Unsupervised Machine Learning

MACS 40800
Spring Quarter 2021

Meeting Days & Times:

- Blend of asynchronous and synchronous (via Zoom)
- Synchronous Sessions:
- Classroom: Zoom

	Dr. Philip Waggoner	TA1	TA2
Email	pdwaggoner@uchicago.edu	TA1	TA2
Office Hours	Th 9-11am	TA1	TA2
Github	pdwaggoner	TA1	TA2

Introductory Remarks

Welcome to *Unsupervised Machine Learning*! In this class, we take a deep dive into the exciting, though often less-appreciated world of unsupervised machine learning. Very generally, unsupervised learning is concerned with exploring and learning from *unlabeled* data, such that we are not interested in predicting or forecasting some target output. Rather, though techniques and approaches vary pretty widely, the common goal of an unsupervised approach to learning from data is to recover the non-random structure of the full data space.

Flowing from a broad framework comprised of dimension reduction, clustering, and data mining, we cover a lot of ground from preprocessing data and feature engineering to algorithms like uniform manifold approximation and projection, t-SNE, deep autoencoders, and many more. To deepen the value of the applied aspect of the course, other best practices involving data visualization and functional R programming will also be covered throughout.

Course Objectives

By the end of the course, students should be able to:

- Understand what is meant by *unsupervised* machine learning, from terms and mathematical properties, to application, evaluation, and validation
- Understand when and where it makes sense to apply targeted unsupervised machine learning techniques to learn from data
- Understand strengths and weaknesses of different unsupervised techniques, and thus when and where is makes sense to use one over another
- Apply a suite of unsupervised learning algorithms in the R programming language
- Generate reproducible research reports of projects addressing substantive problems

- Complete a final group project that reflects all aspects of the course, from theoretical considerations to proper application of models and methods

Prerequisites: Though not formally required, it is highly recommended that students have some level of experience with computational methods. Such experience could come from, e.g., MACS 33002: *Introduction to Machine Learning* or other training in statistical methods. Additionally, some level of programming experience in a flexible computing environment is expected, which could come from, e.g., MACS 30500: *Computing for the Social Sciences*. Of note, direct experience with R is highly recommended, but not required.[1] I will provide an R crash course during the first week to ensure everyone is on the same page, but this hardly substitutes for a deeper knowledge of R. If unsure of your ability to succeed in the course, I recommend reaching out to me sooner rather than later to discuss.

Course Structure

For the first time, due to the horrid Coronavirus, I will be teaching this course online. Hopefully this will be the first *and* last time to do so. But as such, I ask for patience from each student as there will likely be a few bumps along the way. Still, I am quite hopeful that it will be a great course given the exciting nature of the topics. That said, each week with a few exceptions, there will be a lecture with a bit of code scattered throughout on one day during a synchronous meeting. The other day of the week will be asynchronous, and consist of a "challenge" of some kind. The idea here, as described more below, is for the challenges to take the place of traditional problem sets. I think this will be both a better assessment of understanding in more "real time", while also being slightly less of a burden on students. More details on these below.

Our primary technologies will be:

1. **Zoom**: Synchronous meetings, lecture, code
2. **Canvas**: Course organization, challenge submissions, presentation submissions
3. **Github Classroom**: Project submissions

Text & Materials

Required

1. Sign up for a free GitHub account at: https://github.com. Github classroom will be used to submit all problem sets. For those unfamiliar with using version control for assignment submissions, see Dr. Soltoff's excellent guide: https://cfss.uchicago.edu/faq/homework-guidelines/#homework-workflow
2. Locally download R (https://cran.r-project.org/mirrors.html) and RStudio (https://rstudio.com/products/rstudio/download/#download)
3. Waggoner, Philip. 2020. *Unsupervised machine learning for clustering in political and social research*. New York: Cambridge University Press. Free PDF version: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3693395

Recommended

1. (ESL) Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2012. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer. Free PDF version: https://web.stanford.edu/~hastie/ElemStatLearn/. A great book to have and be aware of, if you aren't already.

[1] I primarily teach and research in R. While this is the case, though, if students are more comfortable with other languages such as Python or Julia, you are welcome to use them in the course.

2. Kennedy, Ryan, and Philip Waggoner. 2021. *Introduction to R for social scientists: A Tidy programming approach*. Chapman & Hall/CRC Press. See course files on Canvas for free PDF version. Sure I am biased, but this is still a helpful text for those less familiar with R in a computational social science context.
3. VanderPlas, Jake. *Python Data Science Handbook*. O'Reilly. A great reference text for Python translations of many operations we will perform in R.

Assessment

All final grades are rounded to the nearest decimal (e.g., 88.38% = 88.4%). I use the following grading scheme to determine your final grade: A (93-100), A- (90-92), B+ (88-89), B (83-87), B- (80-82), C+ (78-79), C (73-77), C- (70-72), D+ (68-69), D (63-67), D- (60-62), F (0-59). I deduct a letter grade per day any assignment is late (the "clock" starts the minute after the deadline; e.g., for assignment x due at submission time, $t = 5$, $t_x \in \{5:01, \ldots, 11:59\} = B$).

There are five components to final grades:

- 1. Project Proposal (10%)
- 2. Group Presentation (15%)
- 3. Final Report (25%)
- 4. Weekly Challenges (35%)
- 5. Midterm Exam (15%)

It's important to highlight at the outset that a large portion of your grade for the course is a group project, which will take shape in three major stages: proposal, presentation, and report. The reason for a group project is that this emulates the real world of computational problem-solving and "data science," whether academically (e.g., co-authored projects and grants) or in industry (e.g., data scientist on a team of data scientists). The expectations, as with real world projects, are that you first, specialize on a given component of the project (e.g., one member is the *programmer*, another is the *theoretical architect*, etc.) in order to maximize efficiency and productivity. And second, submit and present a *single* final project that addresses a substantive challenge or problem of interest in your preferred domain, whether sociology, political science, computer science, neuroscience, etc. In sum, the goal of the final project is three-fold. *First*, to apply the concepts learned in this class creatively, which will help you understand material more deeply. *Second*, to design and work on a unique project in a team, which is something that you *will* encounter, if you haven't already. *Third*, along with the opportunity to practice and the satisfaction of working creatively, you can use this project to enhance your portfolio or CV.

There is *no* perfect project. Whether your project was an empirical "success" or "failure" is significantly less important to me than your process of getting to the final stage, whatever that may look like. I am interested in your workflow, e.g., what steps were involved in creating your solution, what did you learn, how did your solution perform, what are limitations to be addressed in future iterations, and so on. Your grade will be calculated based on your thoroughness and honesty in reporting.

As the group project comprises a large portion of your final grade, I expect you will be thinking about and working on these projects from Day 1 of the course. Please come talk to me to workshop, problem-solve, and address any questions throughout the life cycle of your project; I want you to succeed and am very available as a result.

There are relatively few guidelines for the subject matter of your final project. My only requirement is that your methods incorporate at least one *unsupervised* machine learning technique covered in the course. Of course you may use more than one, or you may also pair an unsupervised method with a supervised method. For example, you could use a clustering algorithm to derive class labels for previously unlabeled data, which could then be fed to a supervised classifier to predict classes of unseen/new data. Or alternatively, you could compare the many approaches to clustering data (hard, soft, hierarchical, density, spectral, mixtures, etc.) on a single set of data, and offer a critical discussion of the influence of different algorithms on our understanding and conceptualization of data structure. And so on.

You will be working in a team of ~2-4 people, which you will organize on your own. Groups must be organized by the end of the first week of class (Friday). If you are having trouble finding a group, *reach out to me sooner rather than later* to request help and I will assign you to a group. If I don't hear from anyone, I will assume all students are in groups. As time will go very quickly, you are strongly encouraged to **keep in regular communication with your group members**, whether in person, virtually (e.g., Slack, email, text, etc.), or both throughout the quarter. If you have any concerns working with someone in your group, please talk to me for accommodations.

1. Project Proposal (10%): The proposal is where your group will (unsurprisingly) pitch its project idea. **Project proposals are due on Friday of Week 3 by 11:59 pm on Github Classroom.** The goal of this stage is two-fold. First, it is a good initial, low-stakes test of your group dynamics. As this is where you will introduce the problem, methods/solution, data, and so on, such a task requires consolidation and universal contribution to the overall vision of the project; in a word, *teamwork*. And second, the goal of the proposal is to get early-stage feedback from me on the feasibility of the project, both regarding your ability to complete it within the time frame of the course, as well as whether your project is in the scope of the class.

Some particulars: These should be a maximum of 2 pages, excluding references (can be via footnote or reference list). Final submissions should be a single PDF for the whole group submitted via Github Classroom. Use 11- or 12-point standard font. Select any reference style you wish (Chicago, APA, MLA, etc.); just be correct and consistent.

2. Group Presentation (15%): All presentations must be recorded, and include *all* group members. Consider recording a Zoom call in "grid view" with a shared screen for the slides. Presentations are due to the appropriate *Canvas* module on **Sunday of Week 9 by 11:59 pm**, which is prior to the final week of class.

Students have from Monday - Wednesday during the final week 10 of the class to view **all** presentations. Though required to view all, students must only only respond/comment in Canvas on at least **two** other groups' presentations. Comments can include anything from questions to suggestions, but must be constructive and appropriate.

Though I don't have a specific set of guidelines for organizing presentations, good presentations typically include some combination of the following:

- Introduce the topic to a *general* audience with a motivating anecdote, paradox, or example
- Summarize your process, workflow, and your approach or method
- Highlight the key findings, but avoid going into the weeds as time is limited
- "Appendix" set of slides with more detail at the end in case you are questioned on specific details (though we won't have a live Q&A session)

The presentation should be *no more than 8 minutes* in length. All members of the group must participate in the presentation.

I like to give presentation awards to both encourage greater quality, and also democratize class participation. Two awards will be given for presentations:

- **Best Presentation**
- **Most Creative Project**

Awards are determined by class voting, where each student will fill out a ballot via Google forms, and distributed by me via email at the end of the class. Voting is on a scale from 1-10 for each of the 2 categories, where 10 is best. I will collect the ballots as they are filled out and submitted. Upon summing the points across each award category, the highest point-getter wins the respective award. Only 1 winner is allowed for each category, and the winner gets a 5% bonus (half a letter-grade) on the final project report, which can be > 100%. In case of a tie, I will select a winner at random, each with an equal probability of winning (i.e., a single Bernoulli trial). You may vote for yourself, if you're truly blown away by your own creative genius.

3. Final Report (25%): The final project report is the distillation of your group's efforts over the course of the quarter. **Only one report** should be submitted by your group, with each person's name on the report. **Final project reports are due on Thursday of Week 10 to Github Classroom by 11:59 pm.**

To reinforce the *applied* part of this course, students must follow the *Proceedings of the National Academy of Sciences* (PNAS) article format. This is a two-column, single-spaced paper with a **maximum of 6 pages**:

- Follow the PNAS formatting guidelines here: https://www.pnas.org/page/authors/format
- Use their LaTeX template here: https://www.overleaf.com/latex/templates/template-for-preparing-your-research-report-submission-to-pnas-using-overleaf/fzcbzjvpvnxn
- For those unfamiliar with TeX, and in need of a bit of extra help, see my help files from a workshop I taught a few years ago, here: https://github.com/pdwaggoner/LaTeX-Workshop. Of course, start by Googling errors/questions, then reach out to your colleagues, and ultimately reach out to me if you're still stumped after giving troubleshooting a fair effort.

Distilling a complex project into 6 pages, though two columns, is an extremely difficult task. This requires crisp and clear writing, an ability to focus on the most important aspects of the methods and the results, as well as an ability to tie substantive patterns back to the "real world" (i.e., the *so what?* question). Throughout the quarter, I will highlight best practices in high level, professional research and writing of this sort. But it's up to each student to leverage the course to his/her advantage. In practice, this might look like using the initial *Project Proposal* assignment as testing grounds for homing your writing abilities as the proposal is also required to be extremely brief at 1-2 pages. Further, the value of using PNAS as the standard is that all of you will likely, if you haven't already, come into contact with PNAS as they publish articles in virtually every substantive domain, e.g., the physical sciences, social sciences, medical sciences, and so on. In sum, this approach should help you on several dimensions, and I hope you take and deepen these skills in your careers as you progress beyond what we cover in the course.

Data & Code: Each group should include all data and code in reproducible form (e.g., markdown of any flavor, etc.). This should go in the private repository where final papers are submitted, yet code should *not* be included in the final report itself; it may be included in an attached appendix of any length.

4. Weekly Challenges (35%): As noted above, the weekly challenges are designed to replace traditional problem sets. These will be asynchronous, and students will have our normal class period to complete the challenge. That is, most weeks, the second class 1 hour 20 minute class day of the week will be used for students to, on their own, complete the challenge. These challenges are applied, meaning they will comprise both code and substantive questions. For the most part, there will be anywhere from 5-10 questions and usually a bonus question or two. The challenge will go live at the start of the normal class day and time, and then close down at the conclusion of that class period, which again, is 1 hour and 20 minutes. The main goal of these challenges is to allow for a more "real time" assessment of students' retention of a given week's content. Like a problem set, though much narrower in scope, these are like miniature, timed problem sets. A further goal of these is to assess students' abilities to work quickly, and efficiently.

Though the writing should be checked for grammatical and formatting issues where possible, I care much more about the quality of the solutions. Spend however much time you need to ensure first that the questions are answered to the best of your ability. And as a second order goal, make sure they look fairly clean and are easy to follow and grade.

Note: A completed submission will be a **single** PDF (rendered from `.Rmd` or Jupyter Notebooks), and will include: a response to the question, presentation of any equations, code, and output (e.g., plots, tables, etc.) inserted *directly below the response to the question.* In other words, no appendices are necessary for weekly challenges. Just write and run the code, and produce the output in-line as you go. **A single and complete, rendered PDF submitted to the appropriate week's assignment page on Canvas is all that is required.**

5. Midterm Exam (15%): Though not technically at the midpoint of the term, during the second class day of week 7, students will complete a *synchronous* midterm exam. The first class day of week 7 will be canceled and should be used as an independent study day.

During the exam period, which will span the length of the normal class period (1 hour and 20 minutes), students will log in to the class Zoom session for the day, open the midterm exam, complete it within the normal class period, and submit to Canvas once finished. Students may simply log out after they have submitted the exam as though they were leaving the classroom. No need to engage at all. Importantly, students will use *only* their personal computer to complete the exam. Submission of the completed exam should be at the appropriate Canvas assignment page.

On the honor system, no external tools (internet, notes, books, etc.) are allowed, nor is communication with anyone else of any kind; just you and your computer. All exams will be checked for cheating, which if uncovered, will result in a 0% for the exam and a report to the Dean of Students. I don't anticipate issues, of course, but need to make the expectations perfectly clear.

As this midterm exam is technically "synchronous," I ask only that students log into our normal class Zoom session for the day and remain present until their exam is submitted. This will allow students to approach me with any questions in real time (via chat) in an effort to emulate as closely as possible the normal classroom setting. Such a format also encourages a bit of accountability, though diminished by distance to be sure. More details will come the closer we get to the exam day.

Diversity & Inclusion

The University of Chicago is committed to diversity and rigorous inquiry from multiple perspectives. The MAPSS, CIR, and MACSS programs share this commitment and seek to foster productive learning environments based upon inclusion, open communication, and mutual respect for a diverse range of identities, experiences, and positions.

Any suggestions for how we might further such objectives both in and outside the classroom are appreciated and will be given serious consideration. Please share your suggestions or concerns with your instructor, your preceptor, or your program's Diversity and Inclusion representatives: Darcy Heuring (MAPSS), Matthias Staisch (CIR), and Chad Cyrenne (MACSS). You are also welcome and encouraged to contact the Faculty Director of your program.

This course is open to all students who meet the academic requirements for participation. Any student who has a documented need for accommodation should contact Student Disability Services (773-702-6000 or disabilities@uchicago.edu) and the instructor as soon as possible.

Course Outline

Note: The schedule below is tentative and may change. If changed, students will be made aware with sufficient time to adjust.

Week 1: A soft introduction to unsupervised machine learning

- Day 1 (Tuesday) Course intro, EDA, and visualization
- Day 2 (Thursday) Computational foundations: R programming, feature engineering for missing data
- **Friday: Research teams organized by 5 pm CDT**

Week 2: Mining data for natural patterns

- Day 1 (Tuesday) Association rule mining
- Day 2 (Thursday) Weekly challenge due at the end of class to Canvas

Week 3: A classic approach to dimension reduction

- Day 1 (Tuesday) Principal components analysis
- Day 2 (Thursday) Weekly challenge due at the end of class to Canvas
- **Friday: Project Proposals Due to Github Classroom by 11:59 pm CDT**

Week 4: A linear approach to manifold learning

- Day 1 (Tuesday) Locally linear embedding
- Day 2 (Thursday) Weekly challenge due at the end of class to Canvas

Week 5: Neural networks for clustering & dimension reduction (sort of. . .)

- Day 1 (Tuesday) Self-organizing maps
- Day 2 (Thursday) Weekly challenge due at the end of class to Canvas

Week 6: Deep unsupervised learning

- Day 1 (Tuesday) Restricted Boltzmann machines & Autoencoders
- Day 2 (Thursday) Weekly challenge due at the end of class to Canvas

Week 7: Midterm Exam

- Day 1 (Tuesday) **Class Canceled: Study Day**
- Day 2 (Thursday) Midterm Exam (normal class period; *synchronous*)

Week 8: Dimension reduction for visualization

- Day 1 (Tuesday) t-distributed stochastic neighbor embedding (t-SNE) & Uniform manifold approximation and projection (UMAP)
- Day 2 (Thursday) Weekly challenge due at the end of class to Canvas

Week 9: Clustering (***no challenge this week***)

- Day 1 (Tuesday) Hierarchical, hard, and density-based partitioning
- Day 2 (Thursday) Soft and probabilistic partitioning
- **Sunday: Final Group Presentations due on Canvas by 11:59 pm CDT**

Week 10: **No Regular Class Meetings: View/Comment on Presentations; Submit Final Report**

- **Monday - Wednesday: View *all* presentations; Respond to at least *two***
- **Thursday: Final Group Project Report due on Github Classroom by 11:59 pm CDT**