Spring 2023 – CS Special Topics

1. **CS 494** – Papka – Introduction to High Performance Computing**
2. **CS 594 – Chattopadhyay – Empirical Methods in HCC**
3. **CS 594 – Medya – Machine Learning for Graphs/Networks**
4. **CS 594 – Sintos – Geometric Data Structures for Fata Queries**
5. **CS 594 – Zhang – Deep Representation Learning (DPL)**

**CS Undergraduate students must submit a modification of major to use the class as a technical elective

Dept. of Computer Science
University of Illinois at Chicago

Back to list

Fall 2022 – CS Special Topics

CS 494 – Introduction to High Performance Computing

- Instructor: Michael Papka
- Meeting time: TR 2-3:15pm
- CRN: 42278/42289

Figure 1: Argonne National Laboratory's newest supercomputer *Aurora* is currently being installed for operation in 2023/24. ***Introduction to High-Performance Computing*** will look at systems like Aurora and answer questions such as what is it made of, how does one program it and how is different from my laptop?

1 Course Description

How are airplanes built without a physical prototype, how do we understand the evolution of the universe, or how are new cancer treatments identified for initial testing? Big problems require big computers - this course is meant to provide a general introduction to the idea of high performance computing and its role in today's world. This course will discuss the what makes up supercomputer, how they are organized and what are the challenges in developing for massive heterogeneous systems.

2 Course Goal

This course aims to introduce students to high-performance computing (HPC) in a general way that is useful to computer science students and all STEM fields. The course will cover fundamental HPC architecture concepts and parallel computing systems software techniques. The course will give students of other domains the needed knowledge to use supercomputers as a vital tool in their quest for new knowledge. The content includes fundamental architecture aspects of shared-memory and distributed-memory systems, as well as paradigms, algorithms, and languages used to program parallel systems. Students will complete several assignments investigating the use of parallel processing systems.

3 Course Work

The course will be lecture based introducing the proposed topics. Additional pointers will be provided to online supplemental material for reinforcement of topics covered. There will be a midterm and final exam, weekly quiz, as well as three or four programming assignments that will implement the concepts presented in class. Access will be provided to needed computational resources to succeed in the course.

4 Tentative Syllabus

4.1 Proposed Schedule

Week	Topic
1	What and Why of High-Performance Computing
2	Introduction to Class Resources
3	High Performance Computing Systems
4	Benchmarking
5	Resource Management
6	Multiprocessor and Accelerator Architectures
7	OpenMP
8	Message Passing Interface
9	Midterm Exam
10	Parallel Algorithms & Libraries
11	Performance Monitoring
12	Debugging
13	Checkpointing
14	Visualization
15	Future of High-Performance Computing
16	Final Exam

4.2 Proposed Evaluation

Area	Percent of Final Grade
Midterm Exam	15%
Final Exam	15%
Assignments	60%
Quiz & Participation	10%

4.3 Book and Readings

Course material will be drawn from the following sources (preliminary list):

- Miscellaneous readings from The *International Conference for High Performance Computing, Networking, Storage, and Analysis*, *International Conference on Supercomputing*, and *International Parallel and Distributed Processing Symposium* as well as others.
- G. Hager and G. Wellein, *Introduction to High Performance Computing for Scientists and Engineers*, CRC Press, 2010.
- T. Sterling, M. Anderson, and M. Brodowicz, *High Performance Computing Modern Systems and Practices*, Morgan Kaufmann Publishers, 2018.
- J. Reinders, B. Ashbaugh, J. Brodman, M. Kinsner, J. Pennycook, and X. Tian, *Data Parallel C++*, Apress, 2021.
- A. A. Chien, *Computer Architecture for Scientists: Principles and Performance*, Cambridge University Press, 2022.

All material needed for course will be available via lecture or online sources. *High Performance Computing Modern Systems and Practices* is recommended but not required for the course.

4.4 Prerequisite

Students are expected to have taken and received an 'C' or better in **CS 251 Data Structures** and comfortable with programming in C/C++.

Back to list

Fall 2022 – CS Special Topics

CS 594 – Empirical Methods in HCC

- Instructor: Debaleena Chattopadhyay
- Meeting time: TR 2-3:15pm
- CRN: 33792

Course Description

If a project involves building human-centered computing (HCC) applications or understanding human-computer interactions, then you must measure project success/viability with a human-subject study or UX research.

More and more, we are building computing tools that directly interface with humans—to improve student learning, benefit group work, create a healthy habit, provide better recommendations, crowdsource our next trip, protect our sensitive data from malicious websites, or gather critical insights from a big data visualization.

How do we know that these computing systems are effective? What evidence do we have that the research project was successful? How do we conduct a meaningful assessment? What are the appropriate metrics? How can we improve the system? To answer all these questions, an in-depth understanding of state-of-the-art UX research methods is essential.

In this course, students will learn how to use human subject data in both generative and evaluative scenarios. This course will cover a range of empirical methods, such as experiment design, hypothesis testing, log analysis, and grounded theory, which will train graduate students to critically examine the implications of human-computer interactions in different CS areas.

For the detailed syllabus from when it was offered a few years back, see the course webpage.

Fall 2022 – CS Special Topics

CS 594 – Machine Learning for Graphs/Networks

- Instructor: Sourav Medya
- Meeting time: TR 12:30-1:45 pm
- CRN: 33648

1 Course Description

Graphs often represent complex data and are an useful abstraction to model social interac- tions, biological systems, and infrastructure. This course focuses on recent modeling, algo- rithmic and computational developments for many real-world problems that involve graphs (or networks). The course will be divided into three parts. One part will focus primarily on graph representation learning techniques that are useful for many graph related tasks such as node classification, link prediction, and graph classification and their applications on specific domains. The majority of it will be spent on various approaches that generalize deep learning on graphs (e.g., graph neural networks) and many different machine learning related tasks over graphs that are being addressed by these models. In another part of the course, we will go over the foundations of optimization algorithms that are commonly used to improve given networks in tasks such as infrastructure improvement and controlling network processes. The third part will consist of discussions of recent papers from major venues. These papers will showcase the recent advancement of machine learning techniques over graphs.

At the end of the course, the goal is to help students to: (1) adapt existing machine- learning algorithms for graph-based problems, (2) build new graph-based machine learning algorithms to solve related problems; and (3) gain knowledge regarding the recent advances and open questions in the field.

2 Coursework

2.1 Student Evaluation

Students will be evaluated based on homework, class participation, paper presentations, and a class project. The tentative grading criteria is as follows:

Sections	Weights
Homework	20%
Participation	10%
Paper presentation	30%
Course Project	40%

2.2 Prerequisites

The prerequisites for this course are listed below:

- CS 401 (Computer Algorithms I) or equivalent
- CS 412 (Machine Learning) or equivalent
- CS 418 (Introduction to Data science) or equivalent

Basic background in Graphs, Linear Algebra, Algorithms, and Machine Learning will be assumed.

Back to list

2.3 Targeted audience

Graduate students from Computer Science, Mathematics, Electrical Engineering, Statistics, and related areas are encouraged to attend this course.

3 Course Outline

We will cover topics at the intersection of Graphs, and Machine Learning, and Discrete Op- timization. The tentative syllabus and schedule are given below:

Weeks	Topics
1	Logistics, Graph Algorithms I
2	Graph Algorithms II, Network Science I: Properties of Real Networks
3	Network Science II: Network Models
4	Optimization on Graphs I: Discrete Optimization Techniques, Randomized Methods
5	Optimization on Graphs II: Applications
6	Network Embedding I: Traditional, Modern (Property-preserving, Structure-preserving)
7	Network Embedding II: Applications in Information Diffusion, Anomaly Detection, Network Alignment
8	Graph Kernels, Project Ideas
9	Graph Neural Networks (GNNs) I: Methods, Dynamic and Heterogeneous GNNs, Neural approaches for Graph Combinatorial Problems
10	Graph Neural Networks II: Interpretability, Robustness
11	Graph Neural Networks III: Applications
12	Graph Neural Networks IV: Theory
13	Paper Presentation I
14	Paper Presentation II
15	Project Presentations
16	Final week (project report due)

3.1 Textbooks

These textbooks might be helpful for the course. However, we will supplement the books with recent papers and notes.

- William L. Hamilton. Graph representation learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2020
- L. Wu, P. Cui, J. Pei, and L. Zhao. Graph Neural Networks: Foundations, Frontiers, and Applications. Springer, Singapore, 2022
- Ma, Yao, and Jiliang Tang. Deep learning on graphs. Cambridge University Press, 2021.
- Mark Newman. Networks. Oxford university press, 2018

Back to list

Fall 2022 – CS Special Topics

CS 594 – Geometric Data Structures for Data Queries

- Instructor: Stavros Sintos
- Meeting time: TR 3:30-4:45pm
- CRN: 33649

Course Description In the digital age, tera-bytes of data are generated every second. A large amount of raw data would be useless if we could not analyze them to extract useful knowledge. In order to analyze a large data set or a database, a user might ask data queries to a data structure built on the entire or a subset of the input data set. With the data sets becoming increasingly large and complex, queries are also becoming complex, therefore new challenging problems have emerged in the area of query processing. Furthermore, many applications in data science are inherently accompanied with geometry, such as visualization or geospatial analytics. Some of the best algorithms for well-known problems in data management and databases, such as clustering, top-*k* queries, or data matching, use ideas from geometric optimization to achieve high performance. This course exploits how to use geometry to build efficient and practical data structures for various (complex) data queries over large data sets. We will discuss applications in many fields, including interactive analytics, search engines, recommendation systems, automatic fact-checking, computational journalism, and sensor networks. The goal of this course is to provide to the students all the basic tools to conduct research in the area of query processing and computational geometry. In the first half of the course, the instructor will provide the main ideas behind most of the fundamental geometric data structures such that range tree, kd-tree, quad-tree, cover tree etc. In the second half of the course, students will present recent papers from top-tier conferences and journals, and explore multiple applications of geometric data structures in databases and more generally in data science.

Course Work There will be two 75 minutes lectures per week. Students are expected to participate in all the lectures.

Every student will present and lead the discussion of one or two papers during the second half of the semester. The exact number of presentations per student will depend on the number of students enrolled in the course. Students can choose the papers from the list provided by the instructor. Students cal also pick other papers to present after they receive approval from the instructor. The goal is to cover a different topic or application in databases every week.

The students will work on a project of their interest that incorporates ideas discussed in the class. A project can be one of the following:

- Original research. The students can use the ideas presented in the class to propose new or better solutions to a related problem in databases, computational geometry or any other area.
- Extend current research: The students can choose to work on a special case of a research problem they have already started, using ideas and concept presented in the current class.
- Implementation: The students can implement a geometric algorithm or a data structure and run experiments for a practical data science problem on real and synthetic datasets.
- Survey: Students can write a survey about an area related to a topic covered in this class.

Students are allowed to work in teams of 1-3 people. Larger teams are also allowed if the students decide to work on conducting original research or extending current research. In the last week of the semester each team should submit a report and prepare a short presentation. The instructor will hold frequent meetings with each team to guide their progress. There will be no exams in this class.

Prerequisite The course is be accessible to students with a wide range of backgrounds, includ- ing both theoretical and applied areas of computer science and mathematics. The course CS 401– Algorithms I is required since some familiarity over basic algorithmic concepts will be assumed. How- ever, the instructor will cover most of the requirements in the first half of the course.

Back to list

Fall 2022 – CS Special Topics

Course Outline (tentative) In the first half of the course the instructor will use various chapters from the following books:

- M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf. *Computational Geometry: Algorithms and Applications,* Springer, 3rd edition, 2008.
- S. Har-Peled. *Geometric approximation algorithms,* American Mathematical Soc., 2011
- J. Matousek. *Lectures on discrete geometry,* Volume 212. Springer Science & Business Media, 2013.

In the second half of the course, the students will present papers from conferences and journals such as PODS, SIGMOD, VLDB, ICDE, ICALP, TODS.

Week	Topic
	Weeks 1-7: Foundations
Week 1	Introduction to Geometric Data Structures Trees, heaps, interval trees, segment trees
Week 2	Aggregation Range Queries Range tree, kd-tree, R-tree
Week 3	Nearest Neighbor Queries Quad-tree
Week 4	More on Nearest Neighbor Queries WSPD, LSH
Week 5	Simplex Queries Geometric cuttings, Partition tree
Week 6	Ball Queries and Queries in General Metric Spaces Clustering, Cover tree, Net tree
Week 7	Summarization for Data Queries Sampling, Coresets
	Weeks 8-14: Special Topics (Presentations)
Week 8	Durable Top-k Queries
Week 9	Diverse Range Queries
Week 10	Fairness and Data Queries
Week 11	Queries under Uncertainty
Week 12	Join Queries
Week 13	Approximate Query Processing
Week 14	Machine Learning for Approximate Query Processing
Week 15	Project Presentations

Fall 2022 – CS Special Topics

CS 594 – Deep Representation Learning (DPL)

- Instructor: Xinhua Zhang
- Meeting time: W 6-8:40pm
- CRN: 34724

Course Description: The recent success of deep learning stems in a large part from the capability of neural networks in synthesizing salient representations of data, superseding traditional shallow methods that assumes the representation is given. Representation learning has been shown effective for many types of data (images and natural language) with a variety of prior structures (invariance, equivariance, disentanglement), and under a range of learning scenarios (multi-task, multi-modal, domain adaptation, reinforcement learning). It also facilitates the incorporation of diverse desiderata of learning such as fairness, safety, and explainability. We will cover most of the cutting-edge techniques for representation learning and unsupervised/self-supervised learning, especially in the context of deep neural networks, and apply them to important applications.

Goal: The primary goal of this course is to prepare students to conduct research in representation learning -- either in the development of improved representation learning methods or the application of representation learning techniques to new domains or priors. Students will gain familiarity with the key technical challenges surrounding representation learning and state-of-the-art representation learning methods.

Student Deliverables:

- **Research Paper Readings and Presentation**. Students will be required to read all papers. Each student will present at least once, and students will be required to participate in paper discussion. To ensure that students read all assigned papers, they will be required to turn in a summary for each paper before class. Students will be graded based on their presentation as well as their participation during paper discussions.
- **Research**. Students will submit research proposals at the 3rd/ 4th week, a project status update at the end of the 8th week, and a final write-up towards the end of the course (structured as a research paper). Each team will also provide a 20-minute presentation of their project in class.

Grading policy:

Paper summaries and discussion 20%
Paper presentations 20%
Project 60%

Prerequisites: Students are required to have taken and received an 'A' or 'B' in **all** the following courses: CS 412, MATH 310/320, STAT 401, **and** CS 251.

Back to list

Fall 2022 – CS Special Topics

Tentative Weekly Schedule of Topics:

1. Overview and classic representation learning (LLE, Isomap, LDA, CCA, MDS, SVD, Eigenmaps)
2. Modern (deep) DRL and self-supervised learning techniques (contrastive, deep embeddings, etc)
3. Variational auto-encoding
4. Disentangled DRL
5. Invariant and equivariant DRL
6. Secure and robust DRL
7. Fair DRL
8. Explainable DRL
9. Metric/similarity based learning
10. DRL for sequential and temporal data (transformer)
11. DRL for domain adaptation and continual learning
12. DRL for multi-modal and multi-task learning
13. DRL for reinforcement learning
14. DRL for AutoML
15. Course project presentation