CS 109: Probability for Computer Scientists

Summer 2021

<table>
<thead>
<tr>
<th>Instructor:</th>
<th>Alex Tsun, Tim Gianitsos</th>
<th>Time:</th>
<th>MWF 5:30pm – 6:50pm PST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email:</td>
<td>{alextsun,tgianit}@cs</td>
<td>Location:</td>
<td>Online!</td>
</tr>
</tbody>
</table>

(Syllabus Last Updated: 2021-06-18, 16:05:50)

Course Resources:

1. Course Website: https://web.stanford.edu/class/cs109/
2. Gradescope, Edstem, Zoom Links, Calendar, and Lecture Videos all linked from the website above.

Announcement: You should regularly check the class web site for announcements and other information, including the most up-to-date information on problem sets and errata. The class web page will also have the schedule of topics to be covered and links to other class materials, including class handouts. If you have any personal questions, please email both of us directly. Any other non-sensitive questions about course content should be posted on our discussion forum.

Office Hours: See calendar on website.

Textbooks:


Prerequisites: CS 103, CS 106B or X, and multivariate calculus at the level of MATH 51 or CME 100 or equivalent. Here is a quick rundown of some of the mathematical tools we’ll be using in this class: calculus (integration and differentiation), linear algebra (basic operations on vectors and matrices), an understanding of the basics of set theory (subsets, complements, unions, intersections, cardinality, etc.), and familiarity with basic proof techniques (including induction).

Why is CS 109 Important?

While the initial foundations of computer science began in the world of discrete mathematics (after all, modern computers are digital in nature), recent years have seen a surge in the use of probability as a tool for the analysis and development of new algorithms and systems. As a result, it is becoming increasingly important for budding computer scientists to understand probability theory, both to provide new perspectives on existing ideas and to help further advance the field in new ways.

Probability is used in a number of contexts, including analyzing the likelihood that various events will happen, better understanding the performance of algorithms (which are increasingly making use of randomness), or modeling the behavior of systems that exist in asynchronous environments ruled by uncertainty (such as requests being made to a web server). Probability provides a rich set of tools for modeling such phenomena.
and allowing for precise mathematical statements to be made about the performance of an algorithm or a system in such situations.

Furthermore, computers are increasingly often being used as data analysis tools to glean insights from the enormous amounts of data being gathered in a variety of fields; you’ve no doubt heard the phrase “big data” referring to this phenomenon. Probability theory is now used as one of the primary methods for designing new algorithms to model such data, allowing, for example, a computer to make predictions about new or uncertain events. In fact, many of you have already been the users of such techniques. For example, most email systems now employ automated spam detection and filtering. Methods for being able to automatically infer whether or not an email message is spam are frequently rooted in probabilistic methods. Similarly, if you have ever seen online product recommendation (e.g., “customers who bought X are also likely to buy Y”), you’ve seen yet another application of probability in computer science. Even more subtly, answering detailed questions like how many buckets you should have in your a hash table or how many machines you should deploy in a data center (server farm) for an online application make use of probabilistic techniques to give precise formulations based on testable assumptions.

Our goal in this course is to build foundational skills and give you experience in the following areas:

1. **Understanding the combinatorial nature of problems**: Many real problems are based on understanding the multitude of possible outcomes that may occur, and determining which of those outcomes satisfy some criteria we care about. Such understanding is important both for determining how likely an outcome is, but also for understanding what factors may affect the outcome (and which of those may be in our control).

2. **Working knowledge of probability theory**: Having a solid knowledge of probability theory is essential for computer scientists today. Such knowledge includes theoretical fundamentals as well as an appreciation for how that theory can be successfully applied in practice. We hope to impart both these concepts in this class.

3. **Appreciation for probabilistic statements**: In the world around us, probabilistic statements are often made, but are easily misunderstood. For example, when a candidate in an election is said to have a 53% likelihood of winning does this mean that the candidate is likely to get 53% of the vote, or that if 100 elections were held today, the candidate would win 53% of them? Understanding the difference between these statements requires an understanding of the model in the underlying probabilistic analysis.

4. **Applications**: We are not studying probability theory simply for the joy of drawing summation symbols (okay, maybe some people are, but that’s not what we’re really targeting in this class), but rather because there are a wide variety of applications where probability allows us to solve problems that might otherwise be out of reach (or would be solved more poorly without the tools that probability can bring to bear). We’ll look at examples of such applications throughout the class.

5. **An introduction to machine learning and theoretical computer science**: Machine learning is a quickly growing subfield of artificial intelligence which has grown to impact many applications in computing. It focuses on analyzing large quantities of data to build models that can then be harnessed in real problems, such as filtering email, improving web search, understanding computer system performance, predicting financial markets, or analyzing DNA. The use of randomized algorithms and probabilistic data structures has also become prevalent, as they are usually have more elegant implementations than their deterministic counterparts, and have more efficient time and/or space complexity.

**Goals for Summer 2021:**
We fully recognize that this experience cannot replace what we normally have on campus, and that many of you have personal situations that may change throughout the quarter. That being said, we are determined
to reach the following course goals to the best of our ability: (1) To maintain the intellectual rigor of the CS 109 curriculum while providing flexible ways for you to learn, and (2) To foster and maintain human connections and a sense of community throughout this online course. We have adjusted the typical course structure to meet these goals.

Many of these changes that we are making are designed to help foster intellectual nourishment, social connection, and personal accommodation—through accessible, asynchronous content for diverse access, time zones, and contexts, and optional, synchronous discussion to learn together and combat isolation. Please bear in mind that all of us are experiencing remote, online classrooms at this scale for the first time, and we may have to adapt throughout the quarter. Everyone needs support and understanding in this unprecedented moment, and we are here to listen to you. Thanks and welcome to CS 109. (Credit for this wording goes to Brandon Bayne from UNC - Chapel Hill.)

Tentative Course Outline (based on the course textbook):

1. Combinatorial Theory
2. Discrete Probability
3. Discrete Random Variables
4. Continuous Random Variables
5. Multiple Random Variables
6. Concentration Inequalities
7. Statistical Estimation
8. Statistical Inference
9. Applications to Computing

Lectures:
We will be holding live lectures remotely via Zoom on Mondays, Wednesdays, and Fridays, 5:30PM – 6:50PM Pacific Time. Lecture recordings will be posted on Canvas shortly after lecture ends. Short concept checks for each lecture will be due 15 minutes before the next lecture; more on this in a later section.

Grading Breakdown:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Sets (5)</td>
<td>60%</td>
</tr>
<tr>
<td>Concept Checks</td>
<td>16%</td>
</tr>
<tr>
<td>Mini-Quizzes (3)</td>
<td>24%</td>
</tr>
</tbody>
</table>

Problem Sets

- There will be 5 problem sets, equally weighted. There will be a written part and a coding part involved in each.
- The written parts must be typed using \LaTeX, and submitted to Gradescope. A tutorial will be provided and recorded by the end of the first week.
- You must show your work: at a minimum of 1-2 sentences per question, but ideally as much as you would need to explain to a fellow classmate who hadn’t solved the problem before. Be concise. A correct answer with no work is worth nothing, less than a wrong answer with some work.
- You must tag the question parts of your homework correctly on Gradescope. Failure to do so will result in a 0 on every untagged question. Please check your submission by clicking each question, and making sure your solution appears there. Tagging can be included or changed after the assignment is submitted and after it is due as well.
- The coding parts will be written in Python3, with no exceptions. This is because the coding parts will be autograded. There are no hidden tests, and you’ll have unlimited attempts. Whatever you see last on Gradescope for that section will be your grade.

- Regrade requests are due on Gradescope within one week of grades being published.

- It is okay to collaborate in coming up with solutions, but you must list all your collaborators at the top of each homework.

- For some particularly challenging PSets, we will allow groups of up to 2. The exact PSets for group and individual work are yet to be determined. You cannot have the same partner more than once. We will occasionally offer social sessions and surveys for finding partners.

- **(Edstem) Discussion Board Etiquette**

  1. A single thread for each question in each PSet will be created by the staff (e.g., PSet2 Q5). All clarification questions which are not personal should be posted in the corresponding thread to avoid duplicate questions.

  2. All new threads will require a staff member’s approval (but replying on a thread like mentioned above will not). If you have a specific/personal question (e.g., sharing a large part of the solution or your thought process), please make a private post.

- **Late Policy**

  1. Problem Sets 1-4: On-time problem sets will be given 2.5% extra credit. Each problem set will be accepted up to two days late without penalty, with a 25% deduction each day after. (After 6 days, problem sets will not be accepted as we will publish grades and solutions according to a fixed schedule.)

  2. Problem Set 5: Will not be accepted late, nor will it have an on-time bonus.

**Concept Checks**

- Immediately after each lecture, concept checks for the sections covered will appear on Gradescope.

- Concept checks are due at 5:15pm PT on Gradescope on the day of the next lecture (15 min before class).

- You can submit your answers as many times as you want; we will only grade the final submission. Correct answers will reveal the answer explanation; all other answers will not, so you can keep trying until you see the answer explanation.

- **Late Policy**

  1. After the initial deadline, you can submit your concept check up to a week later for partial credit (capped at 50%).

**Mini-Quizzes**

- We will have 3 mini-quizzes given throughout the quarter; dates will be posted on the course website.

- These will have a time limit of 2 hours, but to be completed in a certain 24-hour window.

- These are to be done individually, and will consist of some T/F, multiple choice, and short answers.

- Late quizzes will not be accepted, and will be recorded as 0%.
**Attendance Policy:** Regular attendance to lecture is essential and expected. This class is fast-paced, but we will spend the majority of lecture time solving problems and diving deep into concepts. To incentivize attendance, some problem set questions may be explicitly solved during lectures.

**Academic Honesty:** Lack of knowledge of the academic honesty policy is not a reasonable explanation for a violation. Each student is expected to do their own work on the problem sets in CS 109. Students may discuss problem sets with each other as well as the course staff. Any discussion of problem set questions with others should be noted on a student’s final write-up of the problem set answers. Excessive collaboration (i.e., beyond discussing problem set questions) can result in honor code violations. Questions regarding acceptable collaboration should be directed to the class instructor prior to the collaboration. It is a violation of the honor code to copy problem set solutions from others, or to copy or derive them from solutions found online or in textbooks, previous instances of this course, or other courses covering the same topics (e.g., STATS 116 or similar probability courses at other schools). Copying of solutions from students who previously took this or a similar course is also a violation of the honor code. Finally, a good point to keep in mind is that you must be able to explain and/or re-derive anything that you submit.

Violations of the above or any other issue of academic integrity are taken very seriously, and may be referred to the University. Please refer to the Allen School’s Academic Misconduct webpage for a detailed description of what is allowable and what is not.

**Accommodations:** Students who may need an academic accommodation based on the impact of a disability must initiate the request with the Office of Accessible Education (OAE). Professional staff will evaluate the request with required documentation, recommend reasonable accommodations, and prepare an Accommodation Letter for faculty. For students who have disabilities that don’t typically change appreciably over time, the letter from the OAE will be for the entire academic year; other letters will be for the current quarter only. Students should contact the OAE as soon as possible since timely notice (for example, at least a week before an exam) is needed to coordinate accommodations. Students should also send your accommodation letter to instructors as soon as possible. If you require additional, or different, accommodations specific to the Summer 2021 learning environment, please contact your disability adviser directly.

**Acknowledgements:** Syllabus wording largely influenced by Lisa Yan, Chris Piech, and Mehran Sahami from earlier offerings of the course.