Fall 2024 – CS Special Topics

1. CS 594 – Glavic – Provenance and Explanations
2. CS 594 – Sidiropoulos – Foundations of Permissionless Systems
3. CS 594 – Sintos – Geometric Data Structures for Data Queries

Dept. of Computer Science
University of Illinois at Chicago

Updated: 3/22/2024

CS 594 – Glavic – Provenance and Explanations

- Instructor: Boris Glavic
- Meeting Time: TR 12:30-1:45
- CRN: 40393

1 Course Description

Provenance and explanations are essential tools for building trust-worthy, secure, transparent, and fair data-intensive systems and machine learning pipelines. These tools are used to debug analysis results, to comprehend the results of complex queries, to explore the impact of hypothetical changes to data and/or policies, to audit sensitive computations, and to justify and understand predictions made by machine learning models. This course provides a comprehensive overview of algorithms, systems, and techniques for capturing & managing data provenance, i.e., tracking the origin and creation process of data, as well as for generating explanations for data intensive computations such as declarative queries and machine learning.

The goal of this course is to provide students with the necessary tools to build provenance-enabled systems and develop automated solutions for generating explanations.

2 Course Topics

The following topics will be covered in the course:

- **Provenance & Explanations - Introduction**
  - Motivation & use cases
  - Provenance graphs
  - Explanations for query answers
- **Provenance models**
- **Hypothetical reasoning: what-if and how-to**
  - Incremental view maintenance / what-if queries
  - View update & how-to

- **Explanations**
  - Counterfactual explanations
  - Explanations as (provenance) summarization
  - Attribution and degrees of responsibility (including game theoretic notions of attribution)
- **Explaining missing answers**
- **Provenance capture & management**
  - How to compute provenance efficiently?
  - Storage and computation trade-offs
- **Building provenance-aware & explanation-ready systems**
  - Strategies for capturing and managing provenance
  - How to compute explanations efficiently?

3 Course Organization

3.1 Materials

The following overview articles and textbooks will be helpful, but are optional.

- **Data Provenance - Origins, Applications, Algorithms, and Models.**, *Boris Glavic*. Foundations and Trends® in Databases, vol. 9 (3-4), 209-441, 2021.
- **Trends in Explanations: Understanding and Debugging Data-Driven Systems.**, *Boris Glavic*, *Alexandra Meliou*, *Sudeepa Roy*. Foundations and Trends® in Databases, vol. 11 (3), 226-318, 2021.
- **Principles of Data Integration**, 1th Edition, *Doan*, *Halevy*, *and Ives*, Morgan Kaufmann, 2012

3.2 Grading

- Project: **40%**

- Paper review and presentation: **40%**
- Homework assignment & Quizzes: **10%**
- Active participation in class: **10%**

3.3 Class organization

- The first half of the class will consist mostly of lectures given by the instructor to introduce students to necessary background in databases, provenance, and explanations.
- In the second half, students will read and present research papers related to the topics covered in the course.
- After the first few classes, students will have to decide on a research project. In this project, students will either implement and/or evaluate an existing techniques from a state-of-the-art research paper or work on novel research. The results of these project will be towards the end of the semester. The instructor will provide guidance to students that are interested in publishing their work developed in this course where appropriate.

3.4 Workload

In this course, students will . . .

1. Work on a semester-long research project related to implementing provenance or explanation techniques based on a research paper or working on developing new techniques.
2. Review and present a state-of-the-art research paper from the field.
3. Actively participate in class
4. Homework assignments / quizzes

3.5 Prerequisites

No formal prerequisites, but some background in databases (roughly equivalent to CS480) is expected.

Back to list

CS 594 – Foundations of Permissionless Systems

- Instructor: Tasos Sidiropoulos
- Meeting Time: TR 2-3:15
- CRN: 27441

Course description: A blockchain is a tamperproof sequence of data that can be read and augmented by everyone. The first implementation of such a structure, proposed by Satoshi Nakamoto, is the Bitcoin protocol. Blockchains have found numerous applications, such as cryptocurrencies and smart contracts, and hold the potential to revolutionize the way a democratic society operates.

From the scientific point of view, blockchains present several new exciting opportunities, as well as technical challenges. The intellectual underpinnings of the design of such public ledgers lie in the intersection of cryptography, distributed computing, algorithms, game theory, and economics. This unique combination of diverse scientific disciplines necessitates the development of new theoretical foundations for this emerging area.

Goal: In this course, the students will be exposed to the theoretical foundations underpinning the design and operation of blockchains. Emphasis will be given on understanding how the properties of blockchains lead to several other important primitives, such as cryptocurrencies, smart contracts, digital assets, and so on. Furthermore, the students will learn about important challenges that blockchains face, such as scaling, transaction routing, energy consumption, and so on.

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.

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 discrete math (equivalent to CS 201), algorithms (equivalent to CS 401) and computability theory (equivalent to CS 301) will be assumed. All necessary cryptographic primitives (elements of public key cryptography, zero knowledge proofs, elliptic curve cryptography, and so on) will be introduced during the course.

Exams: There will be no exams

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

Byzantine Agreement: What is the Byzantine generals problem? How does Nakamoto consensus solve the token distribution problem?

Game-theoretic aspects of blockchains: Is the Bitcoin protocol incentive-compatible? How can we mathematically analyze chain forks?

The Bitcoin protocol and its extensions: What is the Bitcoin Backbone protocol? Can we verify parts of a blockchain without reading the whole list of blocks? What are Non-interactive Proofs of Proof of Work?

Network-theoretic aspects of blockchains: What is a peer-to-peer network? What is an eclipse attack? How does this affect the security of cryptocurrencies?

Energy consumption: The Nakamoto consensus is based on a "proof of work" algorithm, which uses an enormous amount of energy. Several other "proof of stake" protocols have been proposed that try to mitigate this issue (Algorand, Fruitchains, Ouroboros, etc). How do these protocols work?

Scalability: The Bitcoin protocol can support only a limited number of transactions per second. Recent theoretical work suggests that this is an inherent limitation of blockchains. In order to bypass this obstacle, various other, so-called "second-layer" algorithms and protocols have been proposed. Some of these proposed solutions include the lightning network, plasma, rollups, side-chains, and so on. How do these work?

Turing-completeness: The Bitcoin protocol supports only a limited number of types of transactions, specified as the Bitcoin script language. This language is powerful enough to program various interesting primitives, such as inter-blockchain transactions, the lightning network, and so on. However, this language is not Turing-complete. Many other blockchains have been created (e.g., Ethereum, Cardano, and so on) that provide a Turing-complete set of transactions, and can be used to implement arbitrary mechanisms, such as voting, governance, public registries, and so on.

Economic aspects of blockchains: The construction of blockchains with Turing-complete languages has lead to the implementation of a plethora of financial primitives that operate without a central authority. These include: decentralized autonomous organizations (DAOs), algorithmic stable coins, automated marked makers, trustless loans, and so on.

Sample of relevant papers:

- Xi Chen, Christos Papadimitriou, Tim Roughgarden, *An Axiomatic Approach to Block Rewards.*
- J Garay, A Kiayias, N Leonardos, *The bitcoin backbone protocol: Analysis and applications.*
- A Kiayias, A Russell, B David, R Oliynykov, *Ouroboros: A provably secure proof-of-stake blockchain protocol.*
- A Kiayias, A Miller, D Zindros, *Non-interactive proofs of proof-of-work.*
- J Chen, S Micali, *Algorand.*
- Y. Gilad, R. Hemo, S. Micali, G. Vlachos, N. Zeldovich, *Algorand: Scaling Byzantine Agree-*

ments for Cryptocurrencies.

- I Eyal and E G Sirer. *Majority is not Enough: Bitcoin Mining is Vulnerable.*
- S Tochner, A Zohar. *How to Pick Your Friends - A Game Theoretic Approach to P2P Overlay Construction.*
- Lewis Gudgeon, Sam Maximilian Werner, Daniel Perez, William J. Knottenbelt. *DeFi Protocols for Loanable Funds: Interest Rates, Liquidity and Market Efficiency.*

Back to list

CS 594 – Geometric Data Structures for Data Queries

- Instructor: Stavros Sintos
- Meeting Time: MW 5-6:15
- CRN: 43915

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 can 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.

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, including 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. However, the instructor will cover most of the requirements in the first half of the course.

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