Department of Computer Science

Chair
• Michael Franklin

Professors
• Yali Amit
• Laszlo Babai
• Andrew Chien
• Frederic Chong
• Todd Dupont
• Ian Foster
• Michael Franklin
• John Goldsmith
• Robert Grossman
• Stuart A. Kurtz
• Ketan Mulmuley
• Alexander Razborov
• John Reppy
• Janos Simon
• Rick L. Stevens
• Rebecca Willett
• Ben Zhao
• Heather Zheng

Associate Professors
• David Cash
• Henry Hoffmann
• Gordon Kindlmann
• Risi Kondor
• Shan Lu
• Anne Rogers

Assistant Professors
• Ravi Chugh
• Andrew Drucker
• Aaron Elmore
• Haryadi Gunawi
• Jun Chen Jiang
• Sanjay Krishnan
• Yanjing Li
• Pedro Lopes
• Michael Maire
• Aaron Potechin
• Blase Eric Ur

Research Associate Professors
• Diana Franklin
• Rayid Ghani

Clinical faculty
• Andrew Binkowski (assistant clinical professor)
• Geraldine Brady (associate clinical professor)
• Amitabh Chaudhary (associate clinical professor)
The Department of Computer Science is dedicated to advancing and improving the knowledge, understanding, and practice of computer science through basic research and education.

Research

We construe the field of computer science broadly to include the complementary concepts of computation, information, and communication. We employ modes of inquiry and creation from pure mathematics to experiment and observation to design and engineering. We investigate computation, information, and communication as inherently interesting phenomena; we also investigate the many ways in which computational concepts engage other topics: computational tools for science and scholarship, computational infrastructure for society.

There is an ongoing major thrust to expand the role of Computer Science and computation at the University, with considerable expansion of the faculty, and expanded support to explore new research areas. Accordingly, the descriptions below, a snapshot of our current active research, is likely to expand.

Current active research areas include computing systems, computer architecture, computer security and privacy, error-tolerant computing and error recovery in computing systems, databases and data intensive computing, theoretical computer science, discrete mathematics, quantum computing, programming languages, machine learning, computational linguistics, computer vision, cloud computing, sustainable computing, scientific computing and visualization, high performance computing, human-computer interaction, computer science education, and interdisciplinary research in computing in the physical, biological, and social sciences.

Artificial intelligence

Research spans the spectrum from foundational work in statistical machine learning to computer vision and computational linguistics. The AI group has strong ties to CAMI, the University’s Computational and Applied Mathematics Initiative.

Computational Mathematics

Our faculty and students study the foundations of simulation technology. This includes the development and mathematical analysis of numerical algorithms for approximating partial differential equations. We also study language and systems aspects of numerical computing. Parallel and high performance computing are an integral part of our efforts.

Systems

Our faculty advance principles and understanding of a broad range of areas, including systems and networking, programming languages and software engineering, software and hardware architecture, data-intensive computing and databases, graphics and visualization, computer security, human-computer interactions, and systems biology. Particular areas of focus include formal definition, design, and implementation of programming languages, data-intensive computing systems and algorithms, large scale distributed and collaborative systems, heterogeneous computer architectures, reliable computing systems, self-tuning systems, and emerging technologies.

Theoretical computer science

We investigate the fundamental concepts underlying computation using and developing mathematical techniques, as well as topics in discrete mathematics. Our faculty specialize in complexity theory, algorithms, discrete mathematics, and combinatorics.

These efforts are enhanced by strong connections to the Center for Data and Applied Computing, an incubator for campus wide initiatives in Data Science, broadly construed, the James Franck Institute, which focuses on condensed matter physics; the Institute for Biophysical Dynamics, which provides a forum for studying questions that arise at the boundary between the biological and physical sciences; and the Institute for Molecular Engineering. In addition, we have collaborations with faculty in academic departments, including the geophysical sciences, linguistics, mathematics, physics, psychology, and statistics, and well as with the Division of Mathematics and Computer Science at Argonne National Laboratory (ANL), which is operated by the University of Chicago for the US Department of Energy. We also have almost seamless collaborations with the Toyota Technological Institute on campus, especially in the areas of Theoretical Computer Science and Machine Learning.

Graduate Programs

We offer two graduate curricula in computer science.

1. A graduate professional curriculum leading to the Master of Science (MS) degree, for students who wish to enter or advance themselves in computer science practice.
2. A graduate research curriculum leading to the PhD degree that prepares students to perform advanced basic research in computer science either in industry or academia. Teaching experience is available for students preparing for academic careers.

Acquire further information about our Masters Program in Computer Science (MPCS) through the MPCS website (http://masters.cs.uchicago.edu), by writing to our MPCS Admissions, Department of Computer Science, University of Chicago, 1100 East 58th Street, Chicago, IL 60637, or by telephoning 773.834.3388. You may also email any questions to our questions@cs.uchicago.edu email address.

Acquire further information about our PhD program through our PhD admissions website (http://csphd.sites.uchicago.edu/page/admission-phd-program), by writing to Admissions, Department of Computer Science, University of Chicago, 1100 East 58th Street, Chicago, IL 60637, or by telephoning 773.702.6011.

General information about our department is available from the departmental website (https://computerscience.uchicago.edu).

The PhD Program

The department offers two PhD tracks: a standard track and a computational mathematics track.

The detailed requirements for the PhD degree and for the MS degree within the PhD program can be found by visiting the Department’s web page (https://computerscience.uchicago.edu). Here is a brief summary:

To obtain an MS degree within the PhD program, students in the PhD program must fulfill the following requirements:

• Course requirements. Five core courses and four electives. The core courses include two in Theory, two in Systems, and one in Machine Learning. Please refer to the web page for details regarding the core courses.

A modified set of core courses applies to the computational mathematics track (see the website). The list of electives is frequently updated; we refer you to the web page.

Students must complete the course requirements by the end of their second year of study. To receive an MS degree within the PhD program, students must receive a grade of at least B in all the nine courses and have a GPA of at least 3.00 in the five core courses, and write a Master’s paper and pass a Master’s examination.

To obtain a PhD degree, students must meet enhanced MS requirements: they must do especially well in the five core courses with a 3.25 average. Details about the requirements can be found in the departmental web page. In addition, students should:

• Pass a Candidacy Exam
• Write and defend a Doctoral Thesis that contains significant original research in computer science.

Teaching Opportunities for Students in the PhD Program

The department takes its undergraduate teaching responsibilities very seriously, and offers supervised teaching opportunities, including lecturing, acting as teaching assistants, and working as lab assistants to its best graduate students.

Computing Facilities

In addition to the general University computing facilities including the Research Computing Center (https://rcc.uchicago.edu/resources) and access to high performance computers at ANL, and our Computer Science Instructional Laboratory (which contains about 50 Macintosh computers and 40 desktops running Linux), the Ryerson Research Computing Service provides the faculty, students, and postdoctoral associates in computer science with computing resources. We have the flexibility to adapt quickly to new research needs.

The resources include: 24 hour 7 day interactive computing on a number of shared computing servers as well as individually assigned desktops. These servers and desktops run the Linux operating system and are interconnected via high speed Ethernet. These systems are supported by substantial amounts of both local and networked disk storage for individual and group use which are backed up regularly. Linux servers are available for general instructional and research purposes as well as hardware and virtual machines which are adapted to specialized needs.

Courses

For the list of courses offered and the course descriptions, please consult the courses section of the departmental web page (https://computerscience.uchicago.edu/graduate/courses).
Computer Science Courses

CMSC 30100. Technical Writing and Presentation. 100 Units.
Clear, logical writing and presentations are foundational skills for computer scientists. This class is meant to introduce computer science students to basic ideas and techniques for effective communication in both writing and presentations. The class will include several complementary components, including critical analysis of technical papers, weekly writing assignments focusing on writing style, clarity, and logical flow, and discussions of style for different research areas and venues. Later weeks will focus on skills for effective technical presentations in different settings, e.g. conference presentations, job talks, and keynotes. The course is primarily targeted towards graduate students, although undergraduates can audit the class (or enroll with permission from the instructor).
Instructor(s): Ben Zhao Terms Offered: Autumn
Prerequisite(s): None

CMSC 31010. Mathematical Foundations. 100 Units.
This course is an introduction to formal tools and techniques which can be used to better understand linguistic phenomena. A major goal of this course is to enable students to formalize and evaluate theoretical claims.
Equivalent Course(s): LING 31010, CMSC 21010, LING 21010

CMSC 31140. Computational Imaging: Theory and Methods. 100 Units.
Computational imaging refers to the process of forming images from data where computation plays an integral role. This course will cover basic principles of computational imaging, including image denoising, regularization techniques, linear inverse problems and optimization-based solvers, and data acquisition models associated with tomography and interferometry. Specific topics may include patch-based denoising, sparse coding, total variation, dictionary learning, computational photography, compressive imaging, inpainting, and deep learning for image reconstruction.
Instructor(s): R. Willett Terms Offered: Spring
Equivalent Course(s): CAAM 31140, STAT 31140

CMSC 31150. Mathematical Toolkit. 100 Units.
Introduction to mathematical techniques of linear algebra and probability used in different areas of computer science. Topics include Linear Algebra (Hilbert spaces, eigenvalues and eigenvectors, SVD, least squares), discrete probability, Gaussian variables, concentration inequalities and dimension reduction, Linear Programming and LP duality. Time permitting, martingales, stochastic processes.
Instructor(s): Tulsiani Terms Offered: Autumn
Equivalent Course(s): TTIC 31150

CMSC 31230. Fundamentals of Deep Learning. 100 Units.
Introduction to fundamental principles of deep learning. Although deep learning systems are evolving rapidly, this course attempts to teach material that will remain relevant and useful as the field changes. The course will emphasize theoretical and intuitive understanding to the extent possible. Expected outcomes: Ability to design and train novel deep learning architectures. An understanding of the general issues and phenomenon sufficient to guide architecture design and training.
Instructor(s): David McAllester Terms Offered: Winter
Prerequisite(s): Introduction to machine learning
Equivalent Course(s): TTIC 31230

CMSC 32001. Topics in Programming Languages. 100 Units.
This course covers a selection of advanced topics in programming languages. Terms Offered: Autumn,Winter,Spring
Prerequisite(s): Consent of department counselor and instructor

CMSC 32000. Computer Architecture. 100 Units.
This course is a survey of contemporary computer organization covering CPU design, instruction sets, control, processors, busses, ALU, memory, pipelined computers, multiprocessors, networking, and case studies. We focus on the techniques of quantitative analysis and evaluation of modern computing systems, such as the selection of appropriate benchmarks to reveal and compare the performance of alternative design choices in system design. We emphasize major component subsystems of high-performance computers: pipelining, instruction-level parallelism, memory hierarchies, input/output, and network-oriented interconnections.
Instructor(s): Hoffmann Terms Offered: Autumn

CMSC 32201. Topics in Computer Architecture. 100 Units.
This course covers a selection of advanced topics in computer architecture. Terms Offered: Autumn,Winter,Spring
Prerequisite(s): Consent of department counselor and instructor

CMSC 32250. Intro to Computer Security. 100 Units.
This course introduces the principles and practice of computer security. It aims to teach how to model threats to computer systems and how to think like a potential attacker. It presents standard cryptographic functions and protocols and gives an overview of threats and defenses for software, host systems, networks, and the Web. It also touches on some of the legal, policy, and ethical issues surrounding computer security in areas such as privacy, surveillance, and the disclosure of security vulnerabilities. The goal of this course is to provide a foundation for further study in computer security and to help better understand how to design, build, and use computer systems more securely.
CMSC 33000. Operating Systems. 100 Units.

CMSC 33001. Topics in Systems. 100 Units.
This course covers a selection of advanced topics in computer systems.
Terms Offered: Autumn, Spring, Winter
Prerequisite(s): Consent of department counselor and instructor

CMSC 33100. Advanced Operating Systems. 100 Units.
This course covers advanced topics in operating systems and systems research. Possible topics include, but are not limited to the following: OS philosophies, networked operating systems, distributed file systems, virtual machines, fault-tolerant systems, resource allocation, parallel computing and multiprocessing, cloud computing, and security.
Instructor(s): Lu Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor

CMSC 33200. Topics: Operating Systems. 100 Units.

CMSC 33210. Usable Security and Privacy. 100 Units.
Questions of usability and privacy in computer systems, including human factors.
Instructor(s): Ur Terms Offered: Spring
Prerequisite(s): Consent of department counselor and instructor

CMSC 33231. Topics in Human Computer Interaction. 100 Units.
Graduate study of topics in Human Computer Interaction (HCI). In this class, we critically examine hot topics that might impact the future generations of computing interfaces; these include: physiological I/O (e.g., brain and muscle computer interfaces), tangible computing (giving shape and form to interfaces), wearable computing (I/O devices closer to the user’s body), rendering new realities (e.g., virtual & augmented reality), haptics (giving computers the ability to generate touch and forces) and unusual auditory interfaces (e.g., silent speech & microphones as sensors). In this class you will: (1) learn about these new developments during the lectures, (2) read HCI papers and summarize these in short weekly assignments, and lastly, (3) start inventing the future of computing interfaces by proposing a new idea in the form of a paper abstract, which you will present at the end of the semester and have it peer-reviewed in class by your classmates.
Instructor(s): Pedro Lopes Terms Offered: Winter

CMSC 33250. Introduction to Computer Security. 100 Units.
This course introduces the principles and practice of computer security. It aims to teach how to model threats to computer systems and how to think like a potential attacker. It presents standard cryptographic functions and protocols and gives an overview of threats and defenses for software, host systems, networks, and the Web. It also touches on some of the legal, policy, and ethical issues surrounding computer security in areas such as privacy, surveillance, and the disclosure of security vulnerabilities. The goal of this course is to provide a foundation for further study in computer security and to help better understand how to design, build, and use computer systems more securely.
Instructor(s): A. Feldman Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor

CMSC 33251. Topics in Computer Security. 100 Units.
Seminar on current topics in computer security.

CMSC 33300. Networks and Distributed Systems. 100 Units.
This course will focus on studying the state of the art in networking and networked systems, from a research and design perspective. We will cover a variety of topics from routing protocols to Internet stability, peer-to-peer, social networks and networking for data centers. Coverage of each topic will dive into fundamental design questions of protocols and systems, including updates from results of currently active research. Readings will focus on classic and current research publications, and students are expected to come in with a solid background on networking basics. Students will learn tools, techniques, and concepts while learning to carry out original research in an open-ended course project, with the end goal of producing real, publishable results by the end of the quarter. Students are also expected to gain experience in two skills: quickly reading technical papers (without sacrificing understanding), and giving clear and well-organized presentations.
Instructor(s): B. Sotomayor Terms Offered: Winter

CMSC 33400. Mobile Computing. 100 Units.
Mobile computing is pervasive and changing nearly every aspect of society. Sensing, actuation, and mediation capabilities of mobile devices are transforming all aspects of computing: uses, networking, interface, form, etc. This course explores new technologies driving mobile computing and their implications for systems and society. Current focus areas include expanded visual experience with computational photography, video and interactive augmented reality, and synchronicity and proximity-detection to enable shared social experiences. Labs expose students to software and hardware capabilities of mobile computing systems, and develop the capability to envision radical new applications for a large-scale course project.
Instructor(s): A. Chien Terms Offered: Not offered 2017-2018.
Prerequisite(s): CMSC 23000 or 23300 or equivalent are required.

CMSC 33501. Topics in Databases. 100 Units.
This course covers a selection of advanced topics in database systems.
Terms Offered: Autumn, Winter, Spring
Prerequisite(s): Consent of department counselor and instructor
CMSC 33520. Data Intensive Computer Systems. 100 Units.
Big Data and Data Analytics have become hot topics as well as drivers of multi-billion dollar industries. With unprecedented data collection from e-commerce, the WWW, scientific instruments, mobile phones, and IoT. The course objective is to expose students to the technical challenges of data-intensive computing systems, including canonical driving problems, research systems, and emerging technologies. While other classes focus on analysis algorithms (or even underlying statistical or machine learning methods), we focus on the computer systems and technology needed to achieve scalable and efficient data-intensive computing systems. Through paper reading, discussions, presentation, and in-depth projects, students will develop a broad familiarity with current challenges and hands-on experience with a range of systems which together provide a solid preparation for research in the area. Course topics include: parallel filesystems, SQL databases, NoSQL/Mapreduce systems, storage class memories (from Flash to Memristor to ReRAM), and popular open source infrastructures such as Spark, Hadoop, VoltDB, Cassandra, Memcached, MongoDB, and others.
Instructor(s): Chien Terms Offered: Spring
Prerequisite(s): Consent of department counselor and instructor

CMSC 33550. Introduction to Databases. 100 Units.
This course is an introduction to database design and programming using the relational model. Topics include DBMS architecture, entity-relationship and relational models, relational algebra, relational calculus, functional dependencies and normal forms, web DBs and PHP, query optimization, and physical data organization. The lab section will guide students through the collaborative implementation of a relational database management system, allowing students to see topics such as physical data organization and DBMS architecture in practice, and exercise general skills such as collaborative software development.
Instructor(s): Elmore Terms Offered: Winter
Prerequisite(s): Consent of department counselor

CMSC 33600. Type Systems for Programming Languages. 100 Units.
This course covers the basic ideas of type systems, their formal properties, their role in programming language design, and their implementation. Exercises involving design and implementation explore the various options and issues.
Terms Offered: Winter
Prerequisite(s): Consent of department counselor
Note(s): CMSC 22100 recommended. Not offered in 2016-17.

CMSC 33700. Computer Graphics. 100 Units.
This course introduces the basic concepts and techniques used in three-dimensional computer graphics. The focus is on real-time rendering techniques, such as those found in computer games. These include coordinate systems and transformations, the graphics pipeline, basic geometric algorithms, texture mapping, level-of-detail optimizations, and shadows. Students are required to complete both written assignments and programming projects using OpenGL.
Instructor(s): J. Reppy Terms Offered: TBD
Prerequisite(s): Consent of department counselor and instructor

CMSC 33710. Scientific Visualization. 100 Units.
Scientific visualization combines computer graphics, numerical methods, and mathematical models of the physical world to create a visual framework for understanding and solving scientific problems. The mathematical and algorithmic foundations of scientific visualization (for example, scalar, vector, and tensor fields) will be explained in the context of real-world data from scientific and biomedical domains. The course is also intended for students outside computer science who are experienced with programming and computing with scientific data. Programming projects will be in C and C++.
Instructor(s): G. Kindlmann Terms Offered: Winter
Prerequisite(s): Consent of department counselor and instructor
Note(s): Strong programming skills and basic knowledge of linear algebra and calculus. This course is offered in alternate years.

CMSC 33750. Machine Learning and Cancer. 100 Units.
In this topics course we will investigate the use of machine learning methods in the study of Cancer and the development of precision oncology. Cancer is a complex disease that impacts millions each year. Recently the concept of precision oncology has gained popularity as an approach to customize Cancer treatments based on the genomic profile and history of the patient, the molecular properties of the patient's tumor and the action and mode of treatments that are available. At the center of any precision medicine approach are large-scale datasets from which predictive models can be built, scalable analysis methods for processing and integrating data and machine learning methods for constructing and evaluating predictive models that can be used in diagnosis, treatment planning, and outcome prediction for patient care. In this course we will work through the development of the entire pipeline from raw data to predictive models. We will develop and evaluate predictive models for drug response, tumor typing, image based diagnosis, and treatment outcomes. We will also develop some population based models that include environmental factors. Students will work through key papers, representative datasets and a variety of machine learning methods including some deep learning models under development in the joint DOE/NCI Cancer project. Familiarity with python and machine learning will be helpful. Students will have an opportunity to do significant project work as part of the course.
Instructor(s): Rick Stevens, Robert Grossman Terms Offered: Autumn

CMSC 33900. Data Visualization. 100 Units.
Data visualizations provide a visual setting in which to explore, understand, and explain datasets. This class describes mathematical and perceptual principles, methods, and applications of "data visualization" (as it is popularly understood to refer primarily to tabulated data). A range of data types and visual encodings will be presented and evaluated. Visualizations will be primarily web-based, using D3.js, and possibly other higher-level languages and libraries.
CMSC 34200. Numerical Hydrodynamics. 100 Units.
This course covers numerical methods for the solution of fluid flow problems. We also make a theoretical evaluation of the methods and experimental study based on the opinionated book Fundamentals of Computational Fluid Dynamics by Patrick J. Roache.
Instructor(s): T. Dupont Terms Offered: Winter
Prerequisite(s): Consent of department counselor. Ability to program; and familiarity with elementary numerical methods and modeling physical systems by systems of differential equations
Note(s): Not offered in 2016-17. Offered in alternate years.

CMSC 34702. Topics in Networks: 100 Units.

CMSC 34703. Topics in Distributed Systems. 100 Units.
Topics in Distributed Systems
Instructor(s): Junchen Jiang Terms Offered: Autumn

CMSC 34900. Topics in Scientific Computing. 100 Units.
This course covers a selection of advanced topics in Scientific Computing.
Instructor(s): Scott Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor
Equivalent Course(s): MAAD 24850

CMSC 34901. Special Topics in Operations Mgt./Mgt. Science. 100 Units.
Course Search (https://intranet.chicagobooth.edu/pub/coursesearch/coursesearch)
Equivalent Course(s): BUSN 40901

CMSC 35000. Introduction to Artificial Intelligence. 100 Units.
This course introduces the theoretical, technical, and philosophical aspects of Artificial Intelligence. We emphasize computational and mathematical modes of inquiry into the structure and function of intelligent systems. Topics include learning and inference, speech and language, vision and robotics, and reasoning and search.

CMSC 35050. Computational Linguistics. 100 Units.
This course introduces the problems of computational linguistics and the techniques used to deal with them, focusing primarily on probabilistic models and techniques. Topics are drawn primarily from phonology, morphology, and syntax. Special topics include automatic learning of grammatical structure and the treatment of languages other than English.
Instructor(s): J. Goldsmith Terms Offered: Spring
Prerequisite(s): CMSC 12200, 15200 or 16200, or by consent
Equivalent Course(s): LING 38600, DIGS 30013

CMSC 35100. Natural Language Processing. 100 Units.
This course introduces the theory and practice of natural language processing, with applications to both text and speech. Topics include regular expressions, finite state automata, morphology, part of speech tagging, context free grammars, parsing, semantics, discourse, and dialogue. Symbolic and probabilistic models are presented. Techniques for automatic acquisition of linguistic knowledge are emphasized.

CMSC 35110. Speech Technologies. 100 Units.
This course will introduce techniques used in speech technologies, mainly focusing on speech recognition. Speech recognition is one of the oldest and most complex structured sequence prediction tasks receiving significant research and commercial attention, and therefore provides a good case study for many of the techniques that are used in other areas of artificial intelligence involving sequence modeling. It is also a good example of the effectiveness of combining statistics and learning with domain knowledge. The course will include practical homework exercises using Matlab and speech toolkits. Expected outcomes: Understand and apply tools for analyzing speech time series such as Fourier analysis and dynamic time warping. Understand and apply hidden Markov models, Gaussian mixtures, and the EM algorithm for speech problems. Understand and apply n-gram language models, smoothing techniques, and their application to speech recognition. Understand generative and discriminative structured prediction approaches for speech problems.
Equivalent Course(s): TTIC 31110
CMSC 35200. Deep Learning Systems. 100 Units.
Deep learning is emerging as a major technique for solving problems in a variety of fields, including computer vision, personalized medicine, autonomous vehicles, and natural language processing. Critical to success in these target domains is the development of learning systems: deep learning frameworks that support the tasks of learning complex models and inferring with those models, and targeting heterogeneous computing devices. This course is aimed as an introduction to this topic. We will cover various aspects of deep learning systems, including: basics of deep learning, programming models for expressing machine learning models, automatic differentiation methods used to compute gradients for training, memory optimization, scheduling, data and model parallel and distributed learning, hardware acceleration, domain specific languages, workflows for large-scale machine learning including hyper parameter optimization and uncertainty quantification, and training data and model serving. The goal is to present a comprehensive picture of how current deep learning systems work, discuss and explore research opportunities, for extending and building on existing frameworks, and deep dive into the accelerators being developed by numerous startups to address the needs of the machine learning community. A typical week will contain one lecture on a specific aspect of deep learning systems and one lab session exploring technologies such as Keras, Tensorflow, CNTK, Mxnet, and PyTorch.
Instructor(s): Ian Foster, Rick Stevens Terms Offered: Autumn
Note(s): This course will provide useful background for students wishing to take our Spring 2019 class on Neuromorphic Computing.

CMSC 35230. Applications of Machine Learning in Large-Scale Computing Systems. 100 Units.
Recent research has demonstrated the feasibility of replacing the traditional heuristics used in computer systems with ones learned from data in areas such as scheduling, data structure design, query optimization, compilers, and control of warehouse scale computing systems. This seminar overviews this recent research trend and studies the characteristics of successful research in this space. The objective is to understand whether this trend will have a long-term impact on the design and implementation of large-scale computing systems. The seminar hopes to highlight the remaining bottlenecks to practical acceptance of machine learning in computing infrastructure and to inspire future systems and learning research.
Terms Offered: Winter
Prerequisite(s): Good working knowledge of computer systems, especially in the areas of operating systems and databases.

CMSC 35246. Deep Learning. 100 Units.
Deep Neural Networks are remarkably effective in large scale learning problems, especially in speech recognition and computer vision. This course aims to cover the basics of Deep Learning, some of the underlying theory, and specific architectures, including Convolutional Neural Networks, Recurrent Neural Networks and the Long Short Term Memory Networks.

CMSC 35200. Mathematical Foundations of Machine Learning. 100 Units.
This course is an introduction to the mathematical foundations of machine learning that focuses on matrix methods and features real-world applications ranging from classification and clustering to denoising and data analysis. Mathematical topics covered include linear equations, regression, regularization, the singular value decomposition, and iterative algorithms. Machine learning topics include the lasso, support vector machines, kernel methods, clustering, dictionary learning, neural networks, and deep learning. Students are expected to have taken calculus and have exposure to numerical computing (e.g. Matlab, Python, Julia, R). Appropriate for graduate students or advanced undergraduates.
Instructor(s): Rebecca Willett Terms Offered: Winter
Prerequisite(s): Competence programming in Matlab, Julia, R, Python, or an equivalent system; knowledge of calculus, mathematical maturity.

CMSC 35300. Neuromorphic Computing. 100 Units.
The human brain consumes around 20 watts-less energy than a light bulb but can perform tasks, such as understanding natural language and interpreting images, that tax megawatt-scale supercomputers. Thus, we may wonder: can we achieve energy efficiencies similar to those of the human brain by building analog electronic circuits that mimic the neuro-biological architectures found in animal nervous systems? This concept, named neuromorphic computing, has become increasingly popular as the energy demands of conventional computers increase. Research in this area is producing a great variety of new computational architectures, microelectronics concepts, algorithmic approaches, and even neuroscience insights. Our goal in this course is to introduce students to the state of knowledge in neuromorphic computing and thus to prepare them to undertake original research in this area. The class will be organized primarily around reading, presenting, and discussing research papers. Topics to be covered include: Neuromorphic concepts Theory and algorithms Microelectronics and devices Programming models and environments Applications: Machine learning, deep learning, robot control, ... Platforms: BrainScaleS, Loihi, SpiNNaker Neuroscience
Terms Offered: Spring TBD

CMSC 35400. Machine Learning. 100 Units.
This course provides hands-on experience with a range of contemporary machine learning algorithms, as well as an introduction to the theoretical aspects of the subject. Topics covered include: the PAC framework, Bayesian learning, graphical models, clustering, dimensionality reduction, kernel methods including SVMs, matrix completion, neural networks, and an introduction to statistical learning theory.
Terms Offered: Spring
Prerequisite(s): Consent of instructor
Equivalent Course(s): STAT 37710, CAAM 37710
CMSC 35401. Topics in Machine Learning: Applied Machine Learning. 100 Units.
We will cover recent research in applied machine learning and systems, particularly with respect to questions of robustness and resilience against attacks on deep learning systems. Students will present papers, lead discussion, and run open ended projects related to the topic.
Terms Offered: TBD Winter

CMSC 35410. Spectral Methods for Machine Learning and Network Analysis. 100 Units.
An introduction to spectral algorithms, emphasizing their power to tackle practical problems in the analysis of networks and high-dimensional data. Topics include spectral graph theory, random walks over networks and their convergence, spectral clustering, subspace projections and embeddings, and numerical algorithms for fundamental linear-algebraic problems.
Instructor(s): Lorenzo Orecchia Terms Offered: Autumn

CMSC 35425. Topics in Statistical Machine Learning. 100 Units.
Topics in Statistical Machine Learning” is a second graduate level course in machine learning, assuming students have had previous exposure to machine learning and statistical theory. The emphasis of the course is on statistical methodology, learning theory, and algorithms for large-scale, high dimensional data. The selection of topics is influenced by recent research results, and students can take the course in more than one quarter.
Terms Offered: To be determined
Equivalent Course(s): STAT 37790

CMSC 35470. Mathematical Computation IIA: Convex Optimization. 100 Units.
The course will cover techniques in unconstrained and constrained convex optimization and a practical introduction to convex duality. The course will focus on (1) formulating and understanding convex optimization problems and studying their properties; (2) understanding and using the dual; and (3) presenting and understanding optimization approaches, including interior point methods and first order methods for non-smooth problems. Examples will be mostly from data fitting, statistics and machine learning.
Instructor(s): Nathan Srebro Terms Offered: Winter
Prerequisite(s): STAT 30900/CMSC 37810
Equivalent Course(s): STAT 31015, TTIC 31070, BUSN 36903, CAAM 31015

CMSC 35600. Image Processing/Computer Vision. 100 Units.
Equivalent Course(s): MPHY 39600

CMSC 35610. Computational Linguistics I. 100 Units.
This course is an introduction to topics at the intersection of computation and language, oriented toward linguists and cognitive scientists. We will study computational linguistics from both scientific and engineering angles -- the use of computational modeling to address scientific questions in linguistics and cognitive science, as well as the design of computational systems to solve engineering problems in natural language processing (NLP). The course will combine analysis and discussion of these approaches with training in the programming and mathematical foundations necessary to put these methods into practice. Our goal in this quarter is for students to leave the course able to engage with and evaluate research in cognitive/linguistic modeling and NLP, and to be able to implement intermediate-level computational models.
Instructor(s): Allyson Ettinger Terms Offered: Autumn
Prerequisite(s): At least two courses in linguistics or other cognitive science field (or permission of instructor)
Equivalent Course(s): CMSC 25610, LING 38610, LING 28610

CMSC 35620. Computational Linguistics II. 100 Units.
This is the second in a two-course sequence providing an introduction to topics at the intersection of computation and language, oriented toward linguists and cognitive scientists. In this quarter we will cover more advanced topics in cognitive/linguistic modeling and natural language processing (NLP), applying more complex programming and mathematical foundations. Our goal in this quarter is for students to leave the course able to implement advanced models and conduct novel research in cognitive/linguistic modeling and NLP.
Instructor(s): Allyson Ettinger Terms Offered: Spring
Prerequisite(s): Computational Linguistics I or permission of instructor
Equivalent Course(s): CMSC 25620, LING 28620, LING 38620

CMSC 35900. Topics in Artificial Intelligence. 100 Units.
This course covers topics in artificial intelligence.
Terms Offered: Autumn, Winter, Spring
Prerequisite(s): Consent of department counselor and instructor

CMSC 36500. Algorithms in Finite Groups. 100 Units.
We consider the asymptotic complexity of some of the basic problems of computational group theory. The course demonstrates the relevance of a mix of mathematical techniques, ranging from combinatorial ideas, the elements of probability theory, and elementary group theory, to the theories of rapidly mixing Markov chains, applications of simply stated consequences of the Classification of Finite Simple Groups (CFSG), and, occasionally, detailed information about finite simple groups. No programming problems are assigned.
Instructor(s): L. Babai Terms Offered: Spring
Prerequisite(s): Consent of department counselor. Linear algebra, finite fields, and a first course in group theory (Jordan-Holder and Sylow theorems) required; prior knowledge of algorithms not required
Note(s): This course is offered in alternate years.
Equivalent Course(s): MATH 37500
CMSC 37000. Algorithms. 100 Units.
This is a graduate level course on algorithms with the emphasis on central combinatorial optimization problems and advanced methods for algorithm design and analysis. Topics covered include asymptotic analysis, greedy algorithms, dynamic programming, amortized analysis, randomized algorithms and probabilistic methods, combinatorial optimization and approximation algorithms, linear programming, and advanced data structures. Expected outcomes: Ability to design and rigorously analyze algorithms using paradigms such as greedy or dynamic programming. Understand the use of linear programming in optimization. Be able to formulate problems as linear programs. Understand linear programming duality and applications to problems such as max flow/min-cut. Be able to write duals for linear programs.
Instructor(s): Yury Makarychev Terms Offered: Winter
Prerequisite(s): Assumes familiarity with proofs and an the asymptotic notation. Some basic knowledge of the notion of NP-hardness is also required.
Equivalent Course(s): TTIC 31010

CMSC 37100. Topics in Algorithms. 100 Units.
This course covers current topics in algorithms.
Terms Offered: Autumn, Winter, Spring
Prerequisite(s): Consent of department counselor. CMSC 27200 or consent of instructor.

CMSC 37110. Discrete Mathematics. 100 Units.
This course emphasizes mathematical discovery and rigorous proof, illustrated on a variety of accessible and useful topics, including basic number theory, asymptotic growth of sequences, combinatorics and graph theory, discrete probability, and finite Markov chains. This course includes an introduction to linear algebra.
Instructor(s): L. Babai Terms Offered: Autumn
Prerequisite(s): Consent of department counselor and instructor

CMSC 37115. Introduction to Mathematical Reasoning via Discrete Mathematics. 100 Units.
In this course, students with little prior exposure to rigorous mathematical reasoning gain experience in approaching mathematical questions, developing concepts, formalizing ideas, turning intuition into rigorous proof. These phases of mathematical thinking are illustrated on a variety of accessible and useful topics. Students practice the quantifier notation both as a shorthand and as one of the organizing principles of formal statements. New concepts are built from such basic mathematical primitives as numbers, sets, and functions. Basic counting is a recurring theme and provides a source for sequences, another recurring theme, which in turn feeds into the study of asymptotic behavior (rates of growth). Further topics to be covered include proof by induction; the elements of number theory (gcd, congruences, the Chinese Remainder Theorem, Fermat's little Theorem); recurrences, Fibonacci numbers, generating functions; the elements of graph theory (trees, paths and cycle, chromatic number, independent sets and cliques, connectivity, planarity, directed graphs), finite probability spaces, random variables, expected value and variance, independence, concentration inequalities, and random graphs.
Terms Offered: Winter
Prerequisite(s): One quarter of calculus

CMSC 37120. Topics in Discrete Mathematics. 100 Units.
Equivalent Course(s): MATH 37120

CMSC 37200. Combinatorics. 100 Units.
Methods of enumeration, construction, and proof of existence of discrete structures are discussed. The course emphasizes applications of linear algebra, number theory, and the probabilistic method to combinatorics. Applications to the theory of computing are indicated, and open problems are discussed.
Instructor(s): L. Babai Terms Offered: Winter
Prerequisite(s): Consent of department counselor. Linear algebra, basic combinatorics, or consent of instructor.
Note(s): Not offered in 2016-17.

CMSC 37503. Approximation Algorithms. 100 Units.
This is a basic course on approximation algorithms, with the main focus on approximation algorithms for central combinatorial optimization problems. We will mostly focus on classical algorithmic results, but will also present some state of the art results and challenges in the area of approximation. The course will cover major algorithmic techniques, including LP-rounding, primal-dual schema, metric methods, SDP rounding and so on. While the main focus of the course is on algorithms, we will also discuss lower bounds on approximation and connections between algorithm design and lower bound proofs. Assumes the knowledge of material covered in the Algorithms course. Expected outcomes: Understand concepts such as approximation factor, polynomial time approximation schemes and hardness of approximation. Understand applications of linear programs (LPs) to design of approximation algorithms. Learn to analyze rounding algorithms for LPs and understand integrality gaps. Be able to apply LP duality. Understand semi-definite programming and its applications to approximation.
Instructor(s): Julia Chuzhoy Terms Offered: Autumn
Equivalent Course(s): TTIC 31080
CMSC 37530. Graph Theory. 100 Units.
This course covers the basics of the theory of finite graphs. Topics include shortest paths, spanning trees, counting techniques, matchings, Hamiltonian cycles, chromatic number, extremal graph theory, Turan's theorem, planarity, Menger's theorem, the max-flow/min-cut theorem, Ramsey theory, directed graphs, strongly connected components, directly acyclic graphs, and tournaments. Techniques studied include the probabilistic method.
Instructor(s): Laszlo Babai Terms Offered: Spring

CMSC 37701. Topics in Bioinformatics. 100 Units.
This course covers current topics in bioinformatics.
Terms Offered: Autumn,Winter,Spring
Prerequisite(s): Consent of department counselor and instructor

CMSC 37810. Mathematical Computation I: Matrix Computation Course. 100 Units.
This is an introductory course on numerical linear algebra, which is quite different from linear algebra. We will be much less interested in algebraic results that follow from axiomatic definitions of fields and vector spaces but much more interested in analytic results that hold only over the real and complex fields. The main objects of interest are real- or complex-valued matrices, which may come from differential operators, integral transforms, bilinear and quadratic forms, boundary and coboundary maps, Markov chains, correlations, DNA microarray measurements, movie ratings by viewers, friendship relations in social networks, etc. Numerical linear algebra provides the mathematical and algorithmic tools for analyzing these matrices. Topics covered: basic matrix decompositions LU, QR, SVD; Gaussian elimination and LU/LDU decompositions; backward error analysis; Gram-Schmidt orthogonalization and QR/complete orthogonal decompositions; solving linear systems, least squares, and total least squares problem; low-rank matrix approximations and matrix completion. We shall also include a brief overview of stationary and Krylov subspace iterative methods; eigenvalue and singular value problems; and sparse linear algebra.
Terms Offered: Autumn
Prerequisite(s): Linear algebra (STAT 24300 or equivalent) and some previous experience with statistics.
Equivalent Course(s): STAT 30900, CAAM 30900

CMSC 37812. Mathematical Computation III: Numerical Methods for PDE's. 100 Units.
The first part of this course introduces basic properties of PDE's; finite difference discretizations; and stability, consistency, convergence, and Lax's equivalence theorem. We also cover examples of finite difference schemes; simple stability analysis; convergence analysis and order of accuracy; consistency analysis and errors (i.e., dissipative and dispersive errors); and unconditional stability and implicit schemes. The second part of this course includes solution of stiff systems in 1, 2, and 3D; direct vs. iterative methods (i.e., banded and sparse LU factorizations); and Jacobi, Gauss-Seidel, multigrid, conjugate gradient, and GMRES iterations.
Terms Offered: Spring
Prerequisite(s): Some prior exposure to differential equations and linear algebra
Equivalent Course(s): STAT 31100, MATH 38309, CAAM 31100

CMSC 38000-38100. Computability Theory I-II.
The courses in this sequence are offered in alternate years.

CMSC 38000. Computability Theory I. 100 Units.
We investigate the computability and relative computability of functions and sets. Topics include mathematical models for computations, basic results such as the recursion theorem, computably enumerable sets, and priority methods.
Instructor(s): D. Hirschfeldt Terms Offered: Spring
Prerequisite(s): Consent of department counselor. MATH 25500 or consent of instructor.
Equivalent Course(s): MATH 30200

CMSC 38100. Computability Theory II. 100 Units.
CMSC 38100 treats classification of sets by the degree of information they encode, algebraic structure and degrees of recursively enumerable sets, advanced priority methods, and generalized recursion theory.
Instructor(s): D. Hirschfeldt Terms Offered: Spring
Prerequisite(s): Consent of department counselor. MATH 25500 or consent of instructor.
Equivalent Course(s): MATH 30300

CMSC 38100. Computability Theory II. 100 Units.
CMSC 38100 treats classification of sets by the degree of information they encode, algebraic structure and degrees of recursively enumerable sets, advanced priority methods, and generalized recursion theory.
Instructor(s): D. Hirschfeldt Terms Offered: Spring
Prerequisite(s): Consent of department counselor. MATH 25500 or consent of instructor.
Equivalent Course(s): MATH 30300

CMSC 38130. Honors Introduction to Complexity Theory. 100 Units.
Computability topics are discussed (e.g., the s-m-n theorem and the recursion theorem, resource-bounded computation). This course introduces complexity theory. Relationships between space and time, determinism and non-determinism, NP-completeness, and the P versus NP question are investigated.
Terms Offered: Winter
CMSC 38300. Numerical Solutions to Partial Differential Equations. 100 Units.
This course covers the basic mathematical theory behind numerical solution of partial differential equations. We investigate the convergence properties of finite element, finite difference and other discretization methods for solving partial differential equations, introducing Sobolev spaces and polynomial approximation theory. We emphasize error estimators, adaptivity, and optimal-order solvers for linear systems arising from PDEs. Special topics include PDEs of fluid mechanics, max-norm error estimates, and Banach-space operator-interpolation techniques.
Instructor(s): L. R. Scott Terms Offered: Spring. This course is offered in alternate years.
Prerequisite(s): Consent of department counselor and instructor
Equivalent Course(s): MATH 38300

CMSC 38400. Cryptography. 100 Units.
Cryptography is the use of algorithms to protect information from adversaries. Though its origins are ancient, cryptography now underlies everyday technologies including the Internet, wifi, cell phones, payment systems, and more. This course is an introduction to the design and analysis of cryptography, including how "security" is defined, how practical cryptographic algorithms work, and how to exploit flaws in cryptography. The course will cover algorithms for symmetric-key and public-key encryption, authentication, digital signatures, hash functions, and other primitives.
Instructor(s): David Cash Terms Offered: Winter

CMSC 38405. Arithmetic Combinatorics. 100 Units.
This course covers a variety of topics in arithmetic combinatorics such as inverse problems, incidence geometry, uniformity, regularity and pseudo-randomness. A special attention will be paid to connections to classical mathematics and theoretical computer science.

CMSC 38410. Quantum Computing. 100 Units.
This course covers mathematical and complexity aspects of quantum computing, putting aside all questions pertaining to its physical realizability. Possible topics include: (1) quantum model of computation, quantum complexity classes, and relations to their classical counterparts; (2) famous quantum algorithms (including Shor and Grover); (3) black-box quantum models (lower and upper bounds); (4) quantum communication complexity (lower and upper bounds); and (5) quantum information theory.
Instructor(s): A. Razborov Terms Offered: Winter. This course is offered in alternate years.
Prerequisite(s): Consent of department counselor. Basic knowledge of computational complexity and linear algebra required; knowledge of quantum mechanics not required
Note(s): Not offered in 2016-17.
Equivalent Course(s): MATH 38410

CMSC 38500. Computability and Complexity Theory. 100 Units.
Part one of this course consists of models for defining computable functions: primitive recursive functions, (general) recursive functions, and Turing machines; the Church-Turing Thesis; unsolvable problems; diagonalization; and properties of computably enumerable sets. Part two of this course deals with Kolmogorov (resource bounded) complexity: the quantity of information in individual objects. Part three of this course covers functions computable with time and space bounds of the Turing machine: polynomial time computability, the classes P and NP, NP-complete problems, polynomial time hierarchy, and P-space complete problems.
Instructor(s): A. Razborov Terms Offered: Winter
Prerequisite(s): Consent of department counselor. Basic knowledge of computational complexity and linear algebra required; knowledge of quantum mechanics not required
Note(s): Not offered in 2016-17.
Equivalent Course(s): TTIC 31060, MATH 30500

CMSC 38502. Topics in Combinatorics and Logic. 100 Units.
We will discuss several ideas, methods and results in Combinatorics and those parts of Mathematical Logic that are close to Theoretical Computer Science. Complexity Theory itself is excluded this year since I will teach a more systematic course on the subject in the Spring Quarter.
Instructor(s): Alexander Razborov Terms Offered: Winter
Prerequisite(s): None
Equivalent Course(s): MATH 38502

CMSC 38600. Complexity Theory A. 100 Units.
This course covers topics in computational complexity theory, with an emphasis on machine-based complexity classes.
Terms Offered: Spring
Prerequisite(s): Consent of department counselor and instructor

CMSC 38700. Complexity Theory B. 100 Units.
This course covers topics in computational complexity theory, with an emphasis on combinatorial problems in complexity.
Instructor(s): A. Razborov Terms Offered: Spring
Prerequisite(s): Consent of department counselor and instructor
Equivalent Course(s): MATH 38703
CMSC 38800. Complexity Theory. 100 Units.
Complexity Theory is the branch of Theoretical Computer Science that studies inherent limitations on the efficiency of performing various computational tasks. In this course I hope to cover at least the most fundamental results from uniform (Turing) complexity, circuit complexity, communication complexity, algebraic complexity and proof complexity.
Instructor(s): Alexander Razborov Terms Offered: Spring
Prerequisite(s): None, but some familiarity with the book "Computational Complexity" by Arora and Barak might be helpful.
Equivalent Course(s): MATH 38800

CMSC 38815. Geometric Complexity. 100 Units.
This course provides a basic introduction to geometric complexity theory, an approach to the P vs. NP and related problems through algebraic geometry and representation theory. No background in algebraic geometry or representation theory will be assumed.
Instructor(s): K. Mulmuley Terms Offered: Autumn. This course is offered in alternate years.
Prerequisite(s): Consent of department counselor and instructor
Note(s): Background in algebraic geometry or representation theory not required
Equivalent Course(s): MATH 38815

CMSC 39000. Computational Geometry. 100 Units.
This course is a seminar on topics in computational geometry.
Instructor(s): K. Mulmuley Terms Offered: Spring. This course is offered in alternate years.
Note(s): Not offered in 2016-17.

CMSC 39010. Computational and Metric Geometry. 100 Units.
The course covers fundamental concepts, algorithms and techniques in computational and metric geometry. Topics covered include: convex hulls, polygon triangulations, range searching, segment intersection, Voronoi diagrams, Delaunay triangulations, metric and normed spaces, low-distortion metric embeddings and their applications in approximation algorithms, padded decomposition of metric spaces, Johnson-Lindenstrauss transform and dimension reduction, approximate nearest neighbor search and locality-sensitive hashing. Expected outcomes: -- Know standard algorithms and data structures for solving geometric problems -- Be able to design efficient algorithms and data structures for solving geometric problems -- Understand basic concepts of metric geometry such as metric and normed space, low distortion embedding, dimension reduction, nearest neighbor search. -- Understand applications of metric geometry to the field of approximation algorithms and other areas of computer science.
Instructor(s): Makarychev, Yury Terms Offered: Winter
Prerequisite(s): Undergraduate-level algorithms, linear algebra and probability classes; a good background in mathematical analysis/calculus.
Equivalent Course(s): TTIC 31100

CMSC 39020. Geometry, Complexity and Algorithms. 100 Units.
This course will try to explore these three topics and their interactions. Among the topics likely to be discussed are metric measure geometry (e.g. concentration of measure) and its use designing algorithms, machine learning, manifold learning, the complexity of the construction of isotopies and nullcobordisms, the Blum-Cucker-Smale theory of real computation and estimates for the complexity of root finding and related problems, persistence homology and applications, and other topics that seem like a good idea as the course develops.
Equivalent Course(s): MATH 38900

CMSC 39600. Topics in Theoretical Computer Science. 100 Units.
A seminar on current research in theoretical computer science.
Terms Offered: Autumn,Winter,Spring
Prerequisite(s): Consent of department counselor and instructor

CMSC 39800. Rdg/Rsch: Computer Science. 300.00 Units.
Directed reading and research in computer science, under the guidance of a faculty member.

CMSC 70000. Advanced Study: Computer Science. 300.00 Units.
Advanced Study: Computer Science
Font Notice

This document should contain certain fonts with restrictive licenses. For this draft, substitutions were made using less legally restrictive fonts. Specifically:

- Times was used instead of Trajan.
- Times was used instead of Palatino.

The editor may contact Leepfrog for a draft with the correct fonts in place.