52935 - Probabilistic Methods in Artificial Intelligence **(Due: 27/06/24)**

Project 2 - Inference in HMM

Instructor: Prof. Gal Elidan *TA:* Ela Fallik *Name:* Hadar Tal

Part II

Programming Assignment - Inference in HMM

1. Exact inference using dynamic-programming (DP)

1. You are supplied with code for the forward and backward algorithms. Use them to implement the `log_posterior_Xt` method in the HMM class.

$$P(X_t = k \mid O_{1:t}) = \frac{P(X_t = k, O_{1:t})}{P(O_{1:t})} = \frac{P(X_t = k, O_{1:t}) \cdot P(O_{t+1:T} \mid X_t = k)}{P(O_{1:T})}$$

$$\begin{aligned}\log P(X_t = k \mid O_{1:t}) &= \log\left(\frac{P(X_t = k, O_{1:t}) \cdot P(O_{t+1:T} \mid X_t = k)}{P(O_{1:T})}\right) \\ &= \log P(X_t = k, O_{1:t}) + \log P(O_{t+1:T} \mid X_t = k) - \log P(O_{1:T}) \\ &= \mathcal{F}[t,k] + \mathcal{B}[t,k] - \text{log\_likelihood}(O_{1:T})\end{aligned}$$

Implemented the `log_posterior_Xt` method in the HMM class to calculate the log posterior $P(X_t = k \mid O_{1:t})$ for each t and k.

2. Exact posteriors: We load the observations from `obs_data.csv`, calculate the marginal posteriors for the $N = 20$ observations in the dataset $p(X_t = 1|o_{1:T}[i])$, and plot the prior distributions $p(X_t = 1)$ versus the mean marginal posteriors $\mu_t = \frac{1}{N}\sum_{i=1}^{N} p(X_t = 1|o_{1:T}[i])$ for each t.
What are the differences and why do they exist? What does it suggest about the distribution from which the observations were sampled?

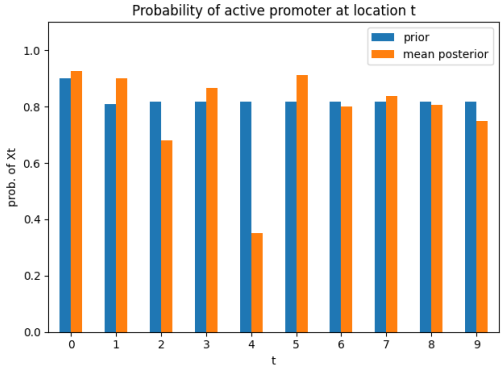

Figure 1: Prior distributions $p(X_t = 1)$ versus the mean marginal posteriors $\mu_t = \frac{1}{N}\sum_{i=1}^{N} p(X_t = 1|o_{1:T}[i])$ for each t.

The plot shows the prior distributions $p(X_t = 1)$ (blue bars) versus the mean marginal posteriors μ_t (orange bars) for each time step t. The differences between the prior and the mean marginal posteriors suggest that the observations have a significant impact on the estimated probabilities of active promoters. Specifically, the mean marginal posteriors differ from the prior distributions due to the evidence provided by the observations. This indicates that the dataset likely contains informative observations that influence

the posterior estimates.
The deviations from the prior suggest that the distribution from which the observations were sampled has regions with varying levels of promoter activity, as captured by the posterior probabilities.

3. We want to use the marginal posteriors $p(X_t|o_{1:T}[i])$ to assign each location t its status and predict areas of active promoter:

$$\hat{X}_t[i] = \arg\max_x p(X_t = x|o_{1:T}[i])$$

Implement a `naive_predict_by_posterior` method in the HMM class.

Implemented the `naive_predict_by_posterior` method in the HMM class.

4. Prediction: We use this method to predict hidden sequences for the given observations. We compare them to the ground-truth sequences from `hidden_data.csv`. What is the accuracy of these predictions (use the same definition as Project 1)? What is the difference compared to the prediction we used in Project 1? Is the rule in Eq.4 a good prediction rule?

Results:

- Exact Posterior Prediction Accuracy: 0.900
- Naive Prediction Accuracy: 0.895

Analysis:

- The exact posterior method is slightly more accurate than the naive method.
- The exact method uses the forward-backward algorithm, considering the entire observation sequence, while the naive method likely uses a simpler, less contextual approach.
- This accuracy difference shows the exact posterior is better at capturing dependencies in the data.
- Eq. 4's rule is a good prediction method, leveraging the full observation sequence for more informed predictions.

2. Sampling-based inference

1. Gibbs sampling: In this algorithm, we'll start with a starting point $X_{1:T}$ sampled from the prior $p(X_{1:T})$, and sample M sequences of hidden states sequentially:

$$X_t^{(m)} \sim p(X_t|X_{-t}^{(m-1)}, o_{1:T}[i]), \quad t = 1, \ldots, T$$

We then use the samples to estimate the posterior $p(X_t = x|o_{1:T}[i])$ for each t.

(a) For observations $o_{1:T}[i]$, at iteration m, we want to use the samples from the previous iteration $\{X_t^{(m-1)}\}_{t=1}^T$ to sample $X_t^{(m)}$ from the distribution $p(X_t|X_{-t}^{(m-1)}, o_{1:T}[i])$. How can we sample from this distribution using the CPDs of the network?

In a Hidden Markov Model (HMM), the Markov blanket of a hidden state X_t includes its immediate neighbors and the observation at time t. Specifically, the Markov blanket of X_t consists of X_{t-1}, O_t, and X_{t+1} (if they exist). This is because in an HMM, X_t is conditionally independent of all other variables given these three variables.
Given this, we can write the conditional distribution for sampling X_t as follows:

$$\begin{aligned}
&P(X_t = k \mid X_{-t}^{(m-1)}, O_{1:T}[i]) \\
&= P(X_t = k \mid X_{t-1}^{(m-1)}, O_t[i], X_{t+1}^{(m-1)}) \\
&= \frac{P_\phi(X_t = k, X_{t-1}^{(m-1)}, O_t[i], X_{t+1}^{(m-1)})}{\sum_{x_t \in \mathrm{Val}(X_t)} P_\phi(X_t = x_t, X_{t-1}^{(m-1)}, O_t[i], X_{t+1}^{(m-1)})} \\
&= \frac{P_\phi(X_{t-1}^{(m-1)}) \cdot P(X_t = k \mid X_{t-1}^{(m-1)}) \cdot P(O_t[i] \mid X_t = k) \cdot P(X_{t+1}^{(m-1)} \mid X_t = k)}{\sum_{x_t \in \mathrm{Val}(X_t)} P_\phi(X_{t-1}^{(m-1)}) \cdot P(X_t = x_t \mid X_{t-1}^{(m-1)}) \cdot P(O_t[i] \mid X_t = x_t) \cdot P(X_{t+1}^{(m-1)} \mid X_t = x_t)} \\
&= \frac{P(X_t = k \mid X_{t-1}^{(m-1)}) \cdot P(O_t[i] \mid X_t = k) \cdot P(X_{t+1}^{(m-1)} \mid X_t = k)}{\sum_{x_t \in \mathrm{Val}(X_t)} P(X_t = x_t \mid X_{t-1}^{(m-1)}) \cdot P(O_t[i] \mid X_t = x_t) \cdot P(X_{t+1}^{(m-1)} \mid X_t = x_t)}
\end{aligned}$$

(b) Given M samples $\{X_t^{(m)}\}_{m=1}^{M}$ from the process described above, how can we calculate the posterior $p(X_t = x|o_{1:T}[i])$, while discarding the first M_{start} samples to allow for a "burn-in" period?

To calculate the posterior $p(X_t = x|o_{1:T}[i])$ for each t, we can use the samples $\{X_t^{(m)}\}_{m=M_{\text{start}}}^{M}$ obtained from the Gibbs sampling process. We can estimate the posterior by counting the number of times $X_t = x$ occurs in the samples and normalizing by the total number of samples. The posterior can be calculated as follows:

$$p(X_t = x|o_{1:T}[i]) = \frac{1}{M - M_{\text{start}}} \sum_{m=M_{\text{start}}}^{M} \mathbb{I}(X_t^{(m)} = x)$$

where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the condition is true and 0 otherwise.

(c) Implement a method `gibbs_sampling_posterior` in the HMM class to calculate the posterior $p(X_t = x|o_{1:T}[i])$ for each t with $M - M_{\text{start}}$ samples.

Implemented the `gibbs_sampling_posterior` method in the HMM class.

(d) We run the Gibbs algorithm with $M \in [10, 50, 70, 100, 200, 300, 500]$ samples to calculate the marginal posteriors of each observation in the dataset. We run the algorithm 5 times for each value of M. We plot the posterior of X_5 for the first 10 observations $p(X_5|o_{1:T}[1:10])$ versus M with confidence intervals $(\mu \pm \sigma)$, and add to the plot the exact posterior calculated using DP. Does the Gibbs estimation converge? Does it converge to the correct posterior? Explain your results. What can you say about the convergence of this algorithm?

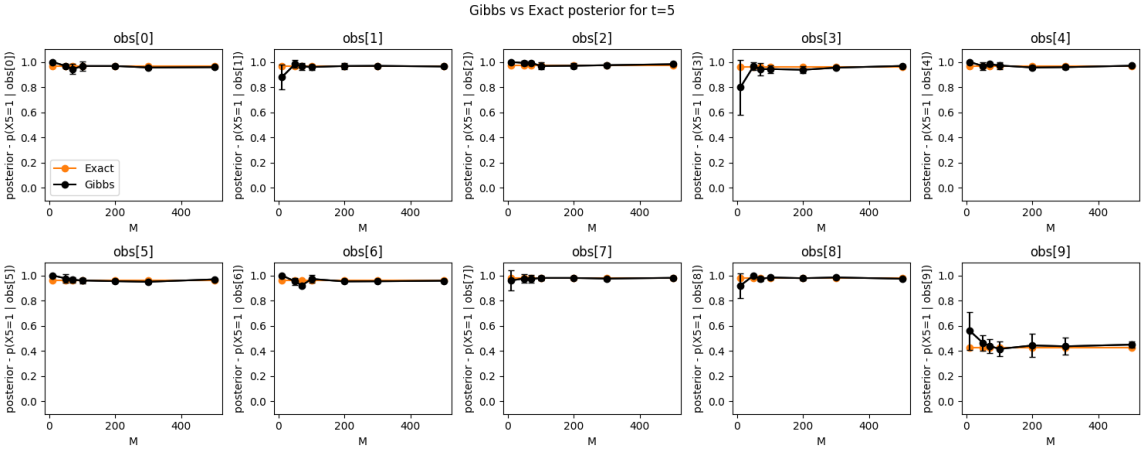

Figure 2: Posterior of X_5 for the first 10 observations $p(X_5|o_{1:T}[1:10])$ versus M with confidence intervals $(\mu \pm \sigma)$.

The plot shows the comparison between the Gibbs sampling estimation and the exact posterior calculated using dynamic programming (DP) for X_5 across the first 10 observations.

Analysis:

- **Convergence:** The Gibbs estimation converges fairly fast as M increases. Even with a relatively small number of samples (e.g., $M = 50$), the estimates are quite stable and close to the exact posterior.
- **Correctness:** The Gibbs estimates converge to values close to the exact posterior calculated using DP, indicating that the Gibbs sampling method is correctly estimating the posterior distributions.
- **Results:** The convergence of the Gibbs sampling algorithm demonstrates its effectiveness in approximating the true posterior distribution, especially with a higher number of samples M. The confidence intervals also decrease with increasing M, showing more precise estimates.
- **Implications:** The fast convergence of the Gibbs sampling method makes it a practical and efficient approach for posterior estimation in HMMs, providing reliable results with relatively few samples.

2. We use the estimated posteriors of each algorithm and the `naive_predict_by_posterior` method to assign each location its status (same as for the exact posterior). We plot the accuracy of the algorithms as a function of M, and plot on the same graph the accuracy of the exact posterior (as a constant function of M). What is the trend?

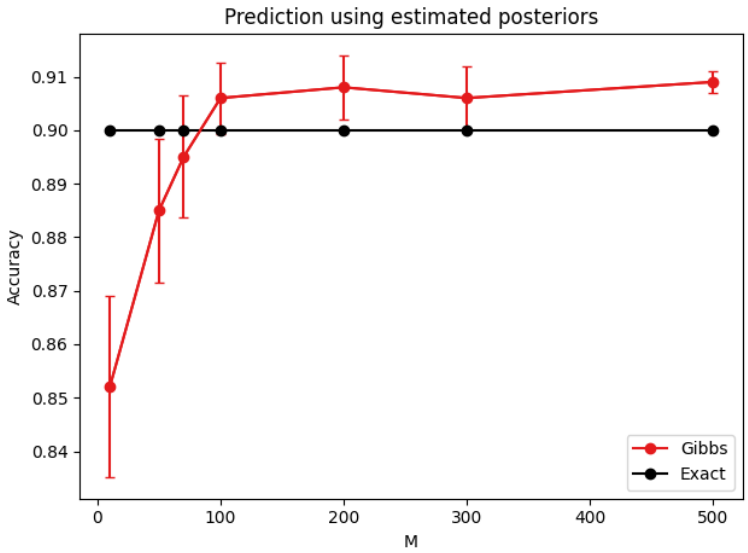

Figure 3: Accuracy of the algorithms as a function of M, with the accuracy of the exact posterior as a constant function of M.

The plot shows the accuracy of the Gibbs sampling algorithm compared to the exact posterior as a function of the number of samples M.

Analysis:

- **Trend:** The accuracy of the Gibbs sampling algorithm increases rapidly with the number of samples M and stabilizes around $M = 100$. The exact posterior accuracy remains constant.
- **Comparison:** The Gibbs sampling method eventually reaches and slightly surpasses the accuracy of the exact posterior, indicating that it is an effective method for estimating the posterior, especially with a larger number of samples.
- **Implications:** The Gibbs sampling method can achieve high accuracy with sufficient samples, making it a reliable alternative to exact inference methods. The slight fluctuation in accuracy with smaller M values suggests that more samples are needed for stabilization.