COMP 330: Tools and Models for Data Science

Description

This course is an introduction to modern data science. Data science is the study of how to extract actionable, non-trivial knowledge from data. The course will focus both on the software tools used by practitioners of modern data science, as well as the mathematical and statistical models that are employed in conjunction with such software tools. On the tools side, we will cover the basics of relational database systems, as well as modern systems for distributed computing based on MapReduce. On the models side, the course will cover standard supervised and unsupervised models for data analysis and pattern discovery.

Prerequisite: (MATH 211 or MATH 221) and COMP 215. Basically, students must have some mathematical sophistication, as well as reasonable programming skills. Most programming will be in Python and SQL (SQL is covered in the course) with a small amount of Java. Students without COMP 215 but with reasonable programming skills can request permission to enroll from the instructor.
Credits: 3
Instructor: Chris Jermaine (cmj4@rice.edu), **Co-Instructor:** Kia Teymourian (Kia.Teymourian@rice.edu)

A Note to Students Who Have Taken COMP 430

A common question I've received is: "I've taken COMP 430 [Database Systems] with you… will there be a lot of overlap in COMP 330? What is new here?" First off, there is *a lot* of new material compared to COMP 430; COMP 430 does not cover Hadoop, Spark, nor does it cover any of the many models or data analysis methods that we'll go into in depth in COMP 330. As far as what's repeated, about 3 weeks of class lectures will on databases, SQL, and the relational model will be review for you if you have taken COMP 430. Further, Assignment 1 in COMP 330 will be very similar to Assignment 2 from COMP 430 (that is, it will closely match in spirit A2 from COMP 430, though the questions and data will not be the same). Also, part of Assignment 4 from COMP 430 is very, very similar to Assignment 2 from COMP 330. Taken together, this means that about 20 to 25% of the material from COMP 330 will seem familiar enough to you to be boring, if you've already taken COMP 430. This is not ideal, and hopefully in the next few years as COMP 330 matures, this will be fixed. But for now, if you've taken COMP 430, some of COMP 330 will be repeated. Whether you choose to take 330 is up to you!

Textbook

There is no textbook for the class. All class material will be conveyed during lecture.

Meeting Times and Locations

Class will be held Monday, Wednesday, Friday from 10:00 to 10:50 in DCH 1075.

Registration

You're responsible for registering for COMP 330 with the university registrar. As stated above, I am open to enrolling students who do not meet the COMP 215 Prereq.

Syllabus

The tentative lecture schedule is:

1. COMP 330 introduction
2. Intro to relational databases 1
3. Intro to relational databases 2
4. The relational algebra
5. Declarative SQL 1
6. Declarative SQL 2
7. Declarative SQL 3
8. Imperative SQL 1
9. Imperative SQL 2
10. Hadoop and MapReduce
11. Hadoop programming (Java)
12. Spark
13. Spark programming (Python)
14. Other "Big Data" technologies
15. Python, NumPy, SciPy 1
16. Python, NumPy, SciPy 2
17. Intro to modeling: numerical vs. probabilistic vs. Bayesian 1
18. Intro to modeling: numerical vs. probabilistic vs. Bayesian 2
19. Optimization basics: Gradient descent
20. Optimization basics: Newton's method
21. Optimization basics: Expectation maximization
22. Optimization basics: MCMC
23. Intro to supervised learning
24. Linear regression and generalized linear models 1
25. Linear regression and generalized linear models 2
26. Regularization
27. SVM and the kernel trick
28. Intro to unsupervised learning
29. K-means / K-medoids
30. Mixture of Gaussians and Gaussian EM
31. Matrix factorization
32. Matchbox model
33. Text: Latent semantic indexing
34. Text: Topic models
35. Association rule mining and the Apriori algorithm

36. Maximal association rule mining

Communication

The class will have a Piazza forum for all day-to-day communication:

https://piazza.com/rice/fall2015/comp330/home

It is expected that if you have a technical question on an assignment or an upcoming exam, you will post it to the forum rather than sending an email to the instructors This guarantees a fast response and means that everyone can benefit from the question and the answer. In general, only inquiries of private or personal nature should be made directly to the instructor ("I need to go out of town on Oct 22nd, can I have an extra day..."). Everything else should be posted on Piazza. You'll get faster feedback from the group than you can get from your instructors.

If you have any communication of a more personal nature and wish to contact the instructors of the class, please send email to Chris and Kia, and include the word "330" in the subject line. Please realize that we get a lot of random email, so if you do not include 330 in the subject line, your email will likely be ignored.

Assignment handouts and turnins, as well as your grades, will be on Owlspace. Everything else will be on Piazza.

Grading and Evaluation

Your grade is based upon a set of programming assignments (80% of your grade; each is worth 10% of your grade, except for the last one, which is worth 20%), and two midterms (10% of your grade each). Your numeric grades will be published to you in OwlSpace.

Final grades are based on the numeric grades, where 90-100 is an A, 80-89 is a B, and so forth. We reserve the right to apply a "curve" to change this, but only for the better. That is, if you've gotten 90%, you're guaranteed at minimum an A- for your final grade, but you might do better.

Assignments

This is an assignment-oriented class. There will be seven programming assignments, all completed individually—except for the last assignment, which will be completed in teams of two. All programming assignments will be in SQL (A1 and A2), Java (A3), or Python (A4, A5, A6). You can complete A7 using whatever language you wish.

The approximate assignment dates and due dates for the assignments will be:

A1 (SQL programming): out Monday, Aug 31, in Thursday, Sep 10
A2 (Linear regression in SQL): out Friday, Sept 11, in Friday, September 18

A3 (Hadoop Programming [Java]): out Friday, Sep 18, in Monday, Sep 28
A4 (Intro to Spark Programming [Python]): out Wednesday, Sep 30, In Friday, Oct 9
A5 (Matrix factorization on Spark): out Monday, Oct 12, in Wed, Oct 21
A6 (Gaussian EM on Spark): out Friday, Oct 23, in Monday, Nov 2
A7 (Final project: classification task): out Friday, Nov 6, in during finals week

Midterms

There will be one one-class midterm during the week of October 13th (we'll determine the exact timing later in the semester) and a second during the last week of classes (again, we'll determine the exact timing later in the semester). Each is worth 10% of your grade.

Lateness

Assignments must be turned in by 11:55PM (5 minutes before midnight) on the day that they are due. You can turn in an assignment up to 24 hours late, in which case you receive a 10% penalty (that is, 10 points are subtracted from an assignment that is worth 100 points), or up to 48 hours late, in which case you receive a 20% penalty. Assignments turned in after that are not accepted. Please note that your turnin time is whatever OwlSpace says, and your turnin is whatever you turn into OwlSpace, **no exceptions**. Because we have so many people in the class, no extensions will be given. Be safe; submit early and often!

If you do not show at a midterm, you won't receive any credit. The only exception is if you get written permission (via email) from Chris to take one of the midterms late (or early). We'll be fairly understanding of midterm conflicts (due to job interviews, athletic team commitments, and other important life events) but permission to take the midterm at a different time must be obtained at least one week before the midterm, **no exceptions**.

We kept on saying **no exceptions**, but there are exceptions in very extreme circumstances, with proper documentation. For example, if you obtain a doctor/dentist note stating that you were so ill at the due date/time that you could not reasonably be expected to meet the deadline, it is possible to get an extension.

Regrade Requests

These must be made within **one week** of an assignment/midterm being returned, during Chris' office hours, to Chris in person. Sending an email does not constitute a regrade request. When you talk to Chris, he'll help you understand whether you've got a legitimate request. If you do, then you'll write that request down formally, print it on paper, and hand it to Chris. Chris will batch these, and them periodically and issue final grade adjustments in bulk for everybody.

Academic Misconduct

In a programming class, there is sometimes a very fine line between "cheating" and acceptable and beneficial interaction between peers. Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. Our goal here is to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way---visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as StackOverflow.

As far as going to the web and using Google, we will apply the "two line rule". Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the "two line rule" inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them. Furthermore, you should **cite your sources**. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

Any violations of these rules will be reported to the Honor Council. Just don't do it!

Students with Disabilities

Students with disabilities should contact the course instructor and Disability Support Services regarding any accommodations that they may need.