Sequential Monte Carlo methods

Lecture 13 – Gibbs sampling

Fredrik Lindsten, Uppsala University

2017-08-28

UPPSALA UNIVERSITET

Outline – Lecture 13

Aim: Show an alternative MCMC procedure (Gibbs sampling) and how it conceptually can be used for learning of dynamical systems

Outline:

1. The Gibbs sampler
2. Gibbs sampling for linear dynamical systems
3. Gibbs sampler for nonlinear dynamical systems

The Gibbs sampler

Challenge with MCMC

Designing **efficient** Metropolis–Hastings kernels for **arbitrary and high-dimensional** target distributions can be very challenging.

Gibbs sampling turns the overall sampling problem into a **series of sub-problems**, each of which is hopefully easier to address.

A first Gibbs sampler

Let $\pi(x_1, x_2)$ be a target distribution over two (groups of) variables.

Basic factorization: $\pi(x_1, x_2) = \pi(x_2 \mid x_1)\pi(x_1)$

Thus:

- If $(X_1, X_2) \sim \pi(x_1, x_2)$, then X_1 is distributed according to $\pi(x_1)$.
- If $X_2^\star \mid (X_1 = x_1) \sim \pi(x_2 \mid x_1)$, then $(X_1, X_2^\star)$ is distributed according to $\pi(x_1, x_2)$.

Starting with a sample from the joint distribution, we can replace any of the variables by a draw from it's full conditional and still have a sample from the joint distribution.

ex) Gibbs sampler illustration

ex) Sample from,

$$\pi(x_1, x_2) = \mathcal{N}\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mid \begin{pmatrix} 10 \\ 10 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}\right).$$

Gibbs sampler:

Initialize $x_1[1] = 0$, $x_2[1] = 0$
for $m = 2, \ldots, M$

Draw $x_1[m] \sim \pi(x_1 \mid x_2[m-1])$;

Draw $x_2[m] \sim \pi(x_2 \mid x_1[m])$.

ex) Gibbs sampler illustration

MCMC kernels

An MCMC sampler generates the Markov chain $\{x[m]\}_{m=1}^M$ by:

- **Initialize:** set $x[1]$ arbitrarily.
- **For** $m = 2$ **to** M**:** sample $x[m] \sim \kappa(x[m-1], x^\star)$.

$\kappa(x, x^\star)$ is a **Markov kernel** on $\mathcal{X}$, i.e. a conditional distribution for the next state $x^\star$ given the current state x.

Basic requirement 1: Stationarity of $\pi(x)$,

$$\int \pi(x)\kappa(x, x^\star)\mathrm{d}x = \pi(x^\star).$$

Basic requirement 2: Ergodicity — κ must allow the state to move in order to explore the state space.

The Gibbs Markov kernel

Target: $\pi(x) = \pi(x_1, \ldots, x_d)$

Input a configuration $x = (x_1, \ldots, x_d)$
for $j = 1, \ldots, d$
Sample $x_j^\star \sim \pi(x_j \mid x_1^\star, \ldots, x_{j-1}^\star, x_{j+1}, \ldots, x_d)$
Output $x^\star = (x_1^\star, \ldots, x_d^\star)$.

Gibbs kernel: This procedure defines a Markov kernel $\kappa(x, x^\star)$ with stationary distribution $\pi(x)$.

Extensions

There are many possible extensions of the basic Gibbs procedure, which also result in valid MCMC kernels.

- **Random scan:** select components to sample randomly (with or without replacement)
- **Overlapping blocks:** the groups of variables need not be disjoint
- **Collapsing:** analytical marginalization of some of the variables **(!)**

Extensions – Composition of MCMC methods

In many cases, exact sampling from some of the full conditional distributions is not possible.

Sufficient to sample from some Markov kernel which has the **full conditional distribution** as stationary distribution — we can make use of a combination of MCMC techniques.

If exact sampling from $\pi(x_j \mid x_{-j})$ is not possible:

$$X_j^\star \sim \kappa_j(x, x_j^\star) \text{ where } \int \kappa_j(x, x_j^\star)\pi(x_j \mid x_{-j})\mathrm{d}x_j = \pi(x_j^\star \mid x_{-j})$$

For instance, κ_j can be a Metropolis–Hastings kernel on the lower dimensional space $\mathcal{X}_j \ni x_j$.

(Short hand notation $x_{-j} = (x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_d)$.)

ex) Metropolis-within-Gibbs

Target:

$$\pi(x_1, x_2) \propto \widetilde{\pi}(x_1, x_2) = \underbrace{\exp\left(-\tfrac{1}{2}(2x_1 + \sin(6.28x_1))^2\right)}_{\widetilde{\pi}(x_1)} \underbrace{\mathcal{N}(x_2 \mid x_1^3, 0.1)}_{\pi(x_2 \mid x_1)}$$

Gibbs sampler:

Set $x_1[1] = 0$, $x_2[1] = 0$
for $m = 2, \ldots, M$
Draw $x_1[m] \sim \kappa_1(x[m-1], x_1^\star)$;
Draw $x_2[m] \sim \pi(x_2 \mid x_1[m])$.
where κ_1 is a Metropolis–Hastings kernel for $\pi(x_1 \mid x_2)$.

Note that $\pi(x_1 \mid x_2) = \frac{\pi(x_1, x_2)}{\pi(x_2)}$. Hence, **conditionally on** x_2,

$$\pi(x_1 \mid x_2) \propto \pi(x_1, x_2) \propto \widetilde{\pi}(x_1, x_2).$$

ex) Metropolis-within-Gibbs

Algorithm 1 Metropolis-within-Gibbs sampler for toy problem

1. **Initialize:** Set $x_1[1] = 0$, $x_2[1] = 0$.
2. **For** $m = 2$ **to** M**, iterate:**
   a. Sample $x_1' \sim \mathcal{N}(x_1 \mid x_1[m-1], 0.5^2)$.
   b. Sample $u \sim \mathcal{U}[0, 1]$.
   c. Compute the acceptance probability
   $$\alpha = \min\left(1, \frac{\widetilde{\pi}(x_1', x_2[m-1])}{\widetilde{\pi}(x_1[m-1], x_2[m-1]}\right)$$
   d. Set
   $$x_1[m] = \begin{cases} x_1' & \text{if } u \leq \alpha \\ x_1[m-1] & \text{otherwise} \end{cases}$$
   e. Draw $x_2[m] \sim \pi(x_2 \mid x_1[m])$.

Gibbs sampling for dynamical systems

ex) Gibbs sampling for linear Gaussian system

Simple LG-SSM,

$$\begin{aligned} X_t &= 0.9X_{t-1} + V_t, & V_t &\sim \mathcal{N}(0, \Theta_1), \\ Y_t &= X_t + E_t, & E_t &\sim \mathcal{N}(0, \Theta_2), \end{aligned}$$

With inverse-Gamma priors: $\Theta_1 \sim \mathcal{IG}(0.1, 0.1)$, $\Theta_2 \sim \mathcal{IG}(0.1, 0.1)$.

Task: Compute $p(\theta \mid y_{1:T})$ for a batch of $T = 100$ observations.

The inverse-Gamma distribution is **conjugate prior** for an unknown variance of a Gaussian likelihood $\Rightarrow$

$$p(\theta_1 \mid x_{0:T}, y_{1:T}) = \mathcal{IG}\left(\theta_1 \mid 0.1 + \tfrac{T}{2}, 0.1 + \tfrac{1}{2}\textstyle\sum_{t=1}^{T}(x_t - 0.9x_{t-1})^2\right),$$
$$p(\theta_2 \mid x_{0:T}, y_{1:T}) = \mathcal{IG}\left(\theta_2 \mid 0.1 + \tfrac{T}{2}, 0.1 + \tfrac{1}{2}\textstyle\sum_{t=1}^{T}(y_t - x_t)^2\right).$$

ex) Gibbs sampling for linear Gaussian system

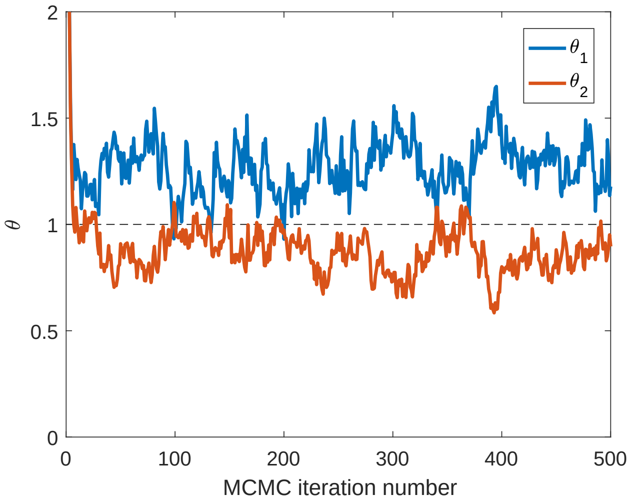
Gibbs sampler:

Initialize $\theta_1[1] = \theta_2[1] = 5$ (arbitrary!)
for $m = 2, \ldots, M$

- Draw $x_{0:T}[m] \sim p(x_{0:T} \mid \theta[m-1], y_{1:T})$,
  by using Kalman smoothing techniques.
- Draw $\theta[m] \sim p(\theta \mid x_{0:T}[m], y_{1:T})$,
  i.e., simulate $\theta_1[m]$ and $\theta_2[m]$ from their inverse-Gamma posteriors.

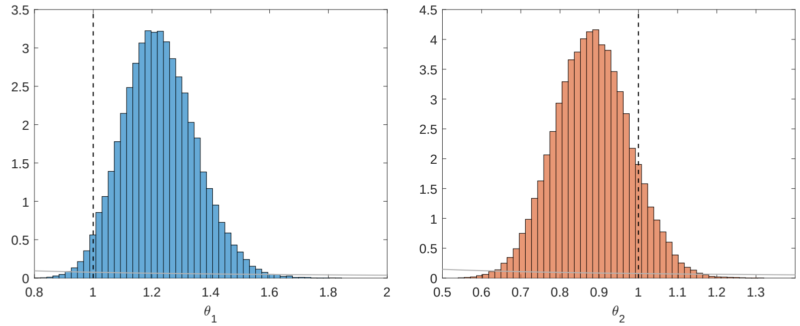
ex) Gibbs sampling for linear Gaussian system

First 500 iterations of the Gibbs sampler for θ_1 and θ_2.



ex) Gibbs sampling for linear Gaussian system

Marginal posterior distributions, $p(\theta_1 \mid y_{1:T})$ and $p(\theta_2 \mid y_{1:T})$, based on 50 000 iterations of the Gibbs sampler.



Gibbs sampling for nonlinear dynamical systems

What about a general nonlinear/non-Gaussian dynamical system?

$$X_t \mid (X_{t-1} = x_{t-1}, \Theta = \theta) \sim p(x_t \mid x_{t-1}, \theta),$$
$$Y_t \mid (X_t = x_t, \Theta = \theta) \sim p(y_t \mid x_t, \theta),$$
$$X_0 \sim p(x_0), \qquad \Theta \sim p(\theta).$$

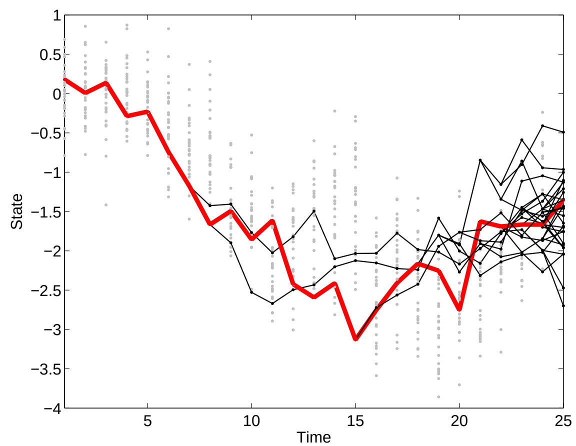
Gibbs sampler:

- Draw $\theta^\star \sim p(\theta \mid x_{0:T}, y_{1:T})$, **OK!**
- Draw $x_{0:T}^\star \sim p(x_{0:T} \mid \theta^\star, y_{1:T})$. **Hard!**

Problem: $p(x_{0:T} \mid \theta, y_{1:T})$ not available!

Idea: Approximate $p(x_{0:T} \mid \theta, y_{1:T})$ using a particle filter?

Sampling based on the PF



With $\mathbb{P}(X_{0:T}^\star = x_{0:T}^i) = w_T^i$ we get $X_{0:T}^\star \overset{\text{approx.}}{\sim} p(x_{0:T} \mid \theta, y_{1:T})$.

Sampling based on the PF

Problems with this approach:

- Based on a PF $\Rightarrow$ approximate sample.
- $p(\theta, x_{1:T} \mid y_{1:T})$ is not a stationary distribution.
- Relies on large N to be successful.
- A lot of wasted computations.

The PMCMC framework allows us to address these issues!

A few concepts to summarize lecture 13

Gibbs sampler: an MCMC sampler that iteratively simulates the unknown variables of the model from their conditional distributions.

MCMC within Gibbs: If exact sampling from some conditional is not possible, we may use any valid MCMC kernel within a Gibbs sampler to simulate from this conditional.

Gibbs sampling for dynamical systems: boils down to sampling the model parameters **with fixes states** + sampling the states with **fixed parameters** (state inference).