Project Member Application

SMAC

2025-26

Mathematics Club

Centre for Innovation

IIT Madras

Instructions

General Instructions

- Mention the following details at the start of your application.

Name:	Insti Nickname:
Roll No:	CGPA:
Phone (WA):	Email:

- Join the Aspiring Team Members WhatsApp group for further updates: Whatsapp Group
- The recommended font is a standard font size 11-13.
- The applications have to be submitted in PDF format, named as:
  <First_name>_<Roll_Number>_Mathematics_Club_PM_SMAC.pdf

  For example, `Atharva_CS23B096_Mathematics_Club_PM_SMAC.pdf`.
- You can upload the finished applications in this Google Form
- You may submit the completed application on or before **11:59 PM, 30/05/2025.**

Note:

- It is fine even if you don't answer all the questions.
- Focus on the compulsory questions before attempting the bonus questions.
- If you have any queries, you can reach out to the Project Leads anytime you want:
  - Atharva Moghe (CS23B096): 9491561608
  - Ojas Phadake (CH22B007): 9422345405

Contents

HR

0. Tell us a bit about yourself. Assume you are a master of flexing and proceed.
1. Why do you want to join this project? What skills or qualities do you possess that make you suitable for the work involved?
2. Mention all PoRs/activities you are planning to take this year. Weekly, how much time do you think you will be able to commit to this project? How much time will you commit to other PoRs and academics?
3. What would keep you motivated throughout the period of two semesters? How would you ensure that you are consistent in contributing to the project throughout the tenure?

§1 The Tale of Tired Mr. Markov

In the heart of old Russia, where the winters are long and the tea is strong,
lived a brilliant but weary mathematician named **Andrey Andreyevich Markov**.

Now, Markov had a problem. A big one.

You see, every day, people would come up to him and ask questions like:
"Professor Markov, what are the odds that I'll get married next year?"
"Professor Markov, if I roll this dice, what's the expected value of my happiness?"

And every time, he'd have to take out his quill and scroll and scribble numbers, probabilities, and equations.

"Is life really just numbers?" he muttered.

§1.1 A Random Walk into the Future

Then, he had an idea. A simple, yet powerful one.

"What if... the future wasn't decided by everything in the past?"
"What if... the next step only depends on where I am right now?"

This, dear reader, was the moment **Markov Chains** were born.

He imagined a drunkard (as all great mathematicians do at some point) stumbling down a street.
At each lamp post, the drunkard flips a coin:

Heads → take a step right
Tails → take a step left

Where the drunkard goes next doesn't depend on how he got there — just on where he is.
That's a **Markov Chain**:

"The future state depends only on the current state, not the history."

§1.2 Stochastic Processes and Markov Chains

A stochastic process is a collection of random variables, representing the evolution of a system over time under uncertainty. Examples include Markov chains, Poisson processes, and Brownian motion. Stochastic modeling involves using mathematical models that incorporate randomness to represent systems or processes that evolve over time in an uncertain or probabilistic way.

A stochastic process that has the property

$$P_{1|n-1}(x_n, t_n \mid x_{n-1}, t_{n-1}; x_{n-2}, t_{n-2}, \dots; x_1, t_1) = P_{1|1}(x_n, t_n \mid x_{n-1}, t_{n-1})$$

is called a **Markov process**. i.e. Conditional probability depends only on the current values X_{n-1} and not any of the previous values.

The transition probability is defined as: $P_{1|1}(x_n, t_n \mid x_{n-1}, t_{n-1})$

A Markov process is fully defined by:

1. $P_1(x_1, t_1)$
2. $P_{1|1}(x_n, t_n \mid x_{n-1}, t_{n-1})$

For example, for $t_1 < t_2 < t_3$:

$$\begin{aligned} P_3(x_3, t_3; x_2, t_2; x_1, t_1) &= P_{1|2}(x_3, t_3 \mid x_2, t_2; x_1, t_1) \times P_2(x_2, t_2; x_1, t_1) \\ &= P_{1|1}(x_3, t_3 \mid x_2, t_2) \times P_{1|1}(x_2, t_2 \mid x_1, t_1) \times P_1(x_1, t_1) \end{aligned}$$

Therefore,

$$P_n(x_1, t_1; x_2, t_2; \ldots; x_n, t_n) = P_1(x_1, t_1) \prod_{i=1}^{n-1} P_{1|1}(x_{i+1}, t_{i+1} \mid x_i, t_i)$$

§1.3 The Problem

A library system by the name of **XYZ Books** has come to town. They have 4 outlets: **A**, **B**, **C**, and **D**. Consider the fact that every book borrowed from them at the start of the day must be returned before the end of the day.

However, you have the flexibility of returning it to *any* of the outlets.

After a couple of months, the owner observed a general trend:

- Of the books borrowed from library A, 25% are returned to A, 35% end up at B, 25% at C, and 15% at D.
- Of the books borrowed from library B, 20% end up at A, 40% are returned to B, 30% at C, and 10% at D.
- Of the books borrowed from library C, 10% end up at A, 20% at B, 60% are returned to C, and 10% at D.
- Of the books borrowed from library D, 5% end up at A, 25% at B, 30% at C, and 40% are returned to D.

Questions

1. Draw the transition diagram for the following process explained above.
2. Read about transition matrices here. Write the transition matrix for the above question.
3. Consider, the following scenario, at the first day of observation, you see the following: 20 % of the books are at A, 20 % of the books are at B, 25 % of the books are at C, 35 % of the books are at D. Read up on what a **state vector** means, and now with only performing **matrix operations**, find out of the state vectors on day 1,2,3.

4. Reason out the matrix operations you used above.
5. Find the state vector on days 100, 150, 200 preferably by writing a code in **Python/C++** to perform the matrix operation you chose to do. Submit the code and the outputs.

§2 The Secrets of Mr. Markov a.k.a. Hidden Markov Models

Mr. Markov did not want all his information to be public, hence he came up with a new model called the Hidden Markov Model. Read more about this here. You can read more up here also.
Please note that going through all the given references is not at all necessary, however we are attaching what we feel could prove beneficial in doing the application.
In case you did not understand that, don't worry :), we have an example for you.

§2.1 A simple example

Suppose we want to determine the average annual temperature at a particular location on earth over a series of years. To make it interesting, suppose the years we are concerned with lie in the distant past, before thermometers were invented. Since we can't go back in time, we instead look for indirect evidence of the temperature.
To simplify the problem, we only consider two annual temperatures, 'hot' and 'cold'. Suppose that modern evidence indicates that the probability of a hot year followed by another hot year is 0.7 and the probability that a cold year is followed by another cold year is 0.6. We'll assume that these probabilities held in the distant past as well. The information so far can be summarized as

	H	C
H	0.7	0.3
C	0.4	0.6

(1)

where H is "hot" and C is "cold".
Also suppose that current research indicates a correlation between the size of tree growth rings and temperature. For simplicity, we consider three tree ring sizes: small, medium and large (S, M, L respectively). The probabilistic relationship between temperature and tree ring sizes is given by

	S	M	L
H	0.1	0.4	0.5
C	0.7	0.2	0.1

(2)

The matrices A (transition), B (observation), and π (initial state) are:

$$A = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}, \quad B = \begin{pmatrix} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{pmatrix}, \quad \pi = \begin{pmatrix} 0.6 & 0.4 \end{pmatrix} \tag{3}$$

Now consider a particular four-year period of interest from the distant past, for which we observe the series of tree rings S, M, S, L. Letting 0 represent S, 1 represent M and 2 represent L, this observation sequence is $\mathcal{O} = (0, 1, 0, 2)$. We want to determine the most likely state sequence of the Markov process given the observations.

§2.2 Notation

- T = length of observation sequence
- N = number of states in model
- M = number of observation symbols
- $Q = \{q_0, q_1, ..., q_{N-1}\}$ = distinct states
- $V = \{0, 1, ..., M-1\}$ = set of possible observations
- A = state transition probabilities
- B = observation probability matrix
- π = initial state distribution
- $O = (O_0, O_1, ..., O_{T-1})$ = observation sequence

For the temperature example of the previous section—with the observations sequence given in $\mathcal{O}$ —we have $T = 4$, $N = 2$, $M = 3$, $Q = \{H, C\}$, $V = \{0, 1, 2\}$ (where we let 0, 1, 2 represent "small", "medium" and "large" tree rings, respectively). In this case, the matrices A, B and π are given in (3).
The matrix $A = \{a_{ij}\}$ is $N \times N$ with

$$a_{ij} = P(\text{state } q_j \text{ at } t+1 \mid \text{state } q_i \text{ at } t)$$

and A is row stochastic. Note that the probabilities a_{ij} are independent of t. The matrix $B = \{b_j(k)\}$ is an $N \times M$ with

$$b_j(k) = P(\text{observation } k \text{ at } t \mid \text{state } q_j \text{ at } t).$$

As with A, the matrix B is row stochastic and the probabilities $b_j(k)$ are independent of t. The unusual notation $b_j(k)$ is standard in the HMM world. An HMM is defined by A, B and π (and, implicitly, by the dimensions N and M). The HMM is denoted by $\lambda = (A, B, \pi)$. Consider a generic state sequence of length four

$$X = (x_0, x_1, x_2, x_3)$$

with corresponding observations

$$O = (O_0, O_1, O_2, O_3).$$

Then π_{x_0} is the probability of starting in state x_0. Also, $b_{x_0}(O_0)$ is the probability of initially observing O_0 and a_{x_0,x_1} is the probability of transiting from state x_0 to state x_1. Continuing, we see that the probability of the state sequence X is given by

$$P(X, O) = \pi_{x_0} b_{x_0}(O_0) a_{x_0,x_1} b_{x_1}(O_1) a_{x_1,x_2} b_{x_2}(O_2) a_{x_2,x_3} b_{x_3}(O_3).$$

Consider again the temperature example given above with observation sequence $O = (0, 1, 0, 2)$, as given in $\mathcal{O}$. Using the above equation, we can compute, say,

$$P(HHCC) = 0.6(0.1)(0.7)(0.4)(0.3)(0.7)(0.6)(0.1) = 0.000212.$$

Questions

6. Given the model $\lambda = (A, B, \pi)$ and a sequence of observations $\mathcal{O}$, find $P(\mathcal{O} \mid \lambda)$. Here, we want to determine a score for the observed sequence $\mathcal{O}$ with respect to the given model λ. (Hint: Look at what forward algorithm or α-pass algorithm here.
7. Given $\lambda = (A, B, \pi)$ and O, find optimal state sequence. Two interpretations:
   a) Dynamic Programming approach finds single best path
   b) HMM approach maximizes expected number of correct states. (Hint: Look at what backward algorithm here)

§3 Algorithms related to HMMs

§3.1 Baum Welch Algorithm

Also known as the forward-backward algorithm, the Baum-Welch algorithm is a dynamic programming approach and a special case of the expectation-maximization algorithm (EM algorithm). Its purpose is to tune the parameters of the HMM, namely the state transition matrix A, the emission matrix B, and the initial state distribution π_0, such that the model is maximally like the observed data.

There are a few phases for this algorithm, including the initial phase, the forward phase, the backward phase, and the update phase. The forward and the backward phase form the E-step of the EM algorithm, while the update phase itself is the M-step. Read up about Baum-Welch algorithm here.

Questions

8. Explain to us (with derivations) the forward and backward procedures as mentioned in the above link in detail. Make sure that you understand every step you write.
9. *Brownie Question*
   - Tell the names of a few fields in which this algorithm is widely used.
   - What is an expectation-maximisation algorithm and where does it crop up?
   - Explain few uses of this algorithm in places we did not know ourselves :)

§3.2 Viterbi Algorithm

Read up Vitterbi Algorithm here.

The Viterbi algorithm is a dynamic programming method used to find the most likely sequence of hidden

states (the Viterbi path) in a Hidden Markov Model (HMM), given a sequence of observed events.
At each time step, it keeps track of the highest probability of any path that ends in a particular state, using:

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i) \cdot a_{ij}\right] \cdot b_j(o_t)$$

Where:

- $\delta_t(j)$: max probability of the best path ending in state j at time t
- a_{ij}: transition probability from state i to j
- $b_j(o_t)$: probability of observing o_t in state j

A backtracking step is used at the end to reconstruct the most likely path.

Questions

10. Explain what you understood from the algorithm given in the link. Additionally, follow the **psuedocode** and use any language of your choice to **code** up the algorithm.

§4 Simulations (Bonus)

Questions

On any of Python/Matlab, simulate the following events:

11. Simulate tossing a coin with probability of heads p
12. (Coin Toss Simulation) Write code to simulate tossing a fair coin to see how the law of large numbers works.
13. Let X have a standard Cauchy distribution.

$$F_X(x) = \frac{1}{\pi}\arctan(x) + \frac{1}{2}$$

Assuming you have $U \sim \text{Uniform}(0, 1)$, explain how to generate X. Then, use this result to produce 1000 samples of X and compute the sample mean. Repeat the experiment 100 times. What do you observe and why?