CSC 412/2506:
Probabilistic Learning and Reasoning

Michal Malyska

University of Toronto

Overview

We continue with Markov Chain Monte Carlo

- Gibbs Sampling
- Hamiltonian Monte Carlo
- Diagnostics

Gibbs Sampling Procedure

Suppose the vector x has been divided into d components

$$x = (x_1, ..., x_d).$$

Start with any $x^{(0)} = (x_1^{(0)}, \ldots, x_d^{(0)})$. In the t-th iteration:

- For $j = 1, \ldots, d$:
  - Sample $x_j^{(t)}$ from the conditional distribution given other components:

$$x_j^{(t)} \sim p(x_j | x_{-j}^{(t-1)})$$

    Where $x_{-j}^{(t-1)}$ represents all the components of x except for x_j at their current values:

$$x_{-j}^{(t-1)} = (x_1^{(t)}, x_2^{(t)}, ..., x_{j-1}^{(t)}, x_{j+1}^{(t-1)}, ..., x_d^{(t-1)})$$

- No accept/reject, only accept.

Example: Bivariate Gaussian

Consider a (simple) problem of sampling from the bivariate Gaussian

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \sim N_2(\mu, \Sigma), \qquad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \ \Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}.$$

We have

$$X_1|X_2 = x_2 \sim N(\mu_1 + \rho(x_2 - \mu_2), 1 - \rho^2)$$

$$X_2|X_1 = x_1 \sim N(\mu_2 + \rho(x_1 - \mu_1), 1 - \rho^2)$$

Given $X^{(0)}$ we proceed iteratively for $t \geq 1$:

$$X_1^{(t)} \sim N(\mu_1 + \rho(x_2^{(t-1)} - \mu_2), 1 - \rho^2)$$

$$X_2^{(t)} \sim N(\mu_2 + \rho(x_1^{(t)} - \mu_1), 1 - \rho^2)$$

Example: Bivariate Gaussian

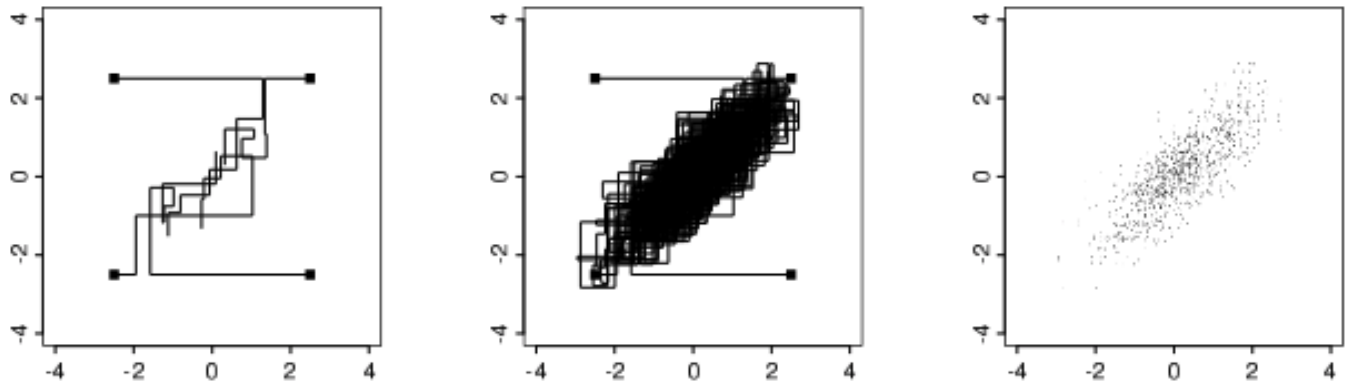


Figure 11.2 *Four independent sequences of the Gibbs sampler for a bivariate normal distribution with correlation $\rho = 0.8$, with overdispersed starting points indicated by solid squares. (a) First 10 iterations, showing the componentwise updating of the Gibbs iterations. (b) After 500 iterations, the sequences have reached approximate convergence. Figure (c) shows the points from the second halves of the sequences, representing a set of correlated draws from the target distribution.*

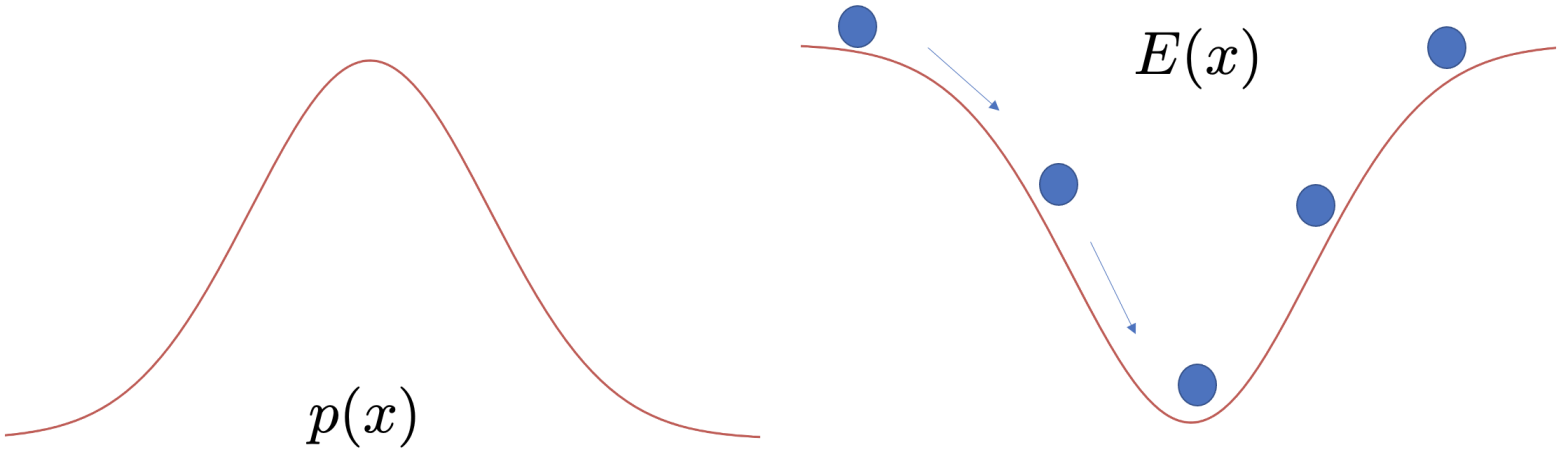
1

(The real power of Gibbs approach comes in situations when the distribution is hard but full-conditionals are simple, e.g. Ising)

[1] From "Bayesian Data Analysis Third edition" by Gelman, Carlin, Stern, Dunson, Vehtari, Rubin

Hamiltonian Monte Carlo

- This is essentially a Metropolis-Hastings algorithm with a specialized proposal mechanism.
- Algorithm uses a physical analogy to make proposals.
- Given the position x, the potential energy is $E(x)$



- Construct a distribution

$$p(x) \propto e^{-E(x)}, \quad \text{with} \quad E(x) = -\log(\tilde{p}(x))$$

where $\tilde{p}(x)$ is the unnormalized density we can evaluate.

Hamiltonian Monte Carlo

- Introduce **momentum** v carrying the kinetic energy

$$K(v) \;=\; \tfrac{1}{2}\|v\|^2 \;=\; \tfrac{1}{2}v^\top v.$$

- Total energy or **Hamiltonian**:

$$H(x, v) = E(x) + K(v).$$

- Energy is preserved:
  - Frictionless ball rolling $(x, v) \to (x', v')$
  - $H(x, v) = H(x', v')$.
- Ideal Hamiltonian dynamics are reversible: reverse v and the ball will return to its start point! $(x', -v') \to (x, -v)$

Hamiltonian Monte Carlo

- The joint distribution:
  - $p(x, v) \propto e^{-E(x)} e^{-K(v)} = e^{-E(x)-K(v)} = e^{-H(x,v)}$
  - Momentum is Gaussian, and independent of the position.
- MCMC procedure
  - Sample the momentum from a Gaussian.
  - Simulate Hamiltonian dynamics, flip sign of the momentum
    - Hamiltonian dynamics is reversible.
    - Energy is constant $p(x, v) = p(x', v') = p(x', -v')$.
- How to simulate Hamiltonian dynamics? Take:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\partial H}{\partial v} = \frac{\partial K}{\partial v}$$
$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{\partial H}{\partial x} = -\frac{\partial E}{\partial x}$$

(Indeed: $\frac{\mathrm{d}H}{\mathrm{d}t} = \sum_i \frac{\partial E}{\partial x_i}\frac{\mathrm{d}x_i}{\mathrm{d}t} + \sum_i \frac{\partial H}{\partial v_i}\frac{\mathrm{d}v_i}{\mathrm{d}t}$ will be zero)

Leap-frog integrator

- A numerical approximation:

$$\begin{aligned} v(t+\tfrac{\epsilon}{2}) &= v(t) - \frac{\epsilon}{2}\frac{\partial E}{\partial x}(x(t)) \\ x(t+\epsilon) &= x(t) + \epsilon \frac{\partial K}{\partial v}(v(t+\tfrac{\epsilon}{2})) \\ v(t+\epsilon) &= v(t+\tfrac{\epsilon}{2}) - \frac{\epsilon}{2}\frac{\partial E}{\partial x}(x(t+\epsilon)) \end{aligned}$$

  (Slightly more accurate than the standard Euler's method)
- We do a fixed number of leap-frog steps.
- Dynamics are still deterministic (and reversible)
- Acceptance probability :

$$\min\left\{1, \frac{\exp(H(x,v))}{\exp(H(x',v'))}\right\}$$

HMC algorithm

The HMC algorithm (run until it mixes):

- Current position: x
- Sample momentum: $v \sim \mathcal{N}(0, I)$.
- Run Leapfrog integrator for L steps and reach (x', v')
- Accept new state $(x', -v')$ (or position x') with probability:

$$\min\left\{1, \frac{\exp(H(x,v))}{\exp(H(x',v'))}\right\}$$

- Low energy points are favored.

MCMC Inference

- Sample from unnormalized posterior.
- Estimate statistics from simulated values of x:
  - mean
  - median
  - quantiles
- **Posterior predictive distribution** of unobserved outcomes can be obtained by further simulation conditional on drawn values of x.
- All of this however requires some care, as MCMC is not without problems.

MCMC diagnostics

- How do we know we have ran the algorithm long enough?
- What if we started very far from where our distribution is?
- Since there is correlation between each item of the chain (autocorrelation), what is the "effective" number of samples?

Good Ideas for MCMC

- Parallel computation is cheap - we can run multiple chains in parallel starting at different points
- We should discard some initial samples - **burn-in phase**.
- We should examine how well the chain is "mixed".

R hat

- Start with m chains each of length n, $X = [x_{ij}] \in \mathbb{R}^{n \times m}$.
  - this will be already after a fixed burn-in phase.
- The **between sequence variance** B is:

$$B = \frac{n}{m-1} \sum_{j=1}^{m} (\bar{x}_{.j} - \bar{x}_{..})^2,$$

where:

$$\bar{x}_{.j} = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \quad \text{and} \quad \bar{x}_{..} = \frac{1}{m} \sum_{j=1}^{m} \bar{x}_{.j} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij}$$

(individual chain means, total mean)

R hat

- The **within sequence variance** W is:

$$W = \frac{1}{m} \sum_{j=1}^{m} s_j^2$$

where:

$$s_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_{.j})^2$$

- **Idea**: If one or more chain has not mixed well, the variance of all the chains combined together should be higher than that of individual chains.

R hat

- Next we compute the average variance:

$$\widehat{\text{var}}^{+}(x) = \frac{n-1}{n}W + \frac{1}{n}B$$

- Finally define **R-hat** coefficient:

$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}^{+}(x)}{W}}$$

- If chains have not mixed well, R-hat is larger than 1.
- **Split-$\widehat{\mathbf{R}}$**: Split each chain into the first and second halves. This can detect non-stationarity within a single chain.

Effective Sample Size

- If $x_1, \ldots, x_n$ are i.i.d. with variance σ^2 then $\text{var}(\bar{x}_n) = \frac{\sigma^2}{n}$.
- In general, without assuming independence

$$\text{var}(\bar{x}) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \text{cov}(x_i, x_j) = \frac{\sigma^2}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \text{corr}(x_i, x_j)$$

  so $\frac{n^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} \text{corr}(x_i, x_j)}$ measures "effective sample size".
- We define the **effective sample size** to be:

$$n_{\text{eff}} = \frac{mn}{1 + 2\sum_{t=1}^{\infty} \rho_t}$$

  where $\rho_t = \text{corr}(x_0, x_t)$ are unknown, so we also estimate them.

Diagnostics Summary

- Once $\hat{R}$ is near 1, and $\hat{n}_{\text{eff}}$ is more than 10 per chain **for all scalar estimands** we collect the mn simulations, (excluding the burn-in).
- We can then draw inference based on our samples. However:
  - Even if the iterative simulations appear to have converged, passed all tests etc. It may still be far from convergence!
- When we declare "convergence" - we mean that all chains appear stationary and well mixed.
- Non of the checks we learned today are hypothesis test. There are no p-values, and no statistical significance.