Bayesian Inversion and Markov Kernels

Kenneth Zhang[1]

[1]OpenAI Inc, San Francisco, California

July 2023

1 Introduction

Welcome to the user manual for the Compositional Bayesian Inversion of Markov Kernels! This guide aims to provide a comprehensive overview and step-by-step usage instructions for the provided PyTorch implementation.

1.1 Markov Kernels and Bayesian Inversion

In various scientific and engineering applications, we often encounter systems that exhibit probabilistic transitions between different states over time. Such systems can be elegantly modeled using Markov kernels. A Markov kernel is a mathematical construct that characterizes the probabilistic evolution of states in a Markov process.

Intuitively, a Markov kernel can be thought of as a "transition rule" that dictates how the probabilities of being in different states change from one time step to another. The core idea is that the future state probabilities depend solely on the current state probabilities and not on the history of states.

On the other hand, Bayesian inversion is a powerful technique used to infer the underlying structure of a system from observed data. In the context of Markov kernels, Bayesian inversion aims to learn the inverse transformation that maps the probabilities of states at a given time step back to the probabilities at the previous time step.

1.2 Compositional Bayesian Inversion

The Compositional Bayesian Inversion of Markov Kernels combines the strengths of Markov kernels and Bayesian inversion in a novel and efficient manner. The approach involves constructing a Bayesian inverter, which learns to approximate the inverse transformation of a given Markov kernel.

In this implementation, we represent the Markov kernel using a transition matrix, where each entry indicates the probability of transitioning from one state to another. The Bayesian inverter is modeled as a neural network, which is trained to learn the inverse transformation of the Markov kernel.

The key insight is that by composing the Markov kernel with the Bayesian inverter, we can obtain an approximate inverse transformation, allowing us to infer the initial state probabilities given the probabilities at a subsequent time step in a transition timeline.

2 Key Components

1. `MarkovKernel`: Represents a Markov kernel with a given transition matrix, which models the probabilistic transition between states in a Markov process using a transition matrix and input data.

2. `BayesianInverter`: A neural network-based Bayesian inverter, which learns to approximate the inverse transformation of a given Markov kernel.

Additionally, the code provides a composition function, `compose`, which combines the Markov kernel and the Bayesian inverter to create a new composed transformation from the transition matrix and inputted data.

3 Usage

1. **Define the Transition Matrix:** To begin, you need to define the transition matrix for your Markov kernel. This matrix should represent the probabilities of transitioning between states in your Markov process.

2. **Create Instances of MarkovKernel and BayesianInverter:** Instantiate the `MarkovKernel` class with the defined transition matrix and the `BayesianInverter` class with the appropriate input and output dimensions.

3. **Apply the Composed Kernel:** Use the `compose` function to combine the Markov kernel and the Bayesian inverter. Apply the resulting composed kernel to your input data to approximate the inverse transformation.

4 Mathematical Intuition

4.1 Bayesian Inversion

The goal of Bayesian inversion is to infer the underlying structure of a system from observed data. In the context of Markov kernels, Bayesian inversion aims to learn the inverse transformation that maps the probabilities of states at a given time step back to the probabilities at the previous time step.

Let us denote:

- X_t as the random variable representing the state at time step t.
- X_{t+1} as the random variable representing the state at the next time step $t+1$.
- $P(X_{t+1}|X_t)$ as the conditional probability of transitioning from state X_t to state X_{t+1} in the Markov kernel.

The Bayesian inversion problem can be formulated as finding the posterior probability $P(X_t|X_{t+1})$, which represents the probability of being in state X_t given the observation of state X_{t+1}. Applying Bayes' theorem, we have:

$$P(X_t|X_{t+1}) = \frac{P(X_{t+1}|X_t) \cdot P(X_t)}{P(X_{t+1})}$$

Where:

- $P(X_t)$ is the prior probability of being in state X_t at time step t.
- $P(X_{t+1})$ is the evidence, representing the probability of observing state X_{t+1}.

The key challenge in Bayesian inversion is to learn the conditional probability $P(X_{t+1}|X_t)$ from the observed data, as it characterizes the underlying dynamics of the system.

4.2 Markov Kernels

A Markov kernel is a mathematical construct used to model the probabilistic transition between states in a Markov process. It can be represented using a transition matrix.

Let X_t be the random variable representing the state at time step t with possible values x_i for $i = 1, 2, \ldots, n$. Then, the Markov kernel is defined as a matrix K where K_{ij} represents the probability of transitioning from state x_i to state x_j in one time step.

Mathematically, the Markov kernel satisfies the following properties:

1. Non-negativity: $K_{ij} \geq 0$ for all i, j.
2. Row Sum: $\sum_{j=1}^{n} K_{ij} = 1$ for all i.

The row sum property ensures that the probabilities for each state x_i sum up to 1, representing a valid probability distribution.

Given a probability distribution $P(X_t)$ over the states at time step t, the transition to the next time step $t+1$ is obtained by matrix multiplication:

$$P(X_{t+1}) = P(X_t) \cdot K$$

Where $P(X_{t+1})$ is the probability distribution over the states at time step $t+1$, and K is the Markov kernel.

This matrix multiplication represents the probabilistic evolution of states in a Markov process, governed by the Markov kernel K.