Syllabus

Instructor: Jeff M. Phillips. | 3404 MEB | http://www.cs.utah.edu/~jeffp

Class Meetings: Mondays and Wednesdays, 1:25pm – 2:45pm, HEB 2004.

Course Web Page: http://www.cs.utah.edu/~jeffp/teaching/DataMining.html

Data mining is the study of efficiently finding structures and patterns in large data sets. Direct goals of these tasks are to find anomalies and outliers, but these key algorithms also form the building blocks for much of the broader field of data analysis. We will focus on several aspects of this: (1) converting from a messy and noisy raw data set to a structured and abstract one, (2) applying scalable and probabilistic algorithms to these well-structured abstract data sets, and (3) formally modeling and understanding the error and other consequences of parts (1) and (2), including choice of data representation and trade-offs between accuracy and scalability. These steps are essential for training as a data scientist.

Algorithms, programming, probability, and linear algebra are required tools for understanding these approaches.

Topics will include: similarity search, clustering, dimensionality reduction, graph analysis, PageRank, and small space summaries. We will also cover several recent developments, and the application of these topics to modern applications, often relating to large internet-scaled applications.

Learning Objectives

On completion of this course students will be able to:

- convert a structured data set (like text) into an abstract data representation such as a vector, a set, or a matrix, with modeling considerations, for use in downstream data analysis
- implement and analyze touchstone data mining algorithms for clustering, dimensionality reduction, graph analysis, and locality sensitive hashing.
- understand, discuss, and evaluate advanced data mining algorithms for clustering, dimensionality reduction, graph analysis, locality sensitive hashing, and managing noisy data.
- work with team to design and execute a multi-faceted data mining project on data which is not already structured for the analysis task, and to compare and evaluate the design choices.
- present progress and final results using written, oral, and visual media on a data analysis project to peers in small groups, to peers in large interactive environment, and to get approval from a superior.

Getting Help

Take advantage of the instructor and TA office hours (posted on course web page). We will work hard to be accessible to students. Please send us email if you need to meet outside of office hours. Don't be shy if you don't understand something: come to office hours, send email, or speak up in class!

Students are encouraged to use a discussion group for additional questions outside of class and office hours. The class will rely on the Canvas discussion group. Feel free to post questions regarding any questions related to class: homeworks, schedule, material covered in class. Also feel free to answer questions, the instructors and TAs will also actively be answering questions. But, **do not post potential homework answers**. Such posts will be immediately removed, and not answered.

All important announcements will be made through the discussion group, there is otherwise no class mailing list.

Prerequisits

A student who is comfortable with basic probability, basic linear algebra, basic big-O analysis, and basic programming and data structures should be qualified for the class. A great primer on the Foundations of Data Analysis can be found in here
https://users.cs.utah.edu/~jeffp/teaching/FoDA.html.

The preferred programming language for the course is python, and assignments will be structured towards its use. The instructor and TAs may not be able to offer help with other programming languages. Programming assignments will often (intentionally) not be as specific as in lower-level classes. This will somewhat simulate real-world settings where one is given a data set and asked to analyze it; in such settings even less direction is provided.

For undergrads, the formal prerequisites are CS 3500 and CS 3190 (which has CS 3130 and Math 2270, or equivalent as pre/co-regs).

For graduate students, there are no enforced pre-requisites. Still it may be useful to review early material in Mathematical Foundation for Data Analysis (mathfordata.github.io; e.g., Chapters 1,3 and first parts of 2,5,7).

In the past, this class has had undergraduates, masters, and PhD students, including many from outside of School of Computing. Most (but not all) have kept up fine, and still most have been challenged. If you are unsure if the class is right for you, contact the instructor.

For an example of what sort of mathematical material I **expect you to be intimately familiar with**, see chapters 1 and 3 in Mathematical Foundation for Data Analysis (mathfordata.github.io). Other relevant material from CS 3190 will be reviewed, but very rapidly.

Grading

The grading will be 30% from homeworks and 40% from a project and 30% from tests.

We will plan to have 7 short homework assignments, roughly covering each main topic in the class. The homeworks will usually consist of an analytical problems set, and sometimes a programming exercise. There will be no specific programming language for the class, but some assignments may be designed around a specific one that is convenient for that task.

Each person in the class will be responsible for a group project. The project will be very open-ended; basically it will consist of finding an interesting data set, exploring it with one or more techniques from class, and presenting what you found. I will try to provide suggestions for data sources and topics, but ultimately the groups will need to decide on their own topic. There will be several intermediate deadlines so projects are not rushed at the end of the semester. Details of the project requirements can be found here:
http://www.cs.utah.edu/~jeffp/teaching/DM/project.pdf

There will be two tests, each covering roughly half the material in class. They will be open notes; you can bring 1 sheet of paper (front and back). No computing devices (including calculators) will be allowed.

Letter Grade Mapping: I will plan to map numerical grades to letter grades at the standard scale:

- 90-100 : A- to A
- 80-90 : B- to B+
- 70-80 : C- to C+
- 60-70 : D- to D+
- below 60 : E

The G- to G to G+ breakdown (for grade G = {A,B,C,D}) will probably align along:

- N0 to N3 : G-
- N3 - N7 : G
- N7 - N9.99 : G+

but I will reserve the right to shift this slightly. I also might also make the letter grade breakdown slightly more favorable (this has occurred for CS 5140 in the past, but not every year).

Late Policy

To get full credit for an assignment, it must be turned in through GradeScope before the start of class, specifically 1:00pm. Once the 1:00pm deadline is missed, those turned in late will lose 10%. Every subsequent 24 hours until it is turned another 10% is deducted. Assignments will not be accepted more than 48 hours late, and will be given a 0.

Assignments will be posted far enough ahead of time that I will not be able to make exceptions if a student falls ill. The exception will be prolonged illness accompanied by a doctor's note.

If you believe there is an error in grading (homeworks or quizzes), you may request a regrading within **one week** of receiving your grade. Requests must be made by email to instructor, explaining clearly why you think your solution is correct.

Collaboration Policy

For assignments, you may discuss answers with anyone, including problem approach, proofs, and code. But all students must write their own code, proofs, and write-ups.

For projects, you may of course work however you like within your groups. You may discuss your project with anyone as well, but if this contributes to your final product, they must be acknowledged (this does not count towards page limits). Of course any outside materials used must be referenced appropriately.

For tests, you must work by yourself. Students talking with other students during the tests will get a 0 score.

Using AI Tools

For the Midterm and Endterm, no calculating devices will be allowed, so with no calculation, you get no AI.

For Assignments and the Project some AI assistance is allowed. You are always permitted to use spell checking, grammar assist, translation help, and code auto-complete. No declaration is needed. You are never permitted to copy and paste the entire assignment question into an AI-assisted chat bot to get help to produce the solution (although translation, e.g., to Spanish if that is your native tongue, is fine). A bare minimum point of the assignments are for you to make sure you understand what it is asking, and what the form of the output should be. If you do use AI to help answer questions after your own prompting (without copy-paste), explain what and how you did it. For some assignment questions, we may restrict the use of AI more – it will be explicitly stated in the assignment. AI is never permitted on the Bonus questions for assignments.

If you are caught using excessive AI for an assignment (or project), you will get a 0, and will not be able to drop that score in your final grade. If this happens twice, you will fail the course, and it will constitute a cheating policy violation.

Kahlert School of Computing Cheating Policy

The Kahlert School of Computing has instituted a two strikes and youre out cheating policy, meaning if you get caught cheating twice in any KSoC classes, you will be unable to take any future SoC courses. https://handbook.cs.utah.edu/2024-2025/CS/Academics/misconduct.php

If a student is caught cheating on a homework or the project, they will receive a failing grade for the course. For a detailed description of the university policy on cheating, please see the University of Utah Student Code: http://www.regulations.utah.edu/academics/6-400.html.

Talking and working together on tests is not permitted, and will be swiftly enforced. Talking without an instructor or TA present during a test will result in confiscation of the test (on the spot) and it will not be graded; a 0 score will be given.

Students with Disabilities

The University of Utah seeks to provide equal access to its programs, services, and activities for people with disabilities. If you need accommodations in this class, reasonable prior notice needs to be given to the Center for Disability Services, 162 Olpin Union Building, 581-5020 (V/TDD). CDS will work with you and the instructor to make arrangements for accommodations.

Addressing Sexual Misconduct

Title IX makes it clear that violence and harassment based on sex and gender (which Includes sexual orientation and gender identity/expression) is a civil rights offense subject to the same kinds of accountability and the same kinds of support applied to offenses against other protected categories such as race, national origin, color, religion, age, status as a person with a disability, veterans status or genetic in-formation. If you or someone you know has been harassed or assaulted, you are encouraged to come speak to the School of Computing Advisors and/or to the Title IX Coordinator in the Office of Equal Opportunity and Affirmative Action, 135 Park Building, 801-581-8365, or the Office of the Dean of Students, 270 Union Building, 801-581-7066. For support and confidential consultation, contact the Center for Student Wellness, 426 SSB, 801-581-7776. To report to the police, contact the Department of Public Safety, 801-585-2677(COPS). More information is available at https://safeu.utah.edu.

Behavior in Class

All students are expected to maintain professional behavior, according to the University of Utah Student Code. Students should read the Code carefully and know that they are responsible for the content. According to Faculty Rules and Regulations, it is the faculty responsibility to enforce responsible classroom behaviors, beginning with verbal warning and progressing to dismissal from the class and a failing grade. Students have the right to appeal such action to the Student Behavior Committee.

Latex

I recommend using LaTex for writing up homeworks. It is something that everyone should know for research and writing scientific documents. This linked overleaf project (https://www.overleaf.com/read/gvhwtfkfvvrk#39dae7) contains a sample .tex file. You can copy this project into your own overleaf account to get started. It also has a figure .pdf to show how to include figures.