STAT 339: PROBABILISTIC MODELING AND MACHINE LEARNING (FALL 2021)

Locations and Times. King 337, MWF 2:30-3:20pm
Instructor. Colin Reimer Dawson (*they/them*)
Office. King 204
Email. `cdawson@oberlin.edu`
Course Website. `http://colindawson.net/stat339/`

Office Hours.
Mondays 3:30-4:20pm (King 204, drop-in, but appts have priority)
Wednesdays 9:00-9:50am (King 204, by appt only)
Thursdays 4:00-5:30pm (King 203, drop-in group office hour)
Fridays 3:30-5:00pm (King 204, drop-in, but appts have priority)

Course Description. This course is an introduction to the field of **machine learning** from the particular perspective of **probabilistic reasoning**. We will discuss **fundamental principles** of machine learning and probabilistic reasoning, as well as specific **models and algorithms** used to do **classification**, **prediction**, **clustering**, and **hidden variable modeling**.

Rather than cover as many different models and algorithms as possible, we will focus on **general principles** of model-building and probabilistic reasoning as illustrated through a **relatively small set of common models and algorithms**.

The overarching goal of the course is for your to strengthen your ability to **reason probabilistically**, and to **equip you to learn** about and/or develop new models and algorithms to solve problems.

Applications will be selected from a variety of domains.

Learning Goals. By the end of this course, you should be able to:

- Explain the difference between **supervised and unsupervised learning**, and between **generative vs discriminative** methods

Date: Last Revised October 3, 2021.

- Understand the **properties, relative strengths, and limitations** of some of the most common probabilistic machine learning models used for **classification, regression, clustering and segmentation**
- **Represent probabilistic models using directed graphs**, and use these graphs to understand under what conditions two variables in the model are dependent or independent
- **Implement and test algorithms** to update the parameters of these models using real and synthetic data in a common computer language equipped with standard scientific computing libraries
- **Choose appropriate models and machine learning algorithms** to solve real problems using real data from a domain of interest

Who This Course is For. The course is designed to appeal to students **interested in modern data science** from a variety of backgrounds, but who have a **minimum foundation in basic programming and calculus**.

This likely includes (among others!) math majors with statistics interest, computer science majors with data science / artificial intelligence interest, natural / social science majors (e.g., psychology, neuroscience, economics, biology, physics) with a methodological/statistical bent, and even humanities majors who have some math background and who are interested in quantitative methods (for, e.g., text analysis).

Note that this course is **quite different in its approach to machine learning** from the other machine learning course offered at the college in the Computer Science department (CSCI 374), and it will **not be redundant to take both**.

It is important to note that we will be getting "into the weeds" of the underlying mathematical and statistical theory underlying the methods we will be using. Although this is not a "proof"-heavy course, there will be a lot of manipulation of mathematical formulas via calculus and algebra, so **expect to read and write a lot of math notation**, and to be **implementing methods in code on a level which is close to the math**: that is, although we will use probability and linear algebra libraries and will not necessarily need to work with data structures much more complicated than lists and arrays, we will not be using "black box" machine learning libraries, but rather implementing the equations we derive.

Prerequisites. The main prerequisites for the class are familiarity with **partial derivatives** and integrals of functions of multiple variables (MATH 231), and familiarity with **basic programming concepts**, such as variables, functions, arguments, loops, and arrays, at the level of CSCI 150 or similar.

Note: You may use whatever programming language you are most comfortable in, but the **solution sets I provide for coding problems will be in Python**. Other languages geared toward scientific computing, such as R, or MATLAB/Octave, would also be reasonable choices, though in some cases these languages are slower than Python at executing computation intensive machine learning algorithms. **I don't recommend using a lower level compiled language** such as Java or C/C++, as it is more time consuming to write and debug code in these languages.

In terms of additional mathematics, we will rely heavily on **probability**, and at times **linear algebra**, but I am **not expecting you to have taken courses** in these subjects; we will go through the main ideas that we will need.

There is **no longer a statistics prerequisite** for this class, although previous experience with statistical thinking is certainly helpful.

If you are unsure whether your background is suitable for this course, **don't hesitate to talk to me**.

Materials

Main Textbook. The main textbook is

- *A First Course in Machine Learning* (2nd edition), by Simon Rogers and Mark Girolami.

You will want the 2nd edition, as it includes several new chapters not in the first edition. We will aim to cover nearly the entire book.

Supplementary Texts. Supplementary readings will come from the following sources, all of which are freely available in pdf online from the authors' websites:

- *Bayesian Reasoning and Machine Learning*, by David Barber (`http://web4.cs.ucl.ac.uk/staff/D.Barber/textbook/171216.pdf`)
- Chapter 1 of *Machine Learning: A Probabilistic Perspective* by Kevin Murphy (`https://www.cs.ubc.ca/~murphyk/MLbook/pml-intro-22may12.pdf`)

Most topics in the course are covered by both the Rogers and Girolami (FCML) and the Barber (BRML) books. In most cases, the treatment in BRML is more advanced and technical, and should be considered optional reading if you are interested in going into more detail; however, there are a few places where FCML does not cover a topic at all.

Software. You may use any programming language(s) you like to write code; scientific computing languages with good support for matrix computations and plotting capabilities are strongly recommended. If possible, I recommend Python, with the support of the `matplotlib` or `seaborn` libraries for plotting, the `numpy` and `scipy` libraries for numerical computation, and the `pandas` library for keeping data organized.

The first (optional) homework assignment will include a chance for those without (much) programming experience to learn some basics about functions, arguments, variables, and arrays and a refresher on those ideas for others who want it.

Miscellany

Laptops in Class. I strongly discourage the use of electronic devices in class, unless there is an in-class activity where a laptop is needed. For note-taking, take handwritten notes if you are physically able to do so: there is evidence that writing by hand improves your cognitive processing of the material.

Communicating With Me. I have created a Slack workspace for the class, at `stat339f2021.slack.com`, which you will receive an invitation to join (if you haven't already). This workspace serves as a convenient way to organize communication between you, me, and your peers.

If you need to contact me outside class time or office hours, sending me a PM on Slack is the best way to ensure a prompt response. If you don't hear back from me within a day or two, don't hesitate to follow up, but **don't expect a response from me the same night if you message me the night before an assignment is due**. I have life and family commitments too!

Accommodations. If you require accommodations of any kind in order for you to do your best work in this class, please let me know as early as possible, and consult as well with the Office of Disability Services (ODS). By college policy, **all requests for accommodation require documentation from ODS.**

Honor Code. The Oberlin College Honor Code formalizes the idea that all work that you submit is your own and that you have given credit to the ideas and work of others when you incorporate them. You will be asked to write and sign the honor pledge on each graded assignment that you hand in. The honor pledge reads: "I have adhered to the Honor Code in this assignment."

What it means to adhere to the honor code depends on context. I describe what it means to follow the honor for each assignment type below.

I take the Honor Code very seriously. This means two things: First, I presume a relationship of mutual trust will not try to police your behavior. However, if I do have reason to believe that this trust has been betrayed, I treat this as a very serious matter, and will not hesitate to involve the Honor Code committee in the matter. I will do my best to remind you of the concrete expectations for each assignment as they come up, but it is ultimately your responsibility to make sure you understand them.

More information about the honor code can be found on the web at the Dean of Students site:
`https://www.oberlin.edu/dean-of-students/student-conduct/academic-integrity/students`

Graded Work

The final grade will be based 50% on homework (9 problem sets, with the lowest out of the first 5, and the lowest out of the last 4 dropped), 20% on a final project, 20% on a takehome midterm, and 10% on participation and engagement, as assessed through short reflections on the day's material.

Homework (50%). There will be an optional (ungraded) primer assignment to get up to speed with some basic prerequisite material, including calculus and coding (specifically scientific computing), and nine graded problem sets. The problem sets are **due electronically on Wednesday nights** by midnight.

Problem sets consist of a mix of **written (conceptual) questions**, **mathematical calculations and derivations**, and **computer implementation**. In some cases I will provide you with partial code in Python; if you choose to use a different language to complete your assignment, you can either translate it, or find a library that does the same thing.

The problem sets will require *significant* time to complete; expect to spend around 10 hours per week on them, though you may be able to do them in less time if your background exceeds the minimum prerequisites for the course. In any case, **do not wait until the day before they're due to start**.

Graded problem sets should be **completed in pairs or threes**. Each official group should turn in one copy of the assignment.

I will **not be able to grade every problem on every assignment**, but will "spot check" them, and give detailed feedback on 1-2 problems per set. I will provide my code and solutions to all problems, however.

Honor Code: You may freely discuss homework assignments with each other, whether or not you are working together; however **each pair must turn in their own original code and writeups; you may not copy or "paraphrase" anyone else's work.** In addition, it is expected that both members of a pair contribute substantially to all pieces of an assignment, namely, the math, coding, and writing for each problem. **The intent is *not* for you to split the assignment in half, each doing part of it, but rather to discuss and collaborate with each other as you work on it**.

Participation and Engagement (10%). After each class I will ask you to write a **short (∼2 sentence) reflection on that day's material via a Slack message**. These are intended to be very much open-ended, and can consist of questions, comments, observations, or a mix of these. The idea is just to get your mind to revisit and consolidate the ideas.

Reflections are **due by midnight on each class day** (so, on Wednesdays you may have both a reflection and a problem set due). You can miss up to three reflections without affecting your grade. Reflections submitted after the deadline but before the following class receive half credit. If you missed a class, indicate that in your message, and write something about the reading or something you're still thinking about from a previous class instead.

Honor Code: Be honest. You may not submit a reflection on anyone else's behalf.

Takehome Exam (20%). There will be **one take-home midterm exam covering the foundational material** of the first 2/3 or so of the semester. The questions will consist of a combination of conceptual and mathematical components, and will not involve coding. This should take significantly less time to do than the problem sets, but will still be substantial: maybe 3-5 hours or so. It is currently scheduled to be **distributed on Friday, December 3rd** and **due Wednesday, December 8th**.

Honor Code: The takehome exam **must be done individually, without consulting any other individuals** apart from me, but you may consult your notes, or other course materials I provide.

Final Project and Presentations (20%). For the final project, you will work in a **small group of your choosing (2-3 people), defining your own classification, clustering, and/or prediction question(s), and implementing and comparing at least three different machine learning approaches** to your problem.

At least one of the three should be a **standard "off the shelf" probabilistic method**, at least one should be **either an original probabilistic model or one of the "advanced" nonparametric models** discussed in Part VI of the course, and the third may be a **non-probabilistic method, or an additional model from one of the other categories**.

Your writeup should be in the form of a **conference-style paper as might be submitted to a machine learning conference** such as the International Conference on Machine Learning (ICML), the Conference on Neural Information Processing Systems (NeurIPS), or Uncertainty in Artificial Intelligence (UAI). These are **typically 6-10 pages** in length (single spaced), including figures, and include an **introduction section** in which you describe your problem and discuss **related work**, a **methods or model section** in which you describe the techniques you are using, a **results section**, including graphical and numerical summaries of the performance of your chosen methods, and a **discussion section**, reflecting on the relative **strengths and weaknesses** of your chosen methods as revealed by your results and perhaps theoretical/practical considerations. You will need to decide on ways to **measure performance** that supply "apples-to-apples" comparisons across methods.

The scheduled final exam period is **Sunday, January 23rd from 2-4 P.M.**, during which time your group will give a **10 minute presentation** of your projects. **Final writeups are due at the presentation**.

Honor Code: Free collaboration with your team members is of course required. Describing the work that other people did before you is an important part of intellectual inquiry, and you must **give credit and cite sources** for any data, code, or ideas that did not originate within your team. **This includes paraphrases** as well as direct quotations. It is fine (and indeed you are encouraged) to call other people's open-source code in your own code. All members of the group must make **approximately equal overall contributions** to the project, though it is possible, for example, for one person to do more coding and another to do more writing, etc. All group members must participate in the presentation.

Approximate Topic Outline

See the course website: `http://colindawson.net/stat339/schedule` for a tentative (and periodically updated) schedule of topics and associated readings.