Gibbs Sampling

Prof. Dr. Narayan Prasad Adhikari
Central Department of Physics
Tribhuvan University Kirtipur, Kathmandu, Nepal

March 17, 2024

Assigned Problems

- Introduction
- Definition and Properties
- Implementation and optimization
- Forming the Sample
- Scanning strategies
- Using the sample
- Reparametrization
- Convergence diagnostics
- Applications

Introduction

- Gibbs sampling was originated in the context of image processing. In this context, the posterior of interest for sampling is a Gibbs distribution. Borrowing concepts from Mechanical Statistics, the density of the Gibbs distribution can be written as

$$f(x_1, x_2, ..., x_d) \propto \exp\left[\frac{-E(x_1, x_2, ..., x_d)}{kT}\right] \quad (1)$$

where k is a positive constant and T is absolute temperature. E is the energy of the system, a positive function, and x_i is the characteristic of interest for the ith component of the system, $i = 1, ..., d$. In Mechanical Statistics, x_i is the position or perhaps the velocity and position of the ith particle and in image processing it is (an indicator of) the colour of the ith pixel of an image.

Introduction

- The energy function E is commonly given by a sum of potential functions V. These sums operate over collections of subgroups of components over which each potential function is evaluated.
- The subgroups generally obey some neighboring relationship in their definition. This leads to a probability specification based on local properties, useful for modelling spatial interaction between components. The main drawback is the difficulty in the determination o f the global properties, such as the normalizing constant.
- Gibbs sampling scheme could in fact be used for a host of other posterior distributions.

In Physics

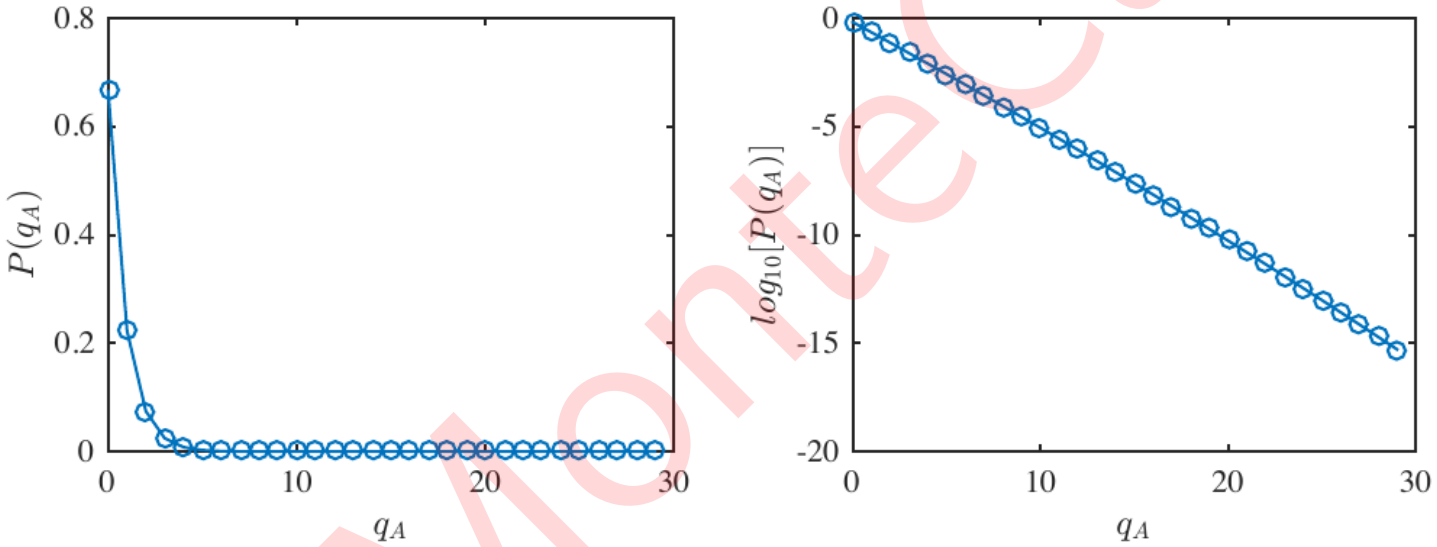


Figure: The Gibbs distribution gives how the energy is distributed in the system.

Definition and Properties

- The Gibbs sampler is a very useful tool for simulations of Markov processes for which the transition matrix cannot be formulated explicitly
- Gibbs sampling is a Markov Chain Monte Carlo (MCMC) scheme where the transition kernel is formed by the full conditional distributions. Let us assume as before that the distribution of interest is $\pi(\theta)$ where $\theta = (\theta_1, ..., \theta_d)'$. Each one of the components θ_i can be a scalar, a vector or a matrix. However in our case we consider them as scalar.

- Consider also that the full conditional distributions $\pi_i(\theta_i) = \pi(\theta_i|\theta_{-i}),\ i = 1, 2, ..., d$ are available.

- This means that they are completely known and can be sampled from.

Definition and Properties

- The problem to be solved is to draw from π when direct generation schemes are costly, complicated or simply unavailable but when generations from the π_i, are possible.
- Gibbs sampling provides an alternative generation scheme based on successive generations from the full conditional distributions.
- **To carry on the Gibbas sampling we use following algorithm (See next page)**

Definition and Properties -Algorithm

- Initialize the iteration counter of the chain $j = 1$ and set initial values

$$\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_d^{(0)})'$$

- Obtain a new value $\theta^{(j)} = (\theta_1^{(j)}, \theta_2^{(j)}, ..., \theta_d^{(j)})'$ from $\theta^{(j-1)}$ through successive generations of values

$$\theta_1^{(j)} = \pi(\theta_1 | \theta_2^{(j-1)}, \theta_3^{(j-1)}, ..., \theta_d^{(j-1)})'$$

$$\theta_2^{(j)} = \pi(\theta_2 | \theta_1^{(j)}, \theta_3^{(j-1)}, ..., \theta_d^{(j-1)})'$$

$$.....$$

$$.....$$

$$\theta_d^{(j)} = \pi(\theta_d | \theta_1^{(j)}, \theta_2^{(j)}, ..., \theta_{d-1}^{(j)})'$$

- Change counter j to $j + 1$ and return to above step until convergence is reached.

Definition and Properties

- When convergence is reached, the resulting value $\theta(j)$ is a draw from π. As the number of iterations increases, the chain approaches its equilibrium condition. Convergence is then assumed to hold approximately.

- The obvious form to obtain a sample of size n from π is to replicate n chains until convergence.

- Alternatively, after convergence all draws from a chain come from the stationary distribution. Therefore n successive values from this chain after the burn-in period will also provide a sample from π .

Algorithm - two variables case

- Data distribution:
- Assume that $y|\theta \sim N(\theta, \Sigma)$ is a bivariate normal distribution with unknown mean $\theta = (\theta_1, \theta_2)$ and known covariance matrix (in covariance matrix variances are given by diagonal elements whereas covariance is given by non-diagonal elements)

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

- Prior distribution: The prior for θ is an improper uniform over the real line i.e. $p(\theta_1, \theta_2) \propto 1$.
- Posterior distribution: Assuming we observe a single observation $y = (y_1, y_2)$,

$$\theta|y \sim N(y, \Sigma)$$

Algorithm - two variables case

- Full conditional distribution:

$$\theta_1|\theta_2, y \sim N(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2)$$
$$\theta_2|\theta_1, y \sim N(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2)$$

  Sample from the posterior distribution using a Gibbs sample assuming $y = (0, 0)$ and $\rho = 0.8$

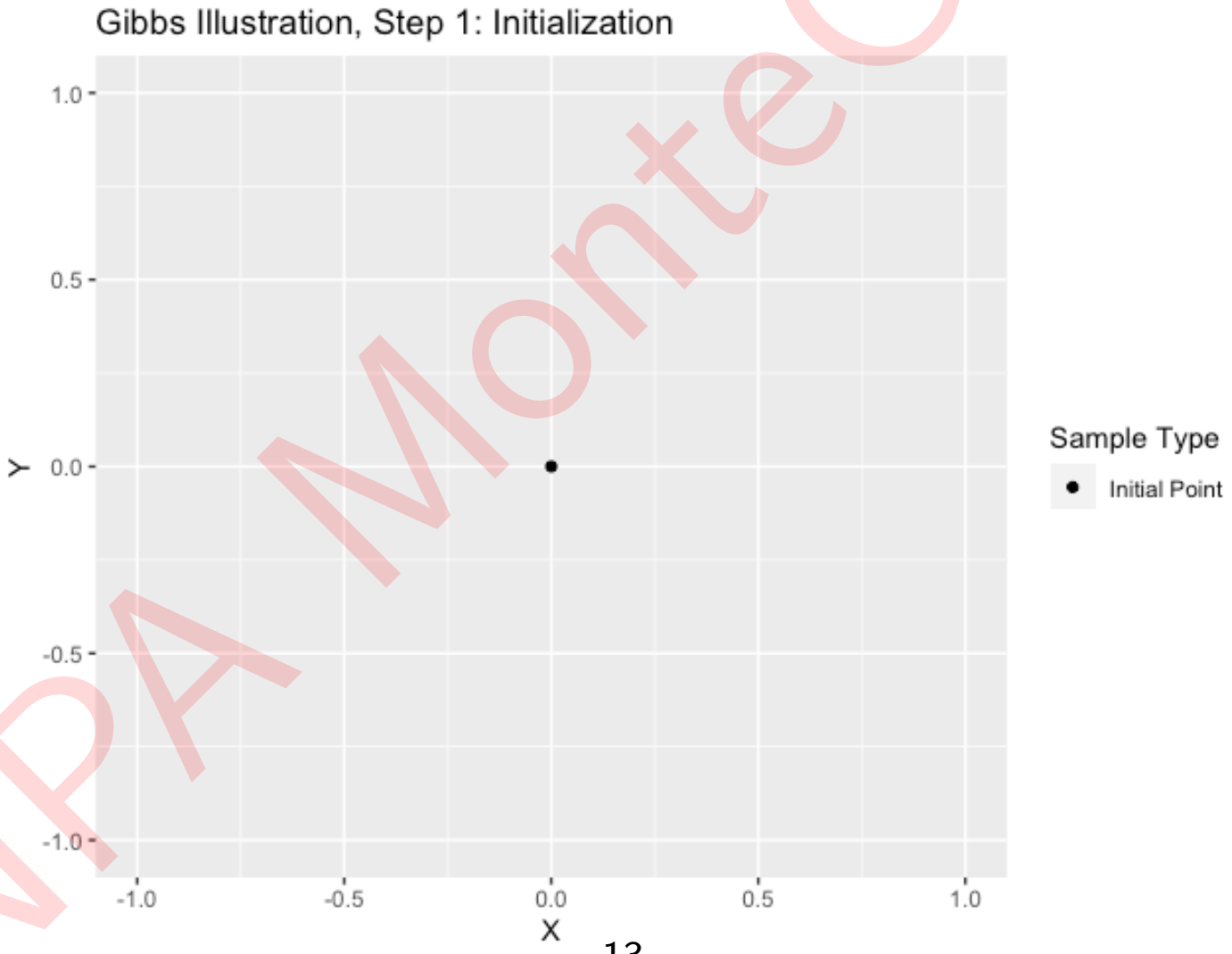
- **Algorithm and code**

Algorithm - two variables case

- The gist of the Gibbs sampler is simple: sample from known conditional distributions, and use that resulting value to sample the next random variable from the following conditional probability distribution, ad infinitum.
- Algorithm:
- 1. Initialize (x_0, y_0) and set time n or t =0
- 2. Draw x_t from conditional distribution $X_t|(Y_{t-1} = y_{t-1} \sim N(\rho y_{t-1}, 1 - \rho^2))$
- 3. Draw y_t from conditional distribution $y_t|(x_t = x_t \sim N(\rho x_t, 1 - \rho^2))$
- 4. Increase t=t+1
- 5. Return to step 2

Algorithm - two variables case

- Lets consider the case for $\rho = 0.9$ Step 1: Initialize $x_0 = 0.;\ y_0 = 0.$ also set the iteration counter n OR (t) to 0.

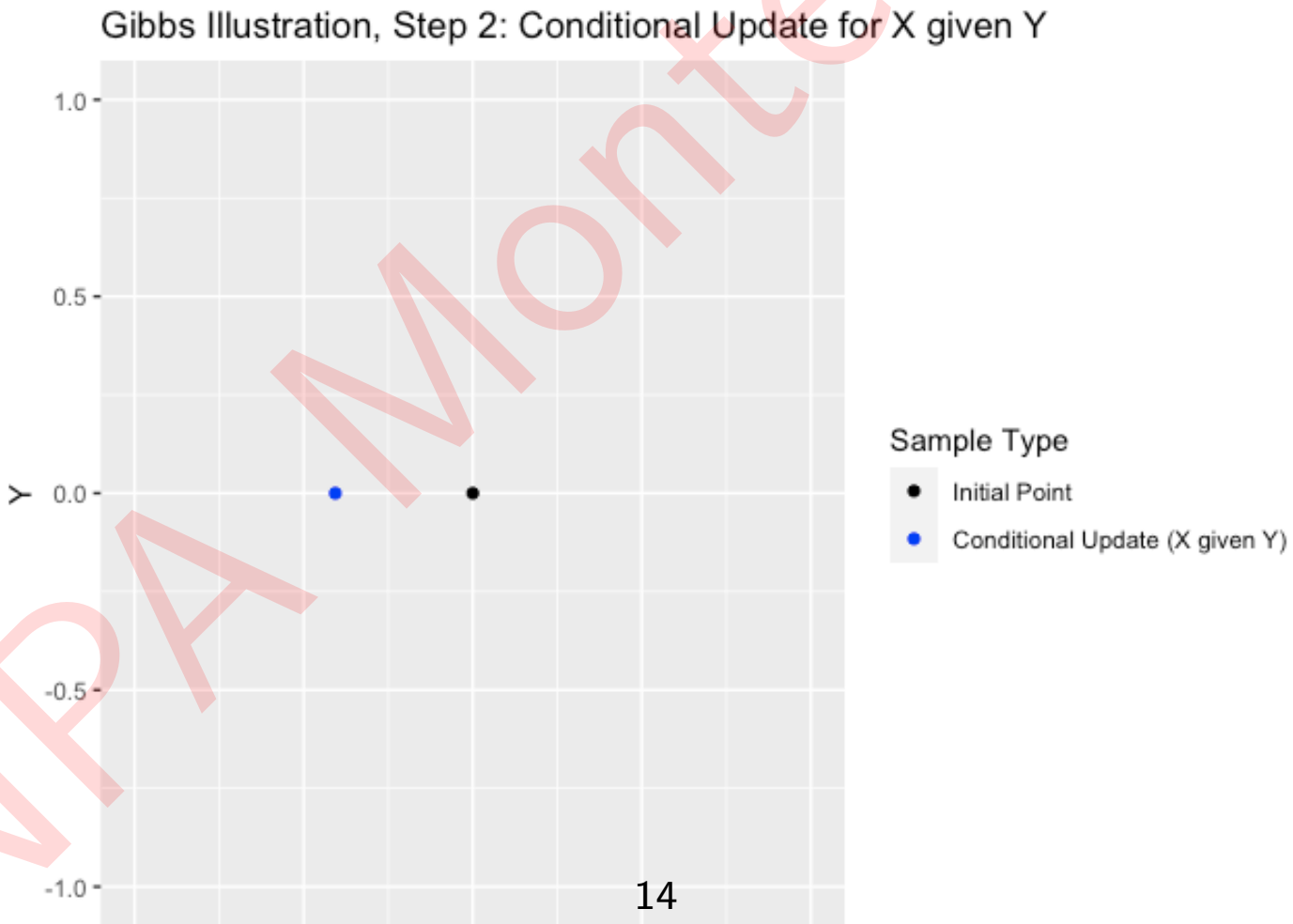


Algorithm - two variables case

- Step 2: Conditional update of X given Y

$$X_1 | \ (Y_0 = 0) \sim N(0 \times \rho, 1 - \rho^2)$$

In one of the case it was -0.4 as shown below.

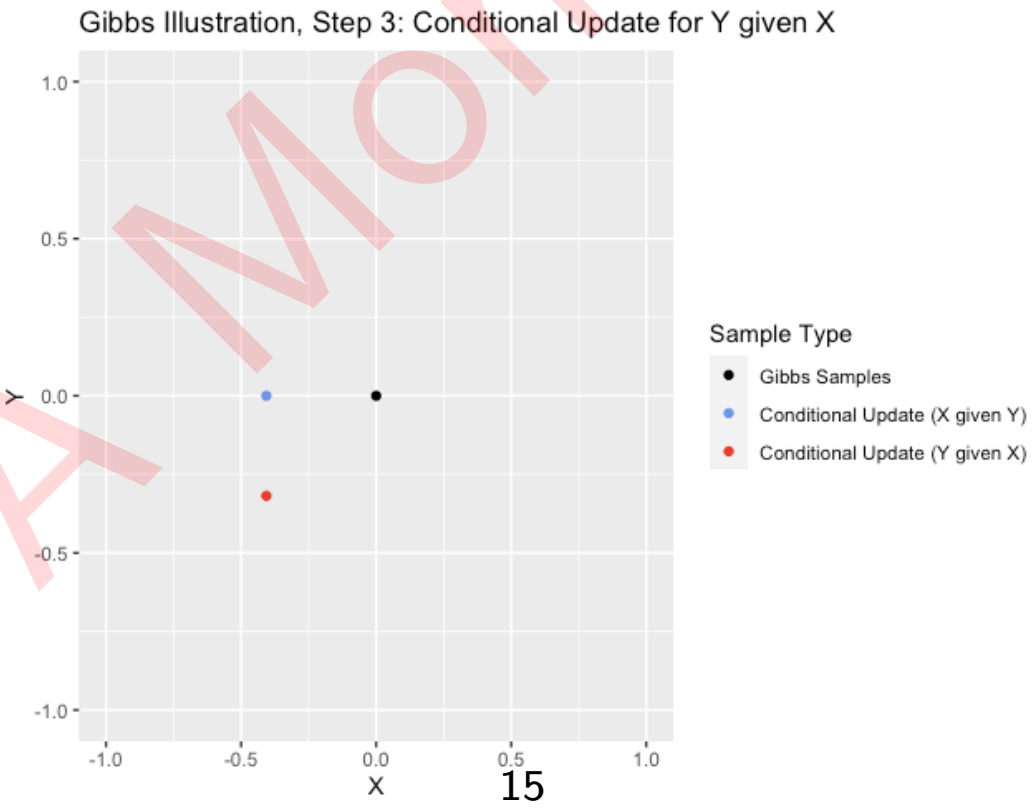


Algorithm - two variables case

- Step 3: Conditional update of Y given X

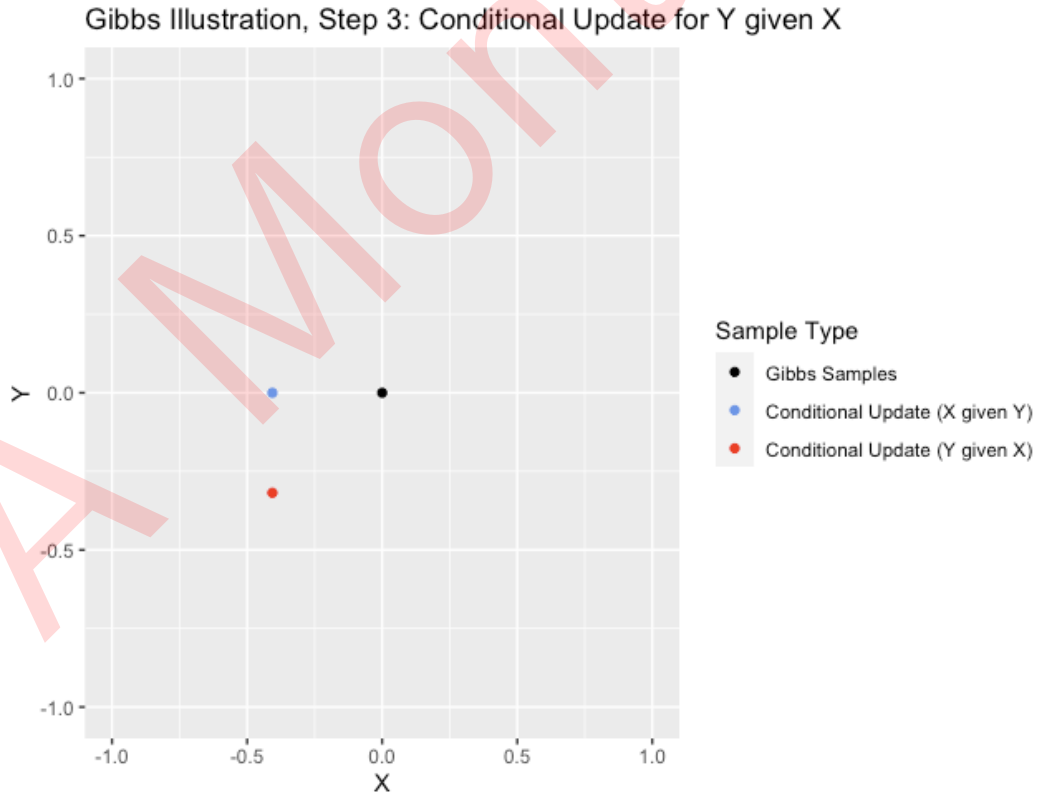
$$Y_1|\ (X_1 = -0.4) \sim N(-0.4 \times \rho, 1 - \rho^2)$$

In one of the case it was -0.32 as shown below. This time, the X coordinate of our new point is the same as the X coordinate of the point from step 2.



Algorithm - two variables case

- Now we consider more steps repeating above processes. In this way one can sample the conditional probability of X given Y and vice versa. For multivariate systems its not easy to handle analytically. Even in this bivariate case for many steps its not possible to solve this problem analytically.



Python code - two variables case

- Now we consider more steps repeating above processes. In this way one can sample the conditional probability of X given Y and vice versa. For multivariate systems its not easy to handle analytically. Even in this bivariate case for many steps its not possible to solve this problem analytically.