CSE 150A / 250A: Probabilistic Reasoning and Learning — Final Project

Due: (Final submission on 11th December, 2025 by 11:59 PM, Pacific Time, via Gradescope)
No Grace Period for project deliverables.

1.1 Overview and Purpose

The following document contains all the guidelines and requirements for the course project[1]. Please read carefully before you begin.

The final project is an opportunity for you to apply the concepts learned in this course to real-world data and problems. You are encouraged to connect theory with practice by **designing probabilistic models**, **performing inference and learning**, and **analyzing how well your model captures patterns in the data**. Finally, you need to write a Project Report on the Problem you have explored.

Projects should focus on the **application of algorithms to data**, **comparison of alternative modeling approaches (literature review or ablation study)**, and **critical analysis of results**. You are welcome to use existing datasets or collect your own, provided your modeling approach is clearly motivated and grounded in probabilistic reasoning.

To be a bit more specific, the general classes of algorithms you have (or will) learn in this course are:

1. **Bayesian Networks (Inference)**: Exact Inference, Approximate Inference
2. **Bayesian Networks (Learning)**: Maximum Likelihood Estimation (MLE), Expectation-Maximization (EM)
3. **Hidden Markov Models (HMMs)** (Within Bayesian Networks): Viterbi Algorithm (Inference), Baum-Welch Algorithm (Learning)
4. **Reinforcement Learning**: Policy/Value Iteration (Model-based); TD-learning/ Q-learning (Model-free)

(!! You will learn Reinforcement Learning Algorithms towards the end of the course. Prefer to choose a project based on other topics if you're not already familiar with RL)

[1] We acknowledge the use of AI assistance (ChatGPT, Gemini) in generating draft content for the project guidelines. All final decisions, structure, and content were curated and verified by the instructional team.

1.2 Note on Performance vs. Understanding

Important: The goal of this project is not to achieve state-of-the-art performance. Projects will be evaluated on the quality of *probabilistic reasoning*, the soundness of *modeling assumptions*, and the correctness of *inference and learning methods* rather than on raw metrics.

What matters most is your ability to:

- Clearly explain your modeling choices and independence assumptions.
- Implement and analyze inference or learning algorithms correctly.
- Interpret the model's results and discuss its limitations thoughtfully.

Strong projects demonstrate a deep understanding of probabilistic modeling, even if their quantitative performance is modest.

1.3 Notes:

1. You **may use external libraries or packages**, but you are responsible for understanding how they work. TA feedback will primarily focus on the content and structure of your project report. All code must be written in Python.
2. The **same Generative AI policy used for the homework assignments applies to the project.**
   (a) You may use AI tools ask questions, get hints, or debug, but don't copy AI output verbatim. Adapt anything you use and make sure you can explain it yourself.
   (b) If you have any questions or worries about whether your collaboration constitutes a violation of academic integrity, feel free to ask us on Piazza.
   (c) If you use Generative AI tools, please include a brief statement in your report describing how these tools were used.
3. If you are **interested in being assigned a random group**, please fill this Google Form
4. You may use Open-Source Code as long as you provide proper attribution.
5. Select Qualitative Analyses / Quantitative Metrics that **make the most sense for your project**. You may consult with TAs to aid your choice.
6. Students **across sections can work together** (eg. a team of 3 250A students + 2 150A students is fine).
7. The **entire team will receive a single grade/score** for the Project Deliverables. Make sure to add your group members (via "View or Edit Group") for your deliverables on Gradescope.
8. You may revise your project topic or details **before Milestone 1** if absolutely necessary, but we encourage you to stick to your initial choices. Use the group formation period to perform exploratory or quick preliminary analyses on potential problem statements.
9. We **advise you to use LaTeX and the NeurIPS format**, but any **neatly typed solution** will do. The report should generally contain the sections listed in the "Project Structure" section. Speak with your assigned instructional staff member if you have a good reason to deviate from this structure.
10. For Gradescope Code submissions - you can **zip your code and upload them to the portal** (Multiple files are okay). You could also provide a Github link as a footnote (but the Professor and TAs must be able to view your code!)
11. You **can choose a topic other than the 4 broad categories listed in the Overview**, as long as it remains in the spirit in the course. Discuss with TAs if you want suggestions or feedback.
12. Use a **consistent style for your citations**. Popular ones include APA, IEEE, ACM etc.

1.4 Project Timeline and Deliverables

You will work on the project in teams of at least **four** and at most **five members**.

The project is divided into multiple milestones to help students stay on track and ensure steady progress. Each milestone includes specific goals and expected interactions with TAs for feedback and guidance.

Feedback for **Milestones 1 & 2** will be provided during the week of submission. Please adhere to the deadlines to ensure timely feedback.

Milestone 3 will be graded according to the rubric. The **Final Submission** will give you an opportunity to improve on your score.

Please Note: There is no Grace Period for any deliverable.

Stage	**Due Date**	**Task Description & End Deliverable**
Group Formation	Thu, Nov 6 (11:59 PM)	Explore Problems, Datasets, and Form a Group, Select Project Topic / Problem Statement. Submit project keywords and dataset description **Deliver:** Group Name, Members, Project Description (few words including Dataset Domain, Method). Each group will be assigned a member of the instructional staff who will serve as your primary point of contact for feedback and guidance throughout the project. Signup Sheet Link
Milestone 1	Mon, Nov 17 (11:59 PM)	Begin analysis on the selected problem and dataset using chosen methods. Communicate with your assigned instructional staff member (via the signup sheet) to confirm your approach and resolve any issues. **Deliver:** Project Plan on Gradescope (1 page. Problem Description, Dataset Source, Methodology)
Milestone 2	Mon, Nov 24 (11:59 PM)	Prepare the first draft of the project report. Attend Office Hours / meet with assigned staff to obtain feedback and incorporate suggestions. **Deliver:** Submit the first draft of the project report and code on Gradescope.
Milestone 3	Mon, Dec 1 (11:59 PM)	Revise and polish the report based on TA feedback. Ensure figures, analysis, and writing are complete and coherent. **Deliver:** Updated project report and code on Gradescope
Final Project Submission	Thu, Dec 11 (11:59 PM)	**Deliver:** Submit the final version of the project report and code on Gradescope.

Table 1: Student Project Timeline and Deliverables

1.5 Project Structure

The project write-up is expected to be prepared as a single report per team, It should be typed (preferably using the general style of **NeurIPS conference** - a LaTeX template is available on Overleaf). The report should have atmost **7 content pages**. The 7-page limit applies to main content only (Sections 1–6). References and any appendices do not count toward this limit.

1. **Problem Description**
   - Clearly state the problem or question your project addresses. Why is this problem important?
   - Motivate its relevance to probabilistic reasoning or learning.
2. **Data Sourcing and Processing**
   - Describe where your data comes from (e.g., Kaggle, UCI, public APIs).
   - Explain any preprocessing steps, such as handling missing data, discretization, or feature selection. Why are these processing steps important for your problem?
3. **Modeling and Inference**
   - Specify your probabilistic model (e.g., Bayesian Network, HMM, MDP).
   - Define assumptions, parameters, and the structure of dependencies. Relate this to the dataset and the problem you are trying to solve.
   - Explain the inference or learning algorithms you implement (e.g., Exact Inference, Gibbs Sampling, EM, Policy Iteration).
4. **Results and Discussion**
   - Explore and compare multiple configurations (e.g., different hyperparameters, hidden states, or network structures). Summarize how these changes affect performance and stability, and explain any observed differences.
   - Present quantitative results (e.g., likelihoods, accuracies, expected returns).
   - Discuss qualitative insights—what aspects of the data does the model capture or miss, and do the learned CPTs or latent states align with intuition?
   - Comment on convergence and scalability: (eg. Was training stable? Would more data help, and at what computational cost?)
5. **Conclusion**
   - Summarize key findings and model performance.
   - Discuss limitations and propose potential extensions or improvements.
6. **Reflections & Contributions**
   - Any suggestions or advice for future students trying such a project?
   - For each team member, include a 1–2 sentence summary outlining their specific tasks and contributions. Each member should also include a brief individual reflection on their personal learning from the project.
   - If you've used Gen AI. Write a note on how you've used it in your project.
7. **References**
   - Please cite or reference all external papers, datasets, and tools used in your project.

1.6 Example Project Ideas

Below are example directions to inspire your own project. You may adapt these or propose your own problem. Work with your assigned instructional staff member to ensure that you have a solid project. You can incorporate any suggestions to further improve it.

(A) Learning Student Performance Models
Predict student grades from features such as study habits, attendance, and stress levels. Use the UCI "Student Performance" dataset. Construct a Bayesian Network with latent variables (e.g., motivation), learn CPTs using Maximum Likelihood or EM, and apply exact inference. Analyze how hidden factors influence outcomes. Hypothesize and answer interesting questions based on inference - (Does access to internet improve a student's grade? Or are they independent?)

(B) Approximate Inference in Large Bayesian Networks
Investigate the accuracy and speed trade-offs of approximate inference methods (e.g., Rejection Sampling, Likelihood weighting, Gibbs sampling) on a large synthetic or real BN. Compare against exact inference where feasible.

(C) Discovering Coding and Non-Coding Regions in a Bacterial Genome
Implement the Baum–Welch algorithm to learn a Hidden Markov Model that distinguishes between coding regions (e.g., C0, C1, C2) and non-coding regions (NC) in a bacterial genome. Use the learned transition and emission probabilities to interpret the latent biological structure, and apply the Viterbi algorithm to infer the most likely sequence of coding and non-coding regions. Data: For instance, see the Bacterial Genome Dataset (NCBI).

(D) Reinforcement Learning for Blackjack
Implement a reinforcement learning agent to learn strategies for the card game *Blackjack*. You may use model-based methods (e.g., policy iteration, value iteration) or model-free methods (e.g., Q-learning, temporal-difference learning) to estimate optimal policies. Visualize convergence behavior and compare policy performance over time. Optionally, compare learned strategies to common human heuristics (e.g., "hit below 17, stand otherwise") and discuss similarities or differences.

1.7 Evaluation Rubric

Projects will be graded according to the following criteria. This rubric is designed to apply broadly across modeling, inference, and reinforcement learning projects. Each criterion is worth up to 3 points (total = 15 points possible).

Criterion	**Excellent (3)**	**Satisfactory (2)**	**Needs Work (1)**
Problem & Dataset	Clear, well-motivated problem. Dataset described with sources, limits, and relevance.	Problem mostly clear. Dataset described but some ambiguity or missing details.	Problem unclear or poorly motivated. Dataset missing key details or source.
Methodology / Experimental Design	Method fully specified and justified. Contains enough detail for replication.	Method described and mostly appropriate but some steps unclear or rationale missing.	Method vague or inappropriate; insufficient information to replicate.
Analysis & Interpretation	Results analyzed thoroughly with appropriate reasoning. Limitations and alternative explanations discussed.	Basic analysis correct with some interpretation; limited discussion of limitations.	Analysis incomplete or incorrect; little or no interpretation.
Writing & Organization	Well structured, concise, and clear. Figures/tables enhance understanding.	Mostly clear with minor organization issues; visuals adequate.	Disorganized or hard to follow; visuals missing or confusing.
Reflection & Iteration	Includes clear, specific statements of each member's contribution and personal takeaways. Strong reflection on successes, failures, and future improvements.	Some reflection; acknowledges a few weaknesses. Contribution statements present but too brief or uneven.	Little or no reflection or missing contribution statements.

Table 2: Grading Rubric (Excellent = 3, Satisfactory = 2, Needs Work = 1, Missing = 0)

Suggested Resources

Category	Resource / Link
Models and Repositories	Bayesian Network Repository – https://www.bnlearn.com/bnrepository/
Datasets	Google Dataset Search – https://datasetsearch.research.google.com/ UCI Machine Learning Repository – https://archive.ics.uci.edu/datasets Kaggle Datasets and Competitions – https://www.kaggle.com/datasets Genome Datasets – https://www.ncbi.nlm.nih.gov/datasets/genome/
Algorithm Implementation Help	Baum-Welch Algorithm (HMM Parameter Learning) – https://acme.byu.edu/00000186-a3db-d5de-afc7-f7df963e0001/hmm-pdf Viterbi Algorithm (Dynamic Programming for HMMs) – Assignment 6
Readymade Implementations	HMM Learning and Inference Library – https://hmmlearn.readthedocs.io/en/latest/ Bayesian Network Structure/CPT Learning (pgmpy) – https://pgmpy.org/detailed_notebooks/10.%20Learning%20Bayesian%20Networks%20from%20Data.html Pomegranate Library (Fast BN and HMM inference) – https://pomegranate.readthedocs.io/en/latest/ bnlearn Python Package (Structure learning) – https://pypi.org/project/bnlearn/

Table 3: Resources for Bayesian Network and HMM-based projects.