Syllabus

Instructor: Jeff M. Phillips. | 3442 MEB | http://www.cs.utah.edu/~jeffp

Class Meetings: Mondays and Wednesdays, 3:00pm – 4:20pm, WEB L104.

Course Web Page: http://www.cs.utah.edu/~jeffp/teaching/cs5140.html

Data mining is the study of efficiently finding structures and patterns in large data sets. 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, probability, and linear algebra are required mathematical tools for understanding these approaches.

Topics will include: similarity search, clustering, regression/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-based companies.

Upon completion, students should be able to read, understand, and implement many data mining research papers.

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. There is no specific language we will use. However, programming assignments will often (intentionally) not be as specific as in lower-level classes. This will partially 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 prerequisits are CS 3500 and CS 3130 and MATH 2270 (or equivalent), and CS 4150 is a corequisite.

In the past, this class has had undergraduates, masters, and PhD students, including many from outside of Computer Science. 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.

Grading

The grading will be 45% from homeworks and 45% from a project and 10% from tests.

We will plan to have 5 or 6 short homework assignments, roughly covering each main topic in the class. The homeworks will usually consist of an analytical problems set, and sometimes a light 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 small project. I will allow small groups to work together. 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/cs5140/project.pdf

There will be two tests, each covering roughly half the material in class. They will be open notes; you can bring in anything on paper. No computers or 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 Canvas by the start of class, specifically 2:45pm. Once the 2:45pm 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 doctors 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.

School of Computing Cheating Policy

The School of Computing has instituted a two strikes and youre out cheating policy, meaning if you get caught cheating twice in any SoC classes, you will be unable to take any future SoC courses. http://www.cs.utah.edu/~ald/cheating_policy.pdf

If a student is caught cheating on a homework or test, 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.

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.

Latex

I highly highly recommend using LaTex for writing up homeworks. It is something that everyone should know for research and writing scientific documents. This linked directory (http://www.cs.utah.edu/~jeffp/teaching/latex/) contains a sample .tex file, as well as what its .pdf compiled outcome looks like. It also has a figure .pdf to show how to include figures.