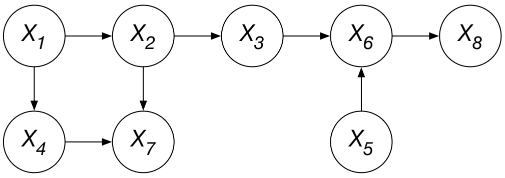
CS5340: Uncertainty Modeling in AI

Tutorial 8

Released: Mar. 20, 2023

1 Gibbs Sampling

You want to run Gibbs sampling on the following graphical model. For each of the random variables below, what is the correct conditional to sample from? **Note:** If there are multiple correct answers, select the one that conditions upon the fewest number of random variables.



Problem 1. Sample x_1.

A. $p(X_1)$ (sample from the prior)

B. $p(X_1|X_2, X_4, X_7)$

C. $p(X_1|X_2, X_4)$

D. $p(X_1|X_2, X_3, X_4, X_7)$

E. $p(X_1|X_2, X_3, X_4, X_7, X_5)$

Problem 2. Sample x_2.

A. $p(X_2|X_1, X_3)$

B. $p(X_2|X_4, X_7)$

C. $p(X_2|X_1, X_3, X_4, X_7)$

D. $p(X_2|X_3, X_7)$

E. $p(X_2|X_1)$

Problem 3. Sample x_3.

A. $p(X_3|X_2, X_6, X_7)$

B. $p(X_3|X_2, X_6)$

C. $p(X_3|X_2)$

D. $p(X_3|X_2, X_5, X_6)$

E. $p(X_3|X_2, X_5, X_6, X_8)$

Problem 4. Sample x_4.

A. $p(X_4|X_1, X_2, X_7)$

B. $p(X_4|X_2, X_7)$

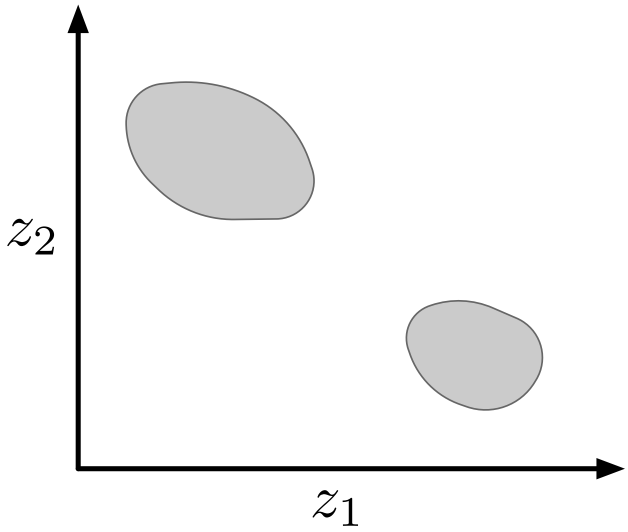
C. $p(X_4|X_1)$

D. $p(X_4|X_1, X_2, X_3, X_7)$

E. $p(X_4|X_1, X_2, X_3, X_7, X_6, X_8)$

2 Discussion Questions

Problem 5. Consider the following distribution of two variables z_1 and z_2 that is uniform over the shaded regions and that is zero everywhere else. **Does the standard Gibbs sampling procedure sample correctly from this distribution**?



Problem 6. Consider the Metropolis-Hastings algorithm. At each step i, we sample a new point from the proposal distribution $q(\mathbf{x}|\mathbf{x}^{(i)})$. In the lectures, we used as an example the simple *isotropic* (spherical) Gaussian centered upon $\mathbf{x}^{(i)}$, i.e.,

$$q(\mathbf{x}|\mathbf{x}^{(i)}) = \mathcal{N}(\mathbf{x}^{(i)}, \sigma^2\mathbf{I})$$

which is a common choice for continuous variables. The variance σ^2 is a parameter of the proposal distribution. **How sensitive is MH sampling to the parameter σ^2.** What are the respective trade-offs when considering how to set σ^2? *Hint:* Consider a elongated bi-variate Gaussian having strong correlations between its components.

3 The Right Transitions

For each of the matrices below, select True if the matrix is valid *transition matrix* over the states for use in a MCMC algorithm. Select False otherwise. Justify your answer. *Hint:* Look up what properties are required for a transition matrix in an MCMC method.

Problem 7.

$$T = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.8 \\ 0.3 & 0.2 & 0.1 \end{bmatrix}$$

Problem 8.

$$T = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.3 & 0.0 & 0.7 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

Problem 9.

$$T = \begin{bmatrix} 0.9 & 0.1 & 0.0 \\ 0.1 & 0.9 & 0.0 \\ 0.5 & 0.3 & 0.2 \end{bmatrix}$$

Problem 10.

$$T = \begin{bmatrix} 0.7 & 0.1 & 0.2 \\ 0.2 & 0.3 & 0.5 \\ 0.0 & 0.6 & 0.4 \end{bmatrix}$$

Problem 11.

$$T = \begin{bmatrix} 0.0 & 0.8 & 0.2 \\ 0.2 & 0.3 & 0.5 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}$$

Problem 12.

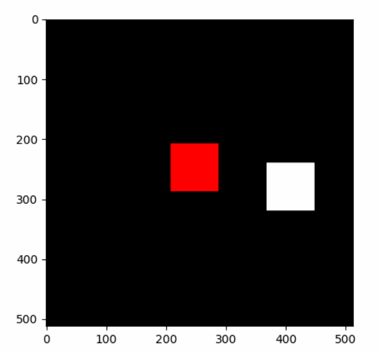
$$T = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$$

Sequential VAE

In Tutorial 7, we studied a state-space model where the transitions and emissions were linear functions, along with Gaussian random variables. This resulted in a linear Gaussian model, which made inference and learning tractable.

In this tutorial, we will extend the linear Gaussian model to the nonlinear regime: we will look at a (relatively) state-of-the-art Sequential VAE model. We will combine deep learning (neural networks) with the state-space model to yield a more expressive sequential model capable of learning from complex high-dimensional image observations.

Problem and Environment Suppose we have a robot (represented by the white rectangle) that moves in a 2D room. The task is to control the robot to reach the goal position (represented by the red rectangle) at the center of the room. We can directly control the velocity of the robot along x and y axes. However, we cannot directly observe the ground truth coordinates of the robot nor the goal. We only have access to high dimensional pixel (image) observations, as shown in the image below:



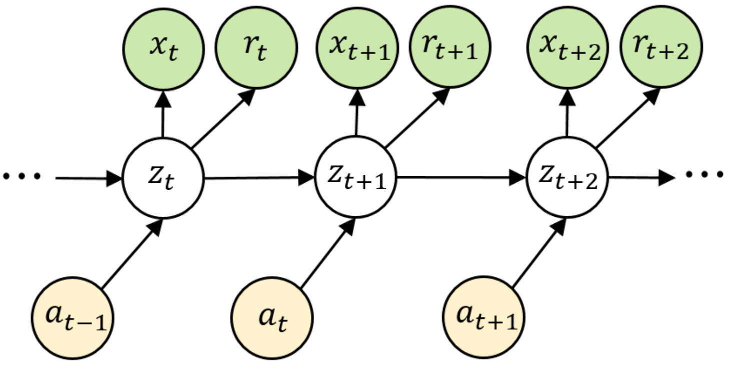
Note that the robot doesn't "know" it is the while square or that the goal is the red square. We have to learn these associations using data.

Overview In this tutorial, we will work through three main steps. First, we will construct a probabilistic model (directed graphical model) for this problem. Then we will use approximate inference to learn the parameters of this model. Finally, we will use this model to control the robot!

Learning the Model

Let us first define our State Space Model (SSM). Intuitively, we will model an agent that is sequentially taking actions in a world and receiving rewards and visual observations. The visual/image observation x_t at time t are generated from the latent state z_t. The model assumes Markovian transitions where the next state is conditioned upon the current state and the action a_t taken by the agent. Upon taking an action, the agent receives reward r_t. The goal of the agent is to maximize its rewards.

Problem 13. The problem above can be formulated as a Bayesian network:



Each of the factorized distributions are modelled using nonlinear functions:

- Transitions: $p_\theta(z_t|z_{t-1}, a_{t-1}) = p(z_t|f_\theta(z_{t-1}, a_{t-1}))$
- Observations: $p_\theta(x_t|z_t) = p(x_t|d_\theta(z_t))$
- Rewards: $p_\theta(r_t|z_t) = p(r_t|r_\theta(z_t))$

where f_θ, d_θ^m, r_θ are neural networks parameterized by θ. First, consider that the actions are always observed. Write out the factorization of the probability $p_\theta(x_{1:T}, r_{1:T}, z_{1:T}|a_{1:T-1})$ corresponding to the DGM above.

Problem 14. Learning this model is intractable due to the nonlinear transition, observation, and reward functions. We will perform variational inference to learn the parameters of the model. Assume we observe trajectories τ sampled from data distribution $p_d(\tau)$. Each trajectory is an observation $\tau = \{(x_t, r_t, a_t)\}_{t=1}^T$.

To obtain the maximum likelihood estimate (MLE) of the parameters θ, which of the following functions should we optimize?

A. $\mathbb{E}_{p_d}[\log p(x_{1:T}, r_{1:T}|a_{1:T-1}; \theta)]$

B. $\mathbb{E}_{p_d}[\log p(x_{1:T}, r_{1:T}, z_{1:T}|a_{1:T-1}; \theta)]$

C. $\mathbb{E}_{p_d}[\log p(\theta|x_{1:T}, r_{1:T}, z_{1:T}, a_{1:T-1})]$

D. $\mathbb{E}_{p_d}[\log p(\theta, x_{1:T}|r_{1:T}, z_{1:T}, a_{1:T-1})]$

E. Any of the above would work.

Solve this problem before moving to the next one.

Problem 15. Note that the maximum likelihood estimation requires us to marginalize out the latent variables $z^i_{1:T}$ for each trajectory τ^i in a dataset $\mathcal{D}$. We will need the variational posterior q. Consider three choices:

A. $q(z^i_{1:T}) = \prod_{t=1} q(z^i_t)$ where the q's are Gaussian distribution that share the same parameters (mean and covariance).

B. $q(z^i_{1:T}) = \prod_{t=1} q^i_t(z^i_t)$ where each q^i_t is a Gaussian distribution with *different* parameters.

C. $q(z^i_{1:T}|x^i_{1:T}, a^i_{1:T-1}) = \prod_{t=1}^T q_\phi(z^i_t|g_\phi(x^i_{1:t}, a^i_{1:t-1}))$ where q_ϕ is a Gaussian distribution and the *inference network* $g_\phi(x_{1:t}, a_{1:t-1})$ is a neural network (usually a recurrent neural network like a LSTM or GRU) parameterized by ϕ that outputs the mean and covariance for each z^i_t. The inference networks provides the parameters for the mean and the covariance of the distributions.

Between A, B and C, which variational distribution is the least expressive? Which is the most expressive?

Problem 16. Consider the variational distribution given in C above, i.e.,

$$q(z^i_{1:T}|x^i_{1:T}, a^i_{1:T-1}) = \prod_{t=1}^{T} q_\phi(z^i_t|g_\phi(x^i_{1:t}, a^i_{1:t-1}))$$

where each q_ϕ is a Gaussian distribution and the *inference network* $g_\phi(x_t, z_{t-1}, a_{t-1})$ is a neural network parameterized by ϕ. The inference network provides the parameters for the mean and the covariance of the distributions. Is $q(z^i_{1:T}|x^i_{1:T}, a^i_{1:T-1})$ a multivariate Gaussian in general? Provide a brief justification.

Problem 17. Suppose we pick the inference network variational distribution given in C above. To simplify notation, we call this $q_\phi(z_t)$. Given all these distributions and trajectories $\tau \sim p_d(\tau)$, we seek to learn the parameters θ and ϕ. We optimize the evidence lower bound (ELBO) under the data distribution p_d using a variational distribution q_ϕ over the latent state variables z_t.

$$\mathbb{E}_{p_d}[\text{ELBO}] \le \mathbb{E}_{p_d}[\log p_\theta(x_{1:T}, r_{1:T}|a_{1:T-1})] \tag{1}$$

where

$$\text{ELBO} = \sum_{t=1}^{T} \left(\mathop{\mathbb{E}}_{q_\phi(z_t)} [\log p_\theta(x_t|z_t)] + \mathop{\mathbb{E}}_{q_\phi(z_t)} [\log p_\theta(r_t|z_t)] \right) \tag{2}$$

$$- \sum_{t=2}^{T} \mathop{\mathbb{E}}_{q_\phi(z_{t-1})} [\text{KL}\left[q_\phi(z_t) \| p_\theta(z_t|z_{t-1}, a_{t-1})\right]] - \text{KL}\left[q_\phi(z_1) \| p_\theta(z_1)\right] \tag{3}$$

Note that we have dropped the explicit conditioning to reduce clutter in the above equation, i.e., $q_\phi(z_t) = q_\phi(z_t|x_{1:t}, a_{1:t-1})$. Derive the ELBO shown above.

Planning and Control

With the ELBO, we can learn the parameters θ (and ϕ) using an off-the-shelf-optimizer like stochastic gradient descent (SGD). Once we have learnt the model, we can use it for planning/control. The idea is quite simple. Say we are at current time step t, we will use the model to simulate possible futures (up to some horizon $t + H$) and find actions that lead to the best return (the sum of discounted rewards).

$$\text{argmax}_{a_{t:t+H-1}} J = \mathrm{E}_{p(z_{t+1:t+H}|a_{t:t+H-1}, z_t)} \left[\sum_{k=1}^{H} \gamma^k r(z_{t+k}) \right] \tag{4}$$

We then take action a_t and then repeat the process. We'll use a simple zero-order method called the cross-entropy method (CEM), which we will demonstrate during tutorial (but will not go into detail here since it is beyond the scope of the course[1]).

[1] For those who are interested, check out: `https://jetnew.io/blog/2021/cem/`