CS 189 / 289 Introduction to Machine Learning
Fall 2024 Jennifer Listgarten, Saeed Saremi HW6

Due 11/20/24 11:59 pm PT

- Homework 6 consists of both written and coding questions.
- We prefer that you typeset your answers using LaTeX or other word processing software. If you haven't yet learned LaTeX, one of the crown jewels of computer science, now is a good time! Neatly handwritten and scanned solutions will also be accepted for the written questions.
- In all of the questions, **show your work**, not just the final answer.

Deliverables:

1. Submit a PDF of your homework to the Gradescope assignment entitled "HW 6 Write-Up". Submit your code to the Gradescope assignment titled "HW 6 Code". **Please start each question on a new page.** If there are graphs, include those graphs in the correct sections. **Do not** put them in an appendix. We need each solution to be self-contained on pages of its own.
   - In your write-up, please state with whom you worked on the homework. This should be on its own page and should be the first page that you submit.
   - In your write-up, please copy the following statement and sign your signature underneath. If you are using LaTeX, you can type your full name underneath instead. We want to make it *extra* clear so that no one inadvertently cheats.

     "I certify that all solutions are entirely in my own words and that I have not looked at another student's solutions. I have given credit to all external sources I consulted."
   - **Replicate all of your code in an appendix**. Begin code for each coding question on a fresh page. Do not put code from multiple questions in the same page. When you upload this PDF on Gradescope, *make sure* that you assign the relevant pages of your code from the appendix to correct questions.

1 Robot Localization with the Viterbi Algorithm

Consider the following semi-realistic model of a robot's motion: we say that the state variable X_t represents the robot's location in a grid-like environment. Suppose the grid has r rows and c columns. Then, there are $N = r \cdot c$ total states and the state space can be written as $\mathcal{S} = \{(i, j) \mid 0 \leq i < r,\ 0 \leq j < c\}$, i.e., each $X_t \in \mathcal{S}$ is represented by a tuple of row and column indices.

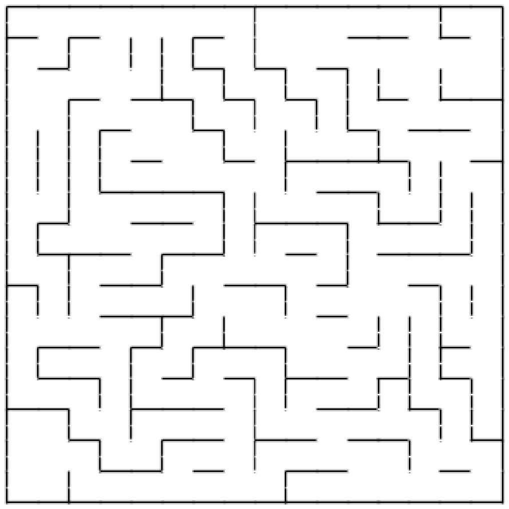


Figure 1: This is just one possible instance of an environment that is randomly generated.

This environment is enclosed by walls and has both horizontal and vertical walls scattered throughout between pairs of adjacent cells. Let NEIGHBORS(q) denote the set of empty grid cells that are adjacent to state q and are reachable from it (i.e., not blocked from q by a wall). Then, we model the robot's motion in this environment as a random walk where it transitions to one of its unblocked neighboring cells uniformly at random, i.e.,

$$P(X_{t+1} = q' \mid X_t = q) = \frac{1}{|\text{NEIGHBORS}(q)|} \text{ if } q' \in \text{NEIGHBORS}(q) \text{ else } 0$$

We don't know where the robot starts, so we will assume that the probability distribution π over the initial states is uniform; that is $P(X_0 = (i, j)) = 1/N$.

The robot has 4 sensors mounted on it, one pointing in each of the 4 cardinal directions. Each sensor returns a single bit output: 1 when a sensor reads a wall directly in front of it and 0 if it reads open space. The observation at a timestep t is a 4-bit sequence returning the sensor values from the left, right, up and down sensors respectively, **in that particular order**. For example, for the environment in the figure above,

- if the robot is present in the top left cell (i.e., $X_t = (0, 0)$), its sensors should read 1011 because there is a wall in the left, up and down directions.
- if the robot is present in the top right cell (i.e., $X_t = (0, 15)$), its sensors should read 0110 because there are walls in the right and up directions.

We note that the number of 0s that a sensor should read in state q is precisely $|\text{NEIGHBORS}(q)|$.

However, the sensors are faulty! Each sensor, independent of the others and the timestep, can fail with probability ϵ and incorrectly flip its output. Therefore, there are 2^4 possible observations that we can read off of the robot, each occurring with different probabilities: if there are d discrepancies between the true sensor reading and the returned sensor reading, the probability of this observation will be $(1-\epsilon)^{4-d}\epsilon^d$.

The true robot states X_t are hidden from us, but we can observe the faulty sensor readings, say denoted by Y_t at timestep t, every time the robot moves to a new state. As such, we can model the scenario above using a hidden markov model.

(a) Our goal is to predict the true robot hidden states from the observed sensor readings. Given a sequence of T observations $o_1, \ldots, o_T$, **implement the Viterbi algorithm to decode the most likely sequence of hidden states.** You have all the information required to compute the transition probabilities $P(X_{t+1} = q' \mid X_t = q)$ and observation emission probabilities $P(Y_t = o_t \mid X_t = q)$.

The starter code contains two files:

- `env.py`: This file encodes the dynamics of the environment. In particular, it defines the `Env` class which creates the underlying environment, tracks the true robot hidden state and emits sensor readings. You may call any pre-defined methods of the `Env` class but not access any of its instance variables from `hmm.py`.
- `hmm.py`: This is where all of your code should go.

Since your goal is to return the best set of hidden states, **don't forget to also backtrace your way through the viterbi probabilities** to find the most likely sequence. Feel free to reference **this handout on HMMs** for more information about the Viterbi algorithm.

Include your code in the appendix and select the appropriate pages when submitting to Gradescope. Also submit `hmm.py` to the Gradescope coding assignment.

(b) Now, we will vary the error parameter ϵ to values like 0, 0.05, 0.1, 0.2, 0.25 and 0.5. For each value of epsilon, we will generate and decode 100 sequences of 100 observations each. We also plot the cumulative accuracy (i.e., the accuracy of the first i predictions when compared to the first i hidden states, for each $i \in [1, 100]$, for a single trajectory), averaged over all 100 trajectories, for each ϵ. Attach this plot in your writeup. Describe any trends that you see in the plot: what happens when ϵ increases from 0 to 0.5? for a fixed epsilon, how does the cumulative accuracy change over time? Do these trends align with your intuition?

2 Graphical Model Potpourri

(a) Show that $a \perp\!\!\!\perp (b, c) \mid d$ implies $a \perp\!\!\!\perp b \mid d$.

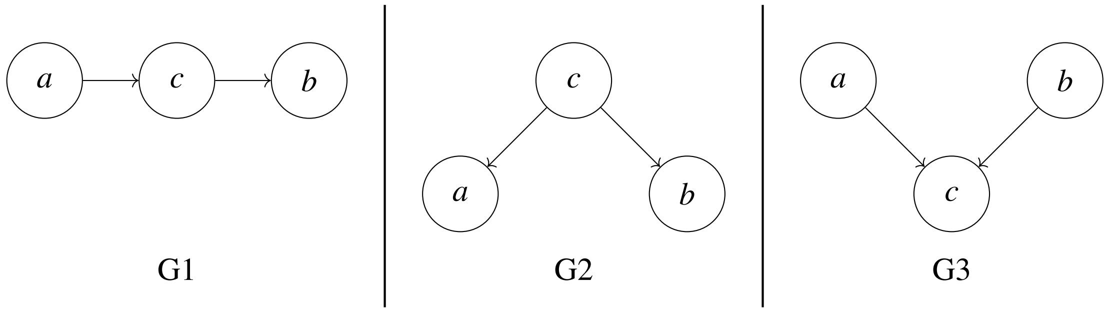
(b) Suppose you have a set of d binary random variables $\{X_1, ..., X_d\}$.

(i) Without making any independence assumptions, what is the minimum number of parameters needed to fully specify the joint distribution?
Hint: You need one fewer parameter compared to the number of outcomes, due to the constraint that the total probability should sum to 1.

(ii) Suppose the random variables are structured as a Markov chain, where each X_i only depends on X_{i-1}. What is the minimum number of parameters needed now?

(iii) How does the parameter complexity change going from no independence assumption to a Markovian assumption?

(c) Consider the following three graphical models with distinct structures.



Match each scenario to the most appropriate graphical model.

(i) A family's decision on where to travel (c) involves considerations such as the parents' work schedules (a) and the children's school holidays (b).

(ii) A record-breaking snowfall (c) spurs both ski resort bookings (a) and demand for winter clothing (b).

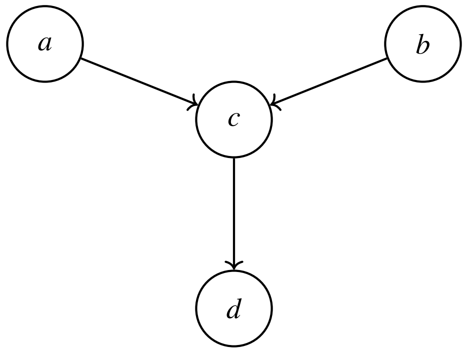
(iii) A person's workout routine (a) affects their energy levels (c), which then influences their work productivity (b).

(iv) The climate of a region (a) impacts the type of vegetation that grows (c), which in turn determines the wildlife population (b).

(v) A country's economic stability (c) affects both employment rates (a) and consumer spending habits (b).

(vi) A restaurant's popularity (c) hinges on both the quality of its food (a) and its social media presence (b).

(d) Consider the following directed graph, where none of the variables are observed.



(i) What is the joint distribution $p(a, b, c, d)$ in terms of the marginal and conditional distributions given in the graph?

(ii) Assuming that a, b, c, d are binary random variables, what is the minimum number of parameters required to specify the joint distribution?

(iii) Show that a and b are independent, or $a \perp\!\!\!\perp b$.

(iv) Assume that for any given node x in the graph, $p(x \mid \mathrm{pa}(x)) \neq p(x)$, where $\mathrm{pa}(\cdot)$ denotes the parent nodes of x. Show that when d is observed, a and b are no longer independent, or $a \not\!\perp\!\!\!\perp b \mid d$.

3 Langevin Dynamics Demo

For this question, go to the Google Colab notebook provided ***here*** [1] to complete the code.

In this notebook, we will walk through a simple demo for Langevin dynamics, where the goal is to sample from a distribution p(x) using only its score function ∇_xlogp(x). Here we assume a toy setting where p(x) is known. In most practical cases we only have access a dataset of samples $\mathcal{D} = \{x_0, x_1, \ldots, x_n\} \sim p(x)$, in which case we might use a technique called score matching to estimate the score function[2]. For more background, you can reference Chapter 14.3 on Langevin Sampling in Bishop and Bishop, 2024.

Recall that the Langevin update equation at timestep t with step size η and random noise $\epsilon \sim \mathcal{N}(0, I)$ is

$$x_{t+1} = x_t + \eta \nabla_x \log p(x) + \sqrt{2\eta}\epsilon$$

Deliverables: Complete the written questions below, which will walk you through the notebook. In addition, submit the .ipynb file to the code assignment.

(a) Complete the functions `langevin_update` and `sample_langevin`. Include a PDF export of the completed code in your write-up.

(b) Using the given default settings, visualize the samples over time. Include an image of the plots and describe what you see. Which is closer to the target shape, early or late timesteps? How does the variance of the samples change over time?
Note: To simplify the runtime and plotting, throughout this problem you will only run the Langevin sampler for a few iterations. In practice, however, you would typically run the sampler for longer (e.g., several thousand iterations), to ensure the Markov chain has converged.

(c) Now let's adjust the radius of the elliptical logpdf. Set `rx, ry = 1.0, 2.5`. How do the final samples compare with the samples for the circular logpdf? Does Langevin sampling fit better to the elliptical or the circular logpdf?
Hint: Even coverage over the entire target distribution is more desirable.

(d) Let's see if we can get a better fit to the elliptical logpdf in (c) by tuning each dimension of `eta`. What is a better selection of `eta`?

(e) Here let's move the center of the logpdf away from the origin. Set `rx, ry = 1.5, 1.5` and `cx, cy = 1.0, 1.0`. How do the final samples compare with centered logpdf? Does Langevin sampling fit better to the centered or off-center logpdf?

(f) For the off-centered logpdf in (e), let's try to get a better fit by tuning the initialization `init_particles`. What can you do to improve the initialization?

[1] https://colab.research.google.com/drive/1fvcWjRJYRLQM83RPKu-uQ4sqAPGTuuxA?usp=sharing
[2] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. NeurIPS 2019.

4 Honor Code

1. **List all collaborators. If you worked alone, then you must explicitly state so.**
2. **Declare and sign the following statement**:

   "I certify that all solutions in this document are entirely my own and that I have not looked at anyone else's solution. I have given credit to all external sources I consulted."

   Signature : ____

   While discussions are encouraged, *everything* in your solution must be your (and only your) creation. Furthermore, all external material (i.e., *anything* outside lectures and assigned readings, including figures and pictures) should be cited properly. We wish to remind you that the consequences of academic misconduct are *particularly severe*!