Fall 2019 – CS Special Topics Courses

1. **CS 494* – Hummel (BioE/CS) – Wearable Technologies**
   - **This class/section is for CS undergrads only**
2. **CS 494 * – Kshemkalyani – Parallel and Distributed Processing**
3. **CS 494* – Mobasheri – Advanced Data Structures and Algorithms**
   - **This class/section is for CS undergrads only**
4. **CS 594 – Caragea – Deep Learning for NLP**
5. **CS 594 – Sidiropoulos – Graph Algorithms**
6. **CS 594 – Stephens – Data Center Networking**
7. **CS 594 – Zheleva – Causal Inference and Learning**
8. **CS 594 – Sidiropoulos – Graph Algorithms**

*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

Updated: 8/20/19

Fall 2019 – CS Special Topics Courses

CS 494– Wearable Technologies

- Instructor: Joe Hummel (CS) and Hananeh Esmailbeigi (BioE)
- Meeting time: T 2-4:50pm
- CRN: 43965

Course Description:
This course, taught primarily by BioE, is a lab-based course in the design and construction of wearable devices. Students work in teams of 4, 2 from CS and 2 from BioE, and use HW and SW skills to design, construct, and analyze wearable devices and the collected data. The class is run like a true lab, where class meets once/week for 3 hours, and class time is devoted mostly to project work. If necessary, projects are completed outside of class time.

Coursework:
3 assigned projects + a final project of team's own design. Each project involves HW and SW integration, some circuit design, data analysis, and a team report. Team work required.

Prerequisites:
CS 251 and 261, and CS 362 if possible.

Fall 2019 – CS Special Topics Courses

CS 494– Parallel and Distributed Processing

- Instructor: Ajay Kshemkalyani
- Meeting time: TR 3:30-4:45pm
- CRN: 44014- undergraduate
  44015- graduate

Course Description:
This class explores the foundations of parallel and distributed computing. Emphasis will be laid on parallel and distributed algorithms, for basic topics ranging from graph algorithms to mutual exclusion, deadlock detection, and predicates detection algorithms, to design of communication algorithms for elementary primitives such as multicast, broadcast, scatter, gather, all-to-all personalized communication on distributed systems as well as clusters based on specific topologies. Basic concepts such as scalability of systems (i.e., combination of architecture plus algorithm) will also be explored. As a case study, the Message-Passing Interface (MPI) will be introduced, and students will be exposed to programming using MPI on UIC's Extreme cluster. Ultimately, the goal is to foster an understanding of the various ways in which concurrent actions in the system and the lack of global knowledge in the system can affect the design of parallel and distributed algorithms.

Student Deliverables:
There will be one or two midterms and a final exam. Students will be expected to complete about 3 or 4 programming assignments using MPI. There will also be a theory-based project/term paper.

Prerequisites:
Programming skills amenable to programming the UIC Extreme cluster, in a language such as C, is required. CS 401 and CS 450 are recommended but not required.

Course website: https://www.cs.uic.edu/~ajayk/c494fa19.html

CS 494– Advanced Data Structures and Algorithms

- Instructor: Nasim Mobasheri
- Meeting time: MWF 1-1:50pm
- CRN: 43881

Course Description:
This course focuses on design and implementation details of non-trivial algorithms. As an extension to 401, the course covers complex data structures with an emphasis on amortized analysis and implementation, as well as related non-trivial algorithms that use the same data structures in their implementation.
The course will also introduce students to methods of coping with intractability by analyzing easy special cases and introduction of SAT solvers. The student will be tasked to identify and investigate other methods of coping with intractability (local search, approximation algorithms, branch and bound, …) in a form of a project/presentation. The course requires extensive programming.

Main Topics:

1. Beyond data structures
    - Trees (AVL, BlackRed, Tries)
    - Heaps (binary, binomial , Fibonacci)
    - Disjoint data sets
2. Overview of algorithms
    - String algorithms / Bitwise manipulation
    - Graph algorithms
        - Traversal, shortest path, MST
    - Dynamic Programming
    - Network flow
        - Bipartite matching, min/max cut
    - Computational geometry
        - Cross product, convex Hull, plane intersections
3. Coping with intractability
    - Case studies of coping strategies with np-hard problems
        - NP-hard problems on easy special cases
        - SAT solvers
4. Student Project/Presentations

Evaluation and Student Deliverables:
The student will have to participate in in-class exercises each session. These exercises can be of a white-board interview format with students forming groups of two or three and presenting proper solution or individual quizzes.

Fall 2019 – CS Special Topics Courses

There will be at least 3 projects and multiple homework assignments. Majority of projects and assignments are programming.
The course will have one midterm and one final exam.

Prerequisites:
CS 401

Fall 2019 – CS Special Topics Courses

CS 594– Deep Learning for Natural Language Processing

- Instructor: Cornelia Caragea
- Meeting time: M 3-5:50pm
- CRN: 43915

Course Objectives: Natural language processing (NLP) is one of the most important technologies to-day due to the large and growing amount of online text that needs to be understood in order to get the enormous value out of it. Although many machine learning models have been developed for NLP appli-cations, recently, deep learning approaches have achieved remarkable results across many NLP tasks. The course provides an introduction to research in deep learning applied to NLP. We will cover topics such as word vector representations, convolutional neural networks, recurrent neural networks, and long-short-term-memory networks. We will also cover tools and software available for building and training deep neural networks. Through lectures and programming and reading assignments students will learn the necessary skills for applying and designing neural networks for practical NLP problems.

Course Work and Evaluation: Students will be evaluated based on reading and programming assign-ments, paper presentations, and a class project. Students are encouraged to attend every lecture and to participate in class discussion. The grading criterion is shown below:

Prerequisites: Linear algebra and calculus, machine learning, natural language processing. CS 412: Introduction to Machine Learning; CS 421: Natural Language Processing.

CS 594– Graph Algorithms

- Instructor: Anastasios Sidiropoulos
- Meeting time: TR 2-3:15
- CRN: 27441

Narrative Description: Large and complex interconnected systems have become ubiquitous in the modern world, from science and engineering, to finance and commerce. In many scenarios, the structure of such systems is modeled by networks of interacting entities. This modeling paradigm can be used when studying a plethora of natural objects and phenomena, such as the web, networks of all kinds – social, transportation, communication, phylogenetic – financial transactions, and soon. The analysis of large and complex networks is therefore a task of increasing importance to society.
However, reaping the potential benefits from the analysis of these objects poses great new challenges for computational sciences. Contemporary algorithmic graph theory seeks to address these challenges by drawing tools and ideas from diverse areas of mathematics, including geometry, statistics, and mathematical programming.

Goal: In this course, the students will be exposed to algorithmic methods used in the analysis of graphs. Emphasis will be given on understanding the state of the art of these methods, and on developing intuition about which methods are appropriate in various application contexts.

Student deliverables: The students will have to read all the papers, and they will be expected to actively participate in all the lectures. Furthermore, each student will present at least one research paper to the class. For the final project, the students will have to submit a proposal of their selected topic within the first half of the course, a final report at the end of the class, and they will be asked to give a brief presentation on their findings

Course Work and Evaluation: Students will be evaluated based on reading and programming assign-ments, paper presentations, and a class project. Students are encouraged to attend every lecture and to participate in class discussion. The grading criterion is shown below:

Prerequisites: The course will be accessible to students with a wide range of backgrounds, including both theoretical and applied areas of computer science. Some familiarity with algorithms will be assumed, equivalent to a CS 401-level course. 1 Exams: There will be no exams.

Fall 2019 – CS Special Topics Courses

CS 594– Data Center Networking

- Instructor: Brent Stephens
- Meeting time: TR 12:30-1:45pm
- CRN: 40393

Course Description:
This class explores technologies, techniques, and designs for cloud data center networking, using real production networks at cloud providers like Google, Microsoft, and Amazon as an example. Key topics covered in this course include protocol independent programmable networking (RMT/P4), RDMA, and Network Function Virtualization (NFV). Additional topics include multipath topologies and routing, load balancing, network virtualization, fault-tolerance, performance isolation, network acceleration, in-network computing, and explicit congestion control. Ultimately, the goal is to foster an understanding of the many different aspects of data center networking in a way that is both comprehensive and current. Students will "build their own cloud network" for experimentation (via CloudLab) throughout the duration of the class. Additionally, students will present previous research efforts on data center networking. The number of presentations will depend on the class size, though will not be more than 2. The course will also include reading and/or programming assignments.

Student deliverables and Class Meetings:
Students will be expected to read approximately one paper per session, present at least one previously completed project over the semester, complete 3 homework assignments, and conduct one group analysis project which will include both a written and presentation component. The final project will be graded based on its correctness, thoroughness, clarity, and soundness of the analysis. Initial class meetings will include lectures on data center networking fundamentals; later meetings will focus on discussing the different aspects of data center networking used in exceptional and recent publications.

Prerequisites:
Programming skill amenable to programming virtual and physical networks (in a language like C) is required. Completion of the student skills and interest survey. Students who are not thesis option MS students or PhD students are encouraged to contact the instructor prior to enrollment. CS 450 is also recommended.

CS 594– Causal Inference and Learning

- Instructor: Elena Zheleva
- Meeting time: TR 11a-12:15pm
- CRNs: 43136

Description:
Reasoning about causal relationships is an integral part of data science and artificial intelligence. The goal of this course is to introduce students to methodologies and algorithms for causal reasoning and connect various aspects of causal inference, including methods developed within machine learning, statistics, and economics. The course will cover state-of-the-art research on causal reasoning and prepare students to conduct research in this area.

Prerequisites:
CS 412 or consent of the instructor.

Grading policy:
Paper summaries and discussion 20%
Paper presentations 20%
Project 60%

Main topics:
1. Introduction to causal inference: identifiability, ignorability, SUTVA, selection bias, confounding, causal effect estimation, randomized controlled trials.
2. Potential outcomes framework: matching and propensity score models, natural experiments and regression discontinuity, instrumental variables.
3. Causal graphs: encoding causal assumptions with graphical models, do-calculus and controlling for confounding, counterfactual and interventional logic, transportability, causal structure learning.
4. Current topics in causal learning: causal invariance search, role of causality in machine learning, causal representations, causal explanation, causal discovery, individual and heterogeneous treatment effects, algorithmic confounding
5. Causal inference for network data: interference bias, inferring network effects from observational data, contagion and influence, graph mining approaches to causal inference, statistical relational learning of causal models

Course website: https://www.cs.uic.edu/~elena/courses/fall19/cs594cil.html

Fall 2019 – CS Special Topics Courses

CS 594– Graph Algorithms

- Instructor: Anastasios Sidiropoulos
- Meeting time: TR 2-3:15 pm
- CRNs: 27441

Description:
Large and complex interconnected systems have become ubiquitous in the modern world, from science and engineering, to finance and commerce. In many scenarios, the structure of such systems is modeled by networks of interacting entities. This modeling paradigm can be used when studying a plethora of natural objects and phenomena, such as the web, networks of all kinds - social, transportation, communication, phylogenetic - financial transactions, and so on. The analysis of large and complex networks is therefore a task of increasing importance to society.

However, reaping the potential benefits from the analysis of these objects poses great new challenges for computational sciences. Contemporary algorithmic graph theory seeks to address these challenges by drawing tools and ideas from diverse areas of mathematics, including geometry, statistics, and mathematical programming.

Prerequisites:
The course will be accessible to students with a wide range of backgrounds, including both theoretical and applied areas of computer science. Some familiarity with algorithms will be assumed, equivalent to a CS 401-level course.

Grading policy:
The students will have to read all the papers, and they will be expected to actively participate in all the lectures. Furthermore, each student will present at least one research paper to the class. For the final project, the students will have to submit a proposal of their selected topic within the first half of the course, a final report at the end of the class, and they will be asked to give a brief presentation on their findings.

Method of instruction: The instruction will be based on the following main components:

- During the first half of the course, the instructor will present various fundamental methods and ideas used in the design of algorithms on graphs. Any necessary prerequisites will also be discussed during this time.
- During the second half of the course, the students will read and present research papers.
- The students will work on a project of their interest that incorporates ideas discussed in the class. The students will have the option to either conduct original research or experimentally evaluate prior work. The project will be performed in teams of 1{3

students. The students will be encouraged to start thinking about possible research topics early in the semester. The instructor will hold frequent meetings with each team to guide their progress.

Exams: There will be no exams.

Readings: Selected books and research papers from the following tentative list of topics:

Community detection: How to find well-connected subgraphs, with applications to the analysis of social networks.

Graph clustering: Algorithms for partitioning a graph, optimizing various connectivity objectives, such as spectral partitioning, min-cuts, sparsest-cuts, multiway-cuts, and so on.

Distances in graphs: Algorithmic methods for geometric problems in graphs, such as the Traveling Salesperson Problem, Minimum Spanning Trees, shortest paths, and so on.

Flows in graphs: Min-cut/max-flow duality, and its extensions to multi-commodity flows. Connections to the geometry of graphs, and applications to divide & conquer.

Graph compression: Methods for representing succinctly large graphs, such as spectral sparsifiers, vertex sparsifiers, graph spanners, and so on.

Algorithmic graph-minor theory: Dynamic programming on graphs via tree decompositions. Algorithms for graphs on surfaces.

Sample of relevant papers:

- Batson, Spielman, and Srivastava, Twice-Ramanujan Sparsifiers.
- T. Leighton and S. Rao, Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms.
- H. Racke, Optimal hierarchical decompositions for congestion minimization in networks.
- D. Spielman, and S.-H. Teng, Spectral partitioning works: Planar graphs and finite element meshes.
- J. R. Lee, S. Gharan, and L. Trevisan, Multiway Spectral Partitioning and Higher-Order Cheeger Inequalities.
- H. L. Bodlaender, A tourist guide through treewidth
- A. Madry, Navigating Central Path with Electrical Flows: From Flows to Matchings, and Back.

The discussion of the above topics will include motivational examples from various application domains, such as the analysis of social networks, bioinformatics, networking, and machine learning.