerte und Eigenvektoren (1)
- 46. Eigenwerte und Eigenvektoren symmetrischer Matrizen (1)
- 47. Quadratische Formen und positiv definite Matrizen (1)
- 48. Quadriken (1)
- 50. Matrixnormen und Eigenwertabschätzungen (1)
- 51. Numerische Berechnung von Eigenwerten und Eigenvektoren (1)

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Mathematics for Computer Scientists 2*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
3	6	jedes Wintersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Joachim Weickert

Dozent/inn/en Prof. Dr. Joachim Weickert
 Prof. Dr. Mark Groves
 Prof. Dr. Henryk Zähle
 Prof. Dr. Christian Bender

Zulassungsvoraussetzungen Mathematik für Informatiker 1 und 2 (empfohlen)

Leistungskontrollen / Prüfungen

- Teilnahme an den Übungen und Bearbeitung der wöchentlichen Übungsaufgaben (50 Prozent der Übungspunkte werden zur Klausurteilnahme benötigt)
- Bestehen der Abschlussklausur oder der Nachklausur

Lehrveranstaltungen / SWS 4 SWS Vorlesung
 + 2 SWS Übung
 = 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
 + 180 h Eigenstudium
 = 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Englisch

Lernziele / Kompetenzen

- Erarbeitung von mathematischem Grundlagenwissen, das im Rahmen eines Informatik- bzw. Bioinformatikstudiums benötigt wird
- Fähigkeit zur Formalisierung und Abstraktion
- Befähigung zur Aneignung weiteren mathematischen Wissens mit Hilfe von Lehrbüchern

Inhalt

Die Zahlen geben die Gesamtzahl der Doppelstunden an.

STOCHASTIK, NUMERIK UND MEHRDIMENSIONALE ANALYSIS

E. NUMERISCHE ERGÄNZUNGEN (3)

- 52. Banachscher Fixpunktsatz (1)
- 53. Interpolation, incl. Splines (2)

F. MEHRDIMENSIONALE ANALYSIS UND NUMERIK (11)

- 54. Stetigkeit und Differentialoperatoren für skalarwertige Funktionen (2)
- 55. Differentialoperatoren für vektorwertige Funktionen (1)
- 56. Totale Differenzierbarkeit (1/2)
- 57. Mittelwertsatz und Satz von Taylor (1 1/2)
- 58. Extrema von Funktionen mehrerer Variabler (1)
- 59. Das Newton-Verfahren (1)
- 60. Extrema mit Nebenbedingungen (1)

- 61. Mehrfachintegrale (1)
- 62. Die Umkehrfunktion und die Transformationsregel (1)
- 63. Variationsrechnung (1)

G. STOCHASTIK (16)

- 64. Grundbegriffe (Ws., Stichprobenraum) (1/3)
- 65. Kombinatorik (2/3)
- 66. Erzeugende Funktionen (1)
- 67. Bedingte Wahrscheinlichkeiten (1)
- 68. Zufallsvariable, Erwartungswert, Varianz (2) (Systemzuverlässigkeit, Varianz, Kovarianz, Jensen)
- 69. Abschätzungen für Abweichungen vom Mittelwert (1) (Momente, Schranken von Markov, Chebyshev, Chernoff, schwaches Gesetz der grossen Zahlen)
- 70. Wichtige diskrete Verteilungen (1)
- 71. Wichtige kontinuierliche Verteilungen (1) (incl. zentraler Grenzwertsatz)
- 72. Multivariate Verteilungen und Summen von Zufallsvariablen (1)
- 73. Parameterschätzung und Konfidenzintervalle (1)
- 74. Hypothesentests (1)
- 75. Methode der kleinsten Quadrate (1)
- 76. Robuste Statistik (2/3)
- 77. Fehlerfortpflanzung (1/3)
- 78. Markowketten (2)
- 79. Pseudozufallszahlen und Monte-Carlo-Simulation (1)

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Mathematics for Computer Scientists 3.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4	6	jedes Sommersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Jens Dittrich

Dozent/inn/en Prof. Dr. Jens Dittrich

Zulassungsvoraussetzungen *Programming 1, Programming 2, Softwarepraktikum oder Projektpraktikum, Mathematik für Informatiker 1, sowie Grundzüge von Algorithmen und Datenstrukturen (jeweils empfohlen).*

Leistungskontrollen / Prüfungen Erfolgreiche Teilnahme an den Übungen/Projekt berechtigt zur Teilnahme an der Abschlussklausur.

Lehrveranstaltungen / SWS 2 SWS Vorlesung
+ 2 SWS Übung
= 4 SWS

Arbeitsaufwand 60 h Präsenzstudium
+ 120 h Eigenstudium
= 180 h (= 6 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen, und ggf. Projekt ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekanntgegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Vorlesung vermittelt grundlegende Kenntnisse über fundamental Konzepte von Datenmanagement und Datenanalyse im Kontext von Big Data und Data Science.

Im Rahmen der Übungen kann während des Semesters ein durchgehendes Projekt durchgeführt. Dies kann zum Beispiel ein soziales Netzwerk (im Stil von Facebook) sein bzw. jedes andere Projekt, in dem Techniken des Datenmanagements eingeübt werden können (z.B. naturwissenschaftliche Daten, Bilddaten, andere Webapplikationen, etc.). Zunächst wird dieses Projekt in E/R modelliert, dann umgesetzt und implementiert in einem Datenbankschema. Danach wird das Projekt erweitert, um auch unstrukturierte Daten verwalten und analysieren zu können. Insgesamt werden so an einem einzigen Projekt alle fundamentalen Techniken gezeigt, die für das Verwalten und Analysieren von Daten wichtig sind.

Inhalt

1 Einführung und Einordnung

Einordnung und Abgrenzung: Data Science
Wert von Daten: Das Gold des 21. Jahrhunderts
Bedeutung von Datenbanksystemen
Architekturen: 2-Tier, 3-Tier, etc
Was sind eigentlich Daten?
Modellierung vs Realität
Kosten mangelhafter Modellierung
Datenbanksystem nutzen vs selbst entwickeln
Positive Beispiele für Apps
Anforderungen
Literaturhinweise
Vorlesungsmodus

- 2 Datenmodellierung
 - Motivation
 - E/R
 - Relationales Modell
 - Hierarchische Daten
 - Graphen und RDF
 - Redundanz, Normalisierung, Denormalisierung
 - Objektrelationale DBMS
- 3 Anfragesprachen
 - Relationale Algebra
 - Hierarchische Anfragesprachen
 - Graphorientierte Anfragesprachen
- 4 SQL
 - Grundlagen
 - Zusammenhang mit relationaler Algebra
 - PostgreSQL
 - Integritätsbedingungen
 - Transaktionskonzept
 - ACID
 - Sichten (und access control lists)
- 5 Implementierungstechniken
 - Übersicht
 - vom WAS zum WIE
 - Kosten verschiedener Operationen
 - EXPLAIN
 - Physisches Design
 - Indexe, Tuning
 - Datenbank-Tuning
 - Regelbasierte Anfrageoptimierung
 - Kostenbasierte Anfrageoptimierung
 - Machine Learning als Anfrageoptimierungstechnik
- 6 Zeitliche und räumliche Daten
 - als Teil des Schemas
 - as of/time travel
 - append-only und Streaming
 - Versioning
 - Snapshotting (Software und OS-basiert)
 - Differential Files/LSM et al
 - Publish/Subscribe
 - Indexstrukturen
- 7 Recovery, Durability, Archivierung
 - Grundproblematik
 - Vergessen vs Komprimieren vs Kondensieren
 - Heiße vs kalte Daten
 - Archivierung
 - Redundanz
 - Implementierungsaspekte
 - UNDO/REDO
 - Logging
- 8 Nebenläufigkeitskontrolle
 - Serialisierbarkeitstheorie
 - Isolationslevels
 - Verteilte Datenbanksysteme: Sharding, HP, VP, permissioned Blockchains
 - Implementierungsaspekte

9 ETL und Data Cleaning

- Datenbankschnittstellen: JDBC et al
- Textdatenbanken: CSV, SQLite
- Data Warehousing
- Schema Matching
- Reporting
- Data Cleaning
- Denormalisierung, Caching, Materialisierung
- Workflows
- ETL und Data Science in Data Science und Machine Learning

10 Big Data

- Was ist eigentlich Big Data?
- Big Data vs Privatheit
- Beispiele: Zusammenführen von Daten
- Physische Barrieren

11 NoSQL

- Key/Value Stores
- KeyDocument Stores: MongoDB
- MapReduce
- Flink
- Spark

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul wurde früher auch unter dem Namen *Informationssysteme* geführt. Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Big Data Engineering*.

Informationen für "Mathematik und Informatik BSc": Diese Veranstaltung ist Teil des Wahlpflicht-Bereichs der Grundlagen der Informatik (siehe Studienordnung §5 (2) 4)

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
3	6	jedes Wintersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Raimund Seidel

Dozent/inn/en Prof. Dr. Raimund Seidel
Prof. Dr. Bernd Finkbeiner
Prof. Dr. Kurt Mehlhorn
Prof. Dr. Markus Bläser

Zulassungsvoraussetzungen *Programming 1* und *2* und *Mathematik für Informatiker 1* und *2* oder vergleichbare Veranstaltungen der Mathematik sind empfohlen.

Leistungskontrollen / Prüfungen Erfolgreiche Bearbeitung der Übungsaufgaben berechtigt zur Klausurteilnahme.

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
+ 180 h Eigenstudium
= 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Studierenden kennen verschiedene Rechenmodelle und ihre relativen Stärken und Mächtigkeiten.

Sie können für ausgewählte Probleme zeigen, ob diese in bestimmten Rechenmodellen lösbar sind oder nicht.

Sie verstehen den formalen Begriff der Berechenbarkeit wie auch der Nicht-Berechenbarkeit.

Sie können Probleme aufeinander reduzieren.

Sie sind vertraut mit den Grundzügen der Ressourcenbeschränkung (Zeit, Platz) für Berechnungen und der sich daraus ergebenden Komplexitätstheorie.

Inhalt

Die Sprachen der Chomsky Hierarchie und ihre verschiedenen Definitionen über Grammatiken und Automaten; Abschlusseigenschaften; Klassifikation von bestimmten Sprachen („Pumping lemmas“);

Determinismus und Nicht-Determinismus;

Turing Maschinen und äquivalente Modelle von allgemeiner Berechenbarkeit (z.B. μ -rekursive Funktionen, Random Access Machines) Reduzierbarkeit, Entscheidbarkeit, Nicht-Entscheidbarkeit;

Die Komplexitätsmaße Zeit und Platz; die Komplexitätsklassen P und NP;

Grundzüge der Theorie der NP-Vollständigkeit

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Introduction to Theoretical Computer Science*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
3	6	jedes Wintersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Raimund Seidel

Dozent/inn/en Prof. Dr. Raimund Seidel
Prof. Dr. Kurt Mehlhorn
Prof. Dr. Markus Bläser

Zulassungsvoraussetzungen *Programming 1* und 2, und *Mathematik für Informatiker 1* und 2 oder vergleichbare Veranstaltungen der Mathematik sind empfohlen.

Leistungskontrollen / Prüfungen Erfolgreiche Bearbeitung der Übungsblätter berechtigt zur Klausurteilnahme.

Lehrveranstaltungen / SWS 2 SWS Vorlesung
+ 2 SWS Übung
= 4 SWS

Arbeitsaufwand 60 h Präsenzstudium
+ 120 h Eigenstudium
= 180 h (= 6 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Studierenden lernen die wichtigsten Methoden des Entwurfs von Algorithmen und Datenstrukturen kennen: Teile-und-Herrsche, Dynamische Programmierung, inkrementelle Konstruktion, „Greedy“, Dezimierung, Hierarchisierung, Randomisierung. Sie lernen Algorithmen und Datenstrukturen bzgl. Zeit- und Platzverbrauch für das übliche RAM Maschinenmodell zu analysieren und auf Basis dieser Analysen zu vergleichen. Sie lernen verschiedene Arten der Analyse (schlechtester Fall, amortisiert, erwartet) einzusetzen.

Die Studierenden lernen wichtige effiziente Datenstrukturen und Algorithmen kennen. Sie sollen die Fähigkeit erwerben, vorhandene Methoden durch theoretische Analysen und Abwägungen für ihre Verwendbarkeit in tatsächlich auftretenden Szenarien zu prüfen. Ferner sollen die Studierenden die Fähigkeit trainieren, Algorithmen und Datenstrukturen unter dem Aspekt von Performanzgarantien zu entwickeln oder anzupassen

Inhalt

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Fundamentals of Data Structures and Algorithms*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1	6	every winter semester	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Prof. Dr. Sebastian Hack
Prof. Dr. Martina Maggio

Zulassungsvoraussetzungen none

Leistungskontrollen / Prüfungen

- Weekly exercises and tests
- Mid-term and final exam
- Re-examination at end of the semester

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English and German

Lernziele / Kompetenzen

Programming 1 is an introduction to the foundations of computer science. Using the functional programming language OCaml, students learn to write programs, to understand precisely what programs mean, to prove them correct, and to analyse their running time. From the first day on, students work hands-on with an interpreter, which serves as a laboratory for experimenting with programs and with the programming language itself.

Upon completion of the module, students

- can write well-structured functional programs in OCaml, using recursion, tail recursion and accumulators, higher-order and polymorphic functions, pattern matching, algebraic data types, options, and exceptions;
- have acquired practical programming skills: experimenting with an interpreter, reading and reacting to type errors, systematic testing and debugging, and using the standard library;
- can design and implement recursive algorithms on numbers, lists, strings, and trees (e.g. sorting algorithms, search trees, tries, prime factorisation);
- understand static typing, polymorphism, and type inference;
- can structure programs with modules, signatures, abstract data types, representation invariants, and functors;
- can program with mutable state (references, arrays, loops, imperative data structures) and understand how data is represented in linear memory;
- can specify functions mathematically, reason about termination, and prove recursive programs correct by induction (natural, structural, and well-founded induction);
- can determine and classify the time complexity of algorithms (running-time functions, recurrences, asymptotic complexity classes);
- understand the structure of programming languages and their formal description by grammars and inference rules (abstract and concrete syntax, static and dynamic semantics);
- can implement the components of a language processor for a small functional language: lexer, a simple parser, type checker, and evaluator;
- are able to use the basic mathematical structures underlying computer science: sets, propositions, tuples, relations, directed graphs, functions, and well-founded orders.

Inhalt

see above

Literaturhinweise

Lecture notes (in English) are provided to the participants.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
2	6	every summer semester	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Sebastian Hack

Dozent/inn/en Prof. Dr. Sebastian Hack
Prof. Dr. Holger Hermanns

Zulassungsvoraussetzungen *Programming 1* and *Mathematics for Computer Scientists 1* and mathematics courses in the study semester or comparable knowledge from other mathematics courses are recommended.

Leistungskontrollen / Prüfungen

- Practical part: students independently implement a series of programming projects in RISC-V assembly, C, and Java. The projects practise the language concepts of the lecture and introduce more complex algorithms and data structures. Automatic tests check the quality of the implementations.
- Lecture part: short written tests (minitests) accompanying the lecture, and a written end-of-term exam (with re-exam).
- Admission to the written exam requires successful participation in the programming projects and the minitests. The exact thresholds are announced at the beginning of the module.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in the written exams. Successful participation in the programming projects and minitests is a prerequisite for admission to the exams. The exact modalities will be announced at the beginning of the module.

Sprache English and German

Lernziele / Kompetenzen

Programming 2 introduces the foundations of imperative and object-oriented programming. The course follows the software stack from the machine level (computer arithmetic, RISC-V machine code) via the systems language C to the object-oriented language Java, and closes the loop with the formal semantics of a small imperative language and a simple compiler that translates it back to machine code. A central goal is that students can identify the different levels of abstraction at work in programming and move between them deliberately, so that they can reason about the behaviour, performance, and security of programs rather than relying on unexamined habits.

After successfully completing the module, students

- understand how computers represent numbers and execute programs, and can read, write, and debug small programs in RISC-V assembly, including function calls, the stack, and a calling convention;
- can implement, debug, and test small to mid-size C programs, and understand the C memory model, pointers, dynamic memory management, translation units, and the consequences of undefined behaviour;
- can design, implement, debug, and test mid-size Java programs using classes, interfaces, inheritance, subtyping, dynamic dispatch, encapsulation, and immutability, use the Java Collections Framework, and can judge when an object-oriented decomposition is appropriate;

- can implement fundamental data structures and algorithms (lists, trees, hash tables, dynamic programming) in an imperative language and choose suitable ones for a given programming task;
- know the formal syntax, operational semantics, and a type system of a small imperative language with pointers and scopes, and can explain type safety and undefined behaviour in these terms;
- can specify the functional correctness of programs, know the fundamental techniques and limits of testing (test oracles, black-box and white-box testing, coverage criteria, fuzzing), can prove simple imperative programs correct with Hoare logic and loop invariants, and understand how such proofs are mechanized via verification conditions;
- understand how a high-level language is translated to machine code by working through the phases of a simple compiler (syntactic analysis, type checking, syntax-directed code generation);
- have practical experience in independently completing programming projects of non-trivial size, including working with an existing code base, automatic tests, and debugging tools.

Inhalt

1. **Computer arithmetic:** positional notation, binary numbers, addition and subtraction, two's-complement integers, shifts
2. **Machine code:** the von Neumann architecture, the RV32I instruction set, the assembler, execution traces, the data segment, memory segmentation, linking, calling functions (stack, calling convention)
3. **Simple algorithms at the machine level:** number conversion, the sieve of Eratosthenes, insertion sort, evaluating polynomials with Horner's method, cycle detection (tortoise and hare)
4. **Introduction to C:** basics, execution traces, imperative variables, the C memory model, translation units, number types, control flow, expressions, calling functions, pointers, dynamic memory management, structs, input and output, undefined behaviour
5. **Algorithms and data structures:** lists, trees, hashing, dynamic programming
6. **Formal semantics:** the syntax of programming languages, abstract syntax and operational semantics of the language C0, extension by pointers (C0p) and scopes (C0b), a simple type system (C0t) and type safety
7. **Testing and verification:** functional correctness and specifications, failures, testing (the oracle problem, black-box testing, white-box testing and coverage, fuzzing), verification with Hoare logic, loop invariants, verification condition generation and mechanized verification
8. **Object-oriented programming with Java:** compilation, main method, and packages; types and reference types; classes; encapsulation; references, aliasing, and immutable objects; inheritance, subtyping, and dynamic dispatch; overriding and overloading methods; the Java Collections Framework; using object orientation properly; the expression problem and the visitor pattern; object-oriented programming in C
9. **A simple compiler:** overview of syntactic analysis, type checking, syntax-directed code generation from C0 to RISC-V

as well as lectures specifically designed for the individual programming projects (e.g. algorithmic background such as Dijkstra's algorithm for route planning or bounding volume hierarchies for ray tracing).

Literaturhinweise

Lecture notes: Sebastian Hack, *An Introduction to Imperative Programming*, <https://prog2.de/book>. Further literature will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1	6	every winter semester	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Jörg Hoffmann
Prof. Dr. Jens Dittrich
Prof. Dr. Bernt Schiele
Prof. Dr. Vera Demberg

Dozent/inn/en Prof. Dr. Jörg Hoffmann
Prof. Dr. Jens Dittrich
Prof. Dr. Bernt Schiele
Prof. Dr. Vera Demberg

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Weekly assignments,
Exam (qualification for exam depends on performance in assignments)

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Based on exam. The exact modalities are specified by the lecturers.

Sprache English

Lernziele / Kompetenzen

Overview of challenges and methods in Data Science and AI. Basic knowledge of key concepts and algorithms.

Inhalt

Introduction to history and concepts of Data Science and AI

- Machine Learning (supervised, unsupervised, reinforcement, neural networks)
- (adversarial) Search, Planning
- Reasoning
- Modeling and Simulation
- Data Management, Big Data Engineering, and Analytics

The methods will be covered in the context of applications, such as Game Playing, Computer Vision, Autonomous Driving, Language Processing, Social Networks.

The exercises will cover methodological, algorithmic, as well as practical aspects. Where basic programming or scripting skills are required, the lecture and exercises will introduce these skills.

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
3	6	every winter semester	1 semester	4	6

Modulverantwortliche/r Prof. Dr. Jilles Vreeken
Prof. Dr. Isabel Valera

Dozent/inn/en Prof. Dr. Jilles Vreeken
Prof. Dr. Isabel Valera

Zulassungsvoraussetzungen The lecture assumes basic knowledge in statistics, linear algebra, and programming. It is advisable to have successfully completed *Mathematics for Computer Scientists 2* and *Statistics Lab*. The exercises use the programming language R. We will give a basic introduction to R in the first tutorial. In addition, for preparation the following materials are useful: *R for Beginners* by Emmanuel Paradis (especially chapters 1, 2, 3 and 6) and *An introduction to R* (Venables/Smith).

Leistungskontrollen / Prüfungen Prerequisite for admission to the examination is a cumulative 50% of the points of the theoretical and a cumulative 50% of the points of the practical tasks on the exercise sheets. Depending on the number of participants, the examinations are either written or oral. The final modality will be announced in the first two weeks of the lecture.

Lehrveranstaltungen / SWS 2 h lectures
+ 2 h tutorial
= 4 h (weekly)

Arbeitsaufwand 60 h of classes
+ 120 h private study
= 180 h (= 6 ECTS)

Modulnote Will be determined from performance in exams.

Sprache English

Lernziele / Kompetenzen

In this course we will discuss the foundations – the elements – of machine learning. In particular, we will focus on the ability of, given a data set, to choose an appropriate method for analyzing it, to select the appropriate parameters for the model generated by that method and to assess the quality of the resulting model. Both theoretical and practical aspects will be covered. What we cover will be relevant for computer scientists in general as well as for other scientists involved in data analysis and modeling.

Inhalt

The lecture covers basic machine learning methods, in particular the following contents:

- Introduction to statistical learning
- Overview over Supervised Learning
- Linear Regression
- Linear Classification
- Splines
- Model selection and estimation of the test errors
- Maximum-Likelihood Methods
- Additive Models
- Decision trees

- Boosting
- Dimensionality reduction
- Unsupervised learning
- Clustering
- Visualization

Literaturhinweise

The course broadly follows the book *An Introduction to Statistical Learning with Applications in R*, Springer (2013). In some cases, the course receives additional material from the book *The Elements of Statistical Learning*, Springer (second edition, 2009). The first book is the introductory text, the second covers more advanced topics. Both books are available as free PDFs. Any change of, or additional material will be announced before the start of the course on the course webpage.

Weitere Informationen

Informationen für "Mathematik und Informatik BSc": Diese Veranstaltung ist Teil des Wahlpflicht-Bereichs der Grundlagen der Informatik (siehe Studienordnung §5 (2) 4)

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
2	6	jedes Sommersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Verena Wolf
Prof. Dr. Vera Demberg

Dozent/inn/en Prof. Dr. Verena Wolf
Prof. Dr. Vera Demberg

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen mündliche oder schriftliche Prüfung

Lehrveranstaltungen / SWS 2 SWS Vorlesung
+ 2 SWS Übung
= 4 SWS

Arbeitsaufwand 60 h Präsenzstudium
+ 120 h Eigenstudium
= 180 h (= 6 ECTS)

Modulnote Wird aus Leistungen in der Klausur, sowie den Prüfungsvorleistungen ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben. Alle Modulelemente sind innerhalb eines Prüfungszeitraumes erfolgreich zu absolvieren.

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

- Verständnis der mathematischen Konzepte von Zufallsvariablen und Verteilungen
- Verständnis und Anwendung von Methoden der Punkt- und Intervallschätzung, statistischer Tests
- Verständnis der mathematischen Konzepte von Zustandsdiskreten Markovprozessen und Verwendung solcher Prozesse zur Beschreibung von realen Phänomenen

Inhalt

Probabilities and Discrete Random Variables

- Probability
- discrete RVs
- expectation, variance and quantiles (also visualization of them)
- higher moments
- important discrete probability distributions
- Generating discrete random variates Continuous Random Variables and Laws of Large Numbers
- σ -algebras (very lightweight)
- Continuous Random Variables
- Important Continuous Distributions
- generating continuous random variates
- Chebyshev's inequality
- Weak/Strong Law of Large Numbers
- Central Limit Theorem

Multidimensional Probability Distributions

- joint probability distribution

- conditional probability distribution
- Bayes' Theorem
- covariance and correlation
- independence
- important multidimensional probability distributions

Point Estimation

- (generalized) method of moments
- maximum likelihood estimation
- Bayesian inference (posterior mean/median, MAP)
- Kernel density estimation
- OLS estimator (this is simple regression but should be mentioned here!)
- (shortly: model selection)

Interval Estimation

- confidence intervals for sample mean/variance
- confidence intervals for MLE
- bootstrap confidence interval
- Bayesian credible interval

Statistical Testing

- Level α tests (Z-Test, T-Test)
- p-value
- chi-squared tests, Fisher test
- multiple testing (Bonferroni correction, Holm-Bonferroni method, Benjamini-Hochberg, etc)

Discrete-time Markov chains (only if time)

- transient distributions
- equilibrium distributions
- Monte-Carlo simulation

HMMs

- Baum-Welch-Algorithmus
- Viterbi-Algorithmus

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-5	6	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Jörg Hoffmann

Dozent/inn/en Prof. Dr. Jörg Hoffmann

Zulassungsvoraussetzungen *Programming 1, Programming 2, Fundamentals of Data Structures and Algorithms, and Elements of Machine Learning* or other courses in machine learning are recommended.

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Solving of weekly assignments
- Passing the final written exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from the performance in exams. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Knowledge about basic methods in Artificial Intelligence

Inhalt

Search:

- Uninformed- and informed search procedures
- Monte-Carlo tree search

Planning:

- Formalism and complexity
- Critical-path heuristics
- Delete relaxation heuristics
- Abstraction heuristics

Markov decision processes:

- Discounted reward and expected cost
- Value iteration
- Informed search
- Reinforcement learning

Games:

- Adversarial search
- Learning from self-play

Literaturhinweise

Russel & Norvig Artificial Intelligence: A Modern Approach;
further reading will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-5	6	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Christoph Weidenbach

Dozent/inn/en Prof. Dr. Christoph Weidenbach

Zulassungsvoraussetzungen *Introduction to Computational Logic*

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Weekly assignments
- Practical work with systems
- Passing the final and mid-term exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The goal of this course is to provide familiarity with logics, calculi, implementation techniques, and systems providing automated reasoning.

Inhalt

Propositional Logic – CDCL, Superposition - Watched Literals
First-Order Logic without Equality – (Ordered) Resolution,
Equations with Variables – Completion, Termination
First-Order Logic with Equality – Superposition (SUP) - Indexing

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-5	6	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Jens Dittrich

Dozent/inn/en Prof. Dr. Jens Dittrich

Zulassungsvoraussetzungen especially Saarland University CS department's undergraduate lecture *Big Data Engineering* (former *Informationssysteme*), *Programming 1* and *2*, *Algorithmen und Datenstrukturen* as well as *Nebenläufige Programmierung*

For graduate students:

- motivation for databases and database management systems;
- the relational data model;
- relational query languages, particularly relational algebra and SQL;
- **solid** programming skills in Java and/or C++
- undergrad courses in algorithms and data structures, concurrent programming

Leistungskontrollen / Prüfungen

- Passing a two-hour written exam at the end of the semester
- Successful demonstration of programming project (teams of up to three students are allowed); the project may be integrated to be part of the weekly assignments

Grades are based on written exam; 50% in weekly assignments (in paper and additionally paper or electronic quizzes) must be passed to participate in the final and repetition exams.

A repetition exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

This class may be run as a flipped classroom, i.e. 2 hours of lectures may be replaced by self-study of videos/papers; the other 2 hours may be used to run a group exercise supervised by the professor called "the LAB")

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined based on project, midterm and best of endterm and reexam.

Sprache English

Lernziele / Kompetenzen

Database systems are the backbone of most modern information systems and a core technology without which today's economy – as well as many other aspects of our lives – would be impossible in their present forms. The course teaches the architectural and algorithmic foundations of modern database management systems (DBMS), focussing on database systems internals rather than applications. Emphasis is made on robust and time-tested techniques that have led databases to be considered a mature technology and one of the greatest success stories in computer science. At the same time, opportunities for exciting research in this field will be pointed out.

In the exercise part of the course, important components of a DBMS will be treated and where possible implemented and their performance evaluated. The goal this is to work with the techniques introduced in the lecture and to understand them and their practical implications to a depth that would not be attainable by purely theoretical study.

Inhalt

The course "Database Systems" will introduce students to the internal workings of a DBMS, in particular:

- storage media (disk, flash, main memory, caches, and any other future storage medium)
- data managing architectures (DBMS, streams, file systems, clouds, appliances)
- storage management (DB-file systems, raw devices, write-strategies, differential files, buffer management)
- data layouts (horizontal and vertical partitioning, columns, hybrid mappings, compression, defragmentation)
- indexing (one- and multidimensional, tree-structured, hash-, partition-based, bulk-loading and external sorting, differential indexing, read- and write-optimized indexing, data warehouse indexing, main-memory indexes, sparse and dense, direct and indirect, clustered and unclustered, main memory versus disk and/or flash-based)
- processing models (operator model, pipeline models, push and pull, block-based iteration, vectorization, query compilation)
- processing implementations (join algorithms for relational data, grouping and early aggregation, filtering)
- query processing (scanning, plan computation, SIMD)
- query optimization (query rewrite, cost models, cost-based optimization, join order, join graph, plan enumeration)
- data recovery (single versus multiple instance, logging, ARIES)
- parallelization of data and queries (horizontal and vertical partitioning, shared-nothing, replication, distributed query processing, NoSQL, MapReduce, Hadoop and/or similar and/or future systems)
- read-optimized system concepts (search engines, data warehouses, OLAP)
- write-optimized system concepts (OLTP, streaming data)
- management of geographical data (GIS, google maps and similar tools)
- main-memory techniques

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-5	6	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Joachim Weickert

Dozent/inn/en Prof. Dr. Joachim Weickert

Zulassungsvoraussetzungen Undergraduate mathematics (e.g. Mathematik für Informatiker I-III) and elementary programming knowledge in C

Leistungskontrollen / Prüfungen

- For the homework assignments one can obtain up to 24 points per week. Actively participating in the classroom assignments gives 12 more points per week, regardless of the correctness of the solutions. To qualify for both exams one needs 2/3 of all possible points.
- Passing the final exam or the re-exam.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from the performance in the exam or the re-exam. The better grade counts.

Sprache English

Lernziele / Kompetenzen

Broad introduction to mathematical methods in image processing and computer vision. The lecture qualifies students for a bachelor thesis in this field. Together with the completion of advanced or specialised lectures (9 credits at least) it is the basis for a master thesis in this field.

Inhalt

Inhalt

1. Basics
 - 1.1 Image Types and Discretisation
 - 1.2 Degradations in Digital Images
2. Colour Perception and Colour Spaces
3. Image Transformations
 - 3.1 Continuous Fourier Transform
 - 3.2 Discrete Fourier Transform
 - 3.3 Image Pyramids
 - 3.4 Wavelet Transform
4. Image Compression
5. Image Interpolation
6. Image Enhancement
 - 6.1 Point Operations

- 6.2 Linear Filtering and Feature Detection
- 6.3 Morphology and Median Filters
- 6.3 Wavelet Shrinkage, Bilateral Filters, NL Means
- 6.5 Diffusion Filtering
- 6.6 Variational Methods
- 6.7 Deconvolution Methods
- 7. Texture Analysis
- 8. Segmentation
 - 8.1 Classical Methods
 - 8.2 Variational Methods
- 9. Image Sequence Analysis
 - 9.1 Local Methods
 - 9.2 Variational Methods
- 10. 3-D Reconstruction
 - 10.1 Camera Geometry
 - 10.2 Stereo
 - 10.3 Shape-from-Shading
- 11. Object Recognition
 - 11.1 Hough Transform
 - 11.2 Invariants
 - 11.3 Eigenspace Methods

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-5	6	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Gerhard Weikum

Dozent/inn/en Prof. Dr. Gerhard Weikum

Zulassungsvoraussetzungen Good knowledge of undergraduate mathematics (linear algebra, probability theory) and basic algorithms.

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutor groups
- Presentation of solutions in tutor groups
- Passing 2 of 3 written tests (after each third of the semester)
- Passing the final exam (at the end of the semester)

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined by the performance in written tests, tutor groups, and the final exam. Details will be announced on the course web site.

Sprache English

Lernziele / Kompetenzen

The lecture teaches models and algorithms that form the basis for search engines and for data mining and data analysis tools.

Inhalt

Information Retrieval (IR) and Data Mining (DM) are methodologies for organizing, searching and analyzing digital contents from the web, social media and enterprises as well as multivariate datasets in these contexts. IR models and algorithms include text indexing, query processing, search result ranking, and information extraction for semantic search. DM models and algorithms include pattern mining, rule mining, classification and recommendation. Both fields build on mathematical foundations from the areas of linear algebra, graph theory, and probability and statistics.

Literaturhinweise

Will be announced on the course web site.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-5	6	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Isabel Valera

Dozent/inn/en Prof. Dr. Isabel Valera

Zulassungsvoraussetzungen The lecture gives a broad introduction into machine learning methods. After the lecture the students should be able to solve and analyze learning problems.

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials.
- 50% of all points of the exercises have to be obtained in order to qualify for the exam.
- Passing 1 out of 2 exams (final, re-exam).

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Determined from the results of the exams, exercises and potential projects. The exact grading modalities are announced at the beginning of the course.

Sprache English

Lernziele / Kompetenzen

The lecture gives a broad introduction into machine learning methods. After the lecture the students should be able to solve and analyze learning problems.

Inhalt

- Bayesian decision theory
- Linear classification and regression
- Kernel methods
- Bayesian learning
- Semi-supervised learning
- Unsupervised learning
- Model selection and evaluation of learning methods
- Statistical learning theory
- Other current research topics

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-5	6	every winter semester	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Dietrich Klakow

Dozent/inn/en Prof. Dr. Dietrich Klakow

Zulassungsvoraussetzungen *Mathematik für Informatiker I - III* or comparable; good programming skills.

Leistungskontrollen / Prüfungen Written Exam

Lehrveranstaltungen / SWS 2 h lectures
+ 2 h tutorial
+ 2 h project work
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Written exam and graded projects. Exact details will be announced in the first lecture.

Sprache English

Lernziele / Kompetenzen

The participants will be introduced to the key ideas of basic classification algorithms and in particular neural networks. A focus is also the implementation and applications to relevant problems. To achieve this, there will be theoretical exercises as well as project work.

Inhalt

- Classification
- Regression
- Linear Classifiers
- Perceptron
- Support Vector Machines
- Multy-Layer Perceptrons
- Deep Learning Software
- Autoencoders
- LSTMs
- Recurrent Neural Networks
- Sequence-to-sequence learning

Literaturhinweise

Ian Goodfellow and Yoshua Bengio and Aaron Courville
Deep Learning
MIT Press, 2016
<http://www.deeplearningbook.org>

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-6	6	unregelmässig	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Henryk Zähle

Dozent/inn/en Dozent/inn/en der Mathematik

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Regelmäßige, aktive Teilnahme an der Vorlesung und an den begleitenden Übungen; Abschlussprüfung

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 60 h Vorlesung
+ 30 h Übungen
+ 180 h Eigenstudium
(Vor- und Nachbereitung, Bearbeitung Übungsaufgaben)
= 270 h (= 9 ECTS)

Modulnote Durch Klausur(en) und mündliche Prüfung. Der Modus wird zu Beginn der Vorlesung bekannt gegeben.

Sprache Deutsch

Lernziele / Kompetenzen

Inhalt

- Statistische Modelle; Modellwahl und Modellüberprüfung
- Punktschätzungen
- Bereichsschätzungen
- Hypothesentests

Literaturhinweise

Bekanntgabe jeweils vor der Vorlesung auf der Vorlesungsseite im Internet

Weitere Informationen

Methoden: Information durch Vorlesung; Vertiefung durch Eigentätigkeit (Nacharbeit, Übungen).

Anmeldung: Bekanntgabe jeweils rechtzeitig vor Semesterbeginn durch Aushang und im Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-6	6	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Michael Backes

Dozent/inn/en Prof. Dr. Michael Backes
Prof. Dr. Cas Cremers

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Passing the final exam
- A re-exam is normally provided (as written or oral examination).

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined by the performance in exams, tutor groups, and practical tasks.
Details will be announced by the lecturer at the beginning of the course.

Sprache English

Lernziele / Kompetenzen

Description, assessment, development and application of security mechanisms, techniques and tools.

Inhalt

- Basic Cryptography,
- Specification and verification of security protocols,
- Security policies: access control, information flow analysis,
- Network security,
- Media security,
- Security engineering

Literaturhinweise

Will be announced on the course website

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-6	6	jedes Sommersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Henryk Zähle
Prof. Dr. Christian Bender

Dozent/inn/en Dozent/inn/en der Mathematik

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Regelmäßige, aktive Teilnahme an der Vorlesung und an den begleitenden Übungen; Abschlussprüfung

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 60 h Vorlesung
+ 30 h Übungen
+ 180 h Eigenstudium
(Vor- und Nachbereitung, Bearbeitung Übungsaufgaben)
= 270 h (= 9 ECTS)

Modulnote Durch Klausur(en) und mündliche Prüfung. Der Modus wird zu Beginn der Vorlesung bekannt gegeben.

Sprache Deutsch

Lernziele / Kompetenzen

Inhalt

- Maß- und Integrationstheorie
- Allgemeine Wahrscheinlichkeitsräume
- Zufallsvariablen und deren Verteilungen
- Bedingen auf Ereignisse
- Unabhängigkeit
- Erwartungswert, Varianz, Kovarianz, Korrelation
- Charakterisieren von Verteilungen auf euklidischen Räumen (Verteilungsfunktion, erzeugende Funktionen)
- Summen unabhängiger Zufallsvariablen
- Konvergenzbegriffe für Folgen von Wahrscheinlichkeitsmaßen und Folgen von Zufallsvariablen
- Grenzwertsätze für Summen unabhängiger reellwertiger Zufallsvariablen (Gesetze der großen Zahlen, zentraler Grenzwertsatz)
- Multivariate Normalverteilung, multivariater zentraler Grenzwertsatz

Literaturhinweise

Bekanntgabe jeweils vor der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Methoden: Information durch Vorlesung; Vertiefung durch Eigentätigkeit (Nacharbeit, Übungen).

Anmeldung: Bekanntgabe jeweils rechtzeitig vor Semesterbeginn durch Aushang und im Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-6	6	jedes Wintersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Henryk Zähle
Prof. Dr. Christian Bender

Dozent/inn/en Dozent/inn/en der Mathematik

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Regelmäßige, aktive Teilnahme an der Vorlesung und an den begleitenden Übungen; Abschlussprüfung

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 60 h Vorlesung
+ 30 h Übungen
+ 180 h Eigenstudium
(Vor- und Nachbereitung, Bearbeitung Übungsaufgaben)
= 270 h (= 9 ECTS)

Modulnote Durch Klausur(en) und mündliche Prüfung. Der Modus wird zu Beginn der Vorlesung bekannt gegeben.

Sprache Deutsch

Lernziele / Kompetenzen

Inhalt

- Bedingen auf Sigma-Algebren
- Grundlagen stochastischer Prozesse
- Poisson-Prozess
- Brown'sche Bewegung
- Martingaleigenschaft
- Markoveigenschaft

Literaturhinweise

Bekanntgabe jeweils vor der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Methoden: Information durch Vorlesung; Vertiefung durch Eigentätigkeit (Nacharbeit, Übungen).

Anmeldung: Bekanntgabe jeweils rechtzeitig vor Semesterbeginn durch Aushang und im Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4-6	6	at least every two years	1 semester	4	6

Modulverantwortliche/r Prof. Dr. Jörg Hoffmann

Dozent/inn/en Prof. Dr. Jörg Hoffmann

Zulassungsvoraussetzungen *Programming 1, Programming 2, Fundamentals of Data Structures and Algorithms, and Elements of Machine Learning* or other courses in machine learning are recommended. The *Artificial Intelligence* core course provides useful background but is not necessary.

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Solving of weekly assignments
- Passing the final written exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 2 h lectures
+ 2 h tutorial
= 4 h (weekly)

Arbeitsaufwand 60 h of classes
+ 120 h private study
= 180 h (= 6 ECTS)

Modulnote Will be determined from the performance in exams. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Knowledge about methods for learning, verifying and testing action policies in AI Planning; understanding of algorithmic techniques enabling these methods.

Inhalt

- Introduction to basic AI concepts needed in the course
- Partial-order reduction
- Dominance pruning
- SAT-based planning
- ASNet action policies
- Safety verification of neural action policies, basic methods
- Safety verification of neural action policies: policy predicate abstraction
- Testing methods for learned action policies, deterministic and probabilistic settings

Literaturhinweise

There is no text book covering the course topics. Links to relevant publications and other material where available will be provided on the slides

Weitere Informationen

This module was formerly also known as *AI Planning*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
5	6	jedes Semester	1 Semester	2	7

Modulverantwortliche/r Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

Dozent/inn/en Dozent/inn/en der Fachrichtung

Zulassungsvoraussetzungen Grundlegende Kenntnisse im jeweiligen Teilbereich des Studienganges.

Leistungskontrollen / Prüfungen

- Thematischer Vortrag mit anschließender Diskussion
- Aktive Teilnahme an der Diskussion
- Gegebenenfalls schriftliche Ausarbeitung oder Projekt

Lehrveranstaltungen / SWS 2 SWS Seminar

Arbeitsaufwand 30 h Präsenzstudium
+ 180 h Eigenstudium
= 210 h (= 7 ECTS)

Modulnote Wird aus den Leistungen im Vortrag und der schriftlichen Ausarbeitung und/oder dem Seminarprojekt ermittelt. Die genauen Modalitäten werden von dem/der jeweiligen Dozenten/in bekannt gegeben.

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

Die Studierenden haben am Ende der Veranstaltung vor allem ein tiefes Verständnis aktueller oder fundamentaler Aspekte eines spezifischen Teilbereiches der Informatik erlangt.

Sie haben weitere Kompetenz im eigenständigen wissenschaftlichen Recherchieren, Einordnen, Zusammenfassen, Diskutieren, Kritisieren und Präsentieren von wissenschaftlichen Erkenntnissen gewonnen.

Inhalt

Weitgehend selbstständiges Erarbeiten des Seminarthemas:

- Lesen und Verstehen wissenschaftlicher Arbeiten
- Analyse und Bewertung wissenschaftlicher Aufsätze
- Diskutieren der Arbeiten in der Gruppe
- Analysieren, Zusammenfassen und Wiedergeben des spezifischen Themas
- Erarbeiten gemeinsamer Standards für wissenschaftliches Arbeit
- Präsentationstechnik

Spezifische Vertiefung in Bezug auf das individuelle Thema des Seminars.

Der typische Ablauf eines Seminars ist üblicherweise wie folgt:

- Vorbereitende Gespräche zur Themenauswahl
- Regelmäßige Treffen mit Diskussion ausgewählter Beiträge
- ggf. Bearbeitung eines themenbegleitenden Projekts
- Vortrag und ggf. Ausarbeitung zu einem der Beiträge

Literaturhinweise

Material wird dem Thema entsprechend ausgewählt.

Weitere Informationen

Die jeweils zur Verfügung stehenden Seminare werden vor Beginn des Semesters angekündigt und unterscheiden sich je nach Studiengang.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4	6	jedes Sommersemester	1 Semester	2	9

Modulverantwortliche/r Prof. Dr. Jens Dittrich

Dozent/inn/en Prof. Dr. Jens Dittrich
Dozent/inn/en der Fachrichtung

Zulassungsvoraussetzungen *Elements of Data Science and Artificial Intelligence, Elements of Machine Learning, Statistics Lab, Programming 1 und 2*

Leistungskontrollen / Prüfungen

- regelmäßige (mindestens jede zweite Woche) Treffen mit AnwenderInnen und BetreuerInnen,
- Abschlussvortrag am Ende des Semesters,
- Abgabe des erstellten Projektes inklusive Source Code

Lehrveranstaltungen / SWS 2 SWS Seminar

Arbeitsaufwand 30 h Meetings mit DSAI-AnwenderInnen und BetreuerInnen
+ 249 h Eigenstudium
= 270 h (= 9 ECTS)

Modulnote unbenotet

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

- selbständiges Bearbeiten eines kleine DSAI-Projektes
- selbständiges Präsentation von Projektergebnissen
- Arbeit im Team
- Arbeit mit AuftraggeberInnen

Inhalt

In diesem Projektseminar sollen Studierende im Team von 2-3 Studierenden die DSAI-Aufgabenstellung eines/r DSAI-AnwenderIn bearbeiten.

Literaturhinweise

Material wird dem Thema entsprechend ausgewählt.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
6	6	jedes Semester	variabel	2	9

Modulverantwortliche/r Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

Dozent/inn/en Dozent/inn/en der Fachrichtung

Zulassungsvoraussetzungen Erwerb von mindestens 120 CP

Leistungskontrollen / Prüfungen

- Schriftliche Ausarbeitung der Aufgabenstellung der Bachelorarbeit und der relevanten wissenschaftlichen Literatur
- Vortrag über die geplante Aufgabenstellung mit anschließender Diskussion
- Aktive Teilnahme an der Diskussion

Lehrveranstaltungen / SWS 2 SWS Seminar

Arbeitsaufwand

- 30 h Präsenzstudium (Seminar)
- + 30 h Betreuung durch den Lehrstuhl
- + 210 h Eigenstudium
- = 270 h (= 9 ECTS)

Modulnote Wird aus den Leistungen im Vortrag und der schriftlichen Ausarbeitung ermittelt. Die genauen Modalitäten werden von dem/der jeweiligen Dozenten/in bekannt gegeben.

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

Im Bachelorseminar erwirbt der Studierende unter Anleitung die Fähigkeit zum wissenschaftlichen Arbeiten im Kontext eines angemessenen Themengebietes.

Am Ende des Bachelorseminars sind die Grundlagen für eine erfolgreiche Anfertigung der Bachelorarbeit gelegt, und wesentliche Lösungsansätze bereits eruiert.

Das Bachelorseminar bereitet somit die Themenstellung und Ausführung der Bachelorarbeit vor.

Es vermittelt darüber hinaus praktische Fähigkeiten des wissenschaftlichen Diskurses. Diese Fähigkeiten werden durch die aktive Teilnahme an einem Lesekreis vermittelt, in welchem die Auseinandersetzung mit wissenschaftlich anspruchsvollen Themen geübt wird.

Inhalt

Einarbeitung in ein wissenschaftliches Themengebiet innerhalb der Informatik.

Anfertigung einer schriftlichen Ausarbeitung der Aufgabenstellung der Bachelorarbeit und der relevanten wissenschaftlichen Literatur.

Fachvortrag über das Themengebiet und die geplante Aufgabenstellung der Bachelorarbeit.

Das Thema wird in enger Absprache mit dem anleitenden Dozenten definiert.

Literaturhinweise

Dem Themengebiet entsprechende wissenschaftliche Artikel in enger Absprache mit dem Dozenten

Bachelor-Arbeit

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
6	6	jedes Semester	3 Monate	-	12

Modulverantwortliche/r Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

Dozent/inn/en Professoren der Fachrichtung

Zulassungsvoraussetzungen Erfolgreicher Abschluss des *Bachelor-Seminars*

Leistungskontrollen / Prüfungen Schriftliche Ausarbeitung. Sie beschreibt sowohl das Ergebnis der Arbeit als auch den Weg, der zu dem Ergebnis führte. Der eigene Anteil an den Ergebnissen muss klar erkennbar sein. Außerdem Präsentation der Bachelorarbeit in einem Kolloquium, in dem auch die Eigenständigkeit der Leistung des Studierenden überprüft wird.

Lehrveranstaltungen / SWS keine

Arbeitsaufwand 30 h Betreuung durch den Lehrstuhl
+ 330 h Eigenstudium
= 360 h (= 12 ECTS)

Modulnote Beurteilung der Bachelorarbeit durch die Gutachter.

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

Die Bachelor-Arbeit ist eine Projektarbeit, die unter Anleitung ausgeführt wird. Sie soll der Kandidaten/die Kandidatin in der Lage versetzen, innerhalb einer vorgegebenen Frist ein Problem aus dem Gebiet der Informatik selbständig zu lösen und die Ergebnisse in wissenschaftlich angemessener Form zu dokumentieren.

Inhalt

Bearbeitung einer aktuellen Problemstellung aus der Informatik unter Anleitung. Adäquate Dokumentation der Ergebnisse in Form einer wissenschaftlichen Abschlussarbeit.

Das Thema wird in enger Absprache mit dem anleitenden Dozenten definiert.

Literaturhinweise

Dem Themengebiet entsprechende wissenschaftliche Artikel in enger Absprache mit dem anleitenden Dozenten.