SDSC3001 Tutorial 5

Rejection Sampling and Gibbs Sampling

2024.10.10

Sampling

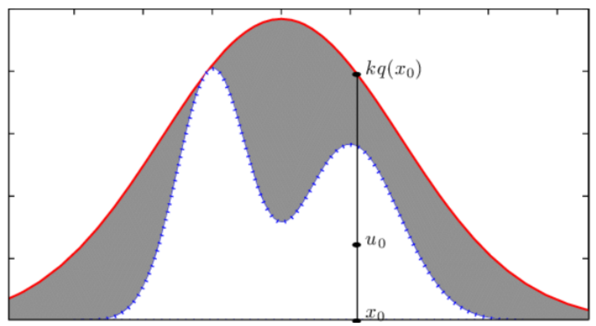
Importance sampling:
To approximate the expectation of a function f(X) and reduce variance (approximation error), we can utilize a new distribution q(X).

Rejection sampling(today)

Gibbs sampling (today)

Rejection Sampling

- Sample $x \sim q(x)$ and $u \sim U_{[0,1]}$
- If $u \leq \frac{p(x)}{kq(x)}$, accept x
- Otherwise, reject x



The algorithm, which was used by John von Neumann[4] and dates back to Buffon and his needle,[5] obtains a sample from distribution X with density f using samples from distribution Y with density g as follows:

- Obtain a sample y from distribution Y and a sample u from $\mathrm{Unif}(0, 1)$ (the uniform distribution over the unit interval).
- Check if $u < f(y)/Mg(y)$.
  - If this holds, accept y as a sample drawn from f;
  - if not, reject the value of y and return to the sampling step.

The algorithm will take an average of M iterations to obtain a sample.[6]

Rejection Sampling

- Probability of sampling x

$$q(x) \times \frac{p(x)}{kq(x)} = p(x)/k$$

- Probability of acceptance

$$\int q(x) \frac{p(x)}{kq(x)}\, dx = 1/k$$

- Accepted samples follow $p(x)$
- Limitations
  - Finding k may be impossible
  - If k is too large, acceptance rate is too small
  - Not efficient in high-dimensional space

Gibbs Sampling

Sample from a multivariate joint probability distribution with dimension n:

$$x^0, x^1, x^2, \ldots \sim p(x_1, x_2, x_3, \cdots, x_n)$$

Sometimes it’s intractable!

Gibbs Sampling

$$p(x_1 | x_2, x_3, \cdots, x_n)$$

$$p(x_2 | x_1, x_3, \cdots, x_n)$$

$$p(x_1, x_2, x_3, \cdots, x_n) \longrightarrow p(x_3 | x_1, x_2, \cdots, x_n)$$

Joint distribution

$$\vdots$$

$$p(x_n | x_1, x_2, \cdots, x_{n-1})$$

Marginal probability distributions is accessible and simpler

Gibbs Sampling

Start from a random vector $(x_1^0, x_2^0, x_3^0, \cdots, x_n^0)$

$$x_1^1 \sim p(x_1 \mid x_2^0, x_3^0, \ldots, x_n^0)$$

We can use the result from earlier, namely x_1^1:

$$x_2^1 \sim p(x_2 \mid x_1^1, x_3^0, \ldots, x_n^0)$$

This might help us yield a slightly more convincing result than simply using the random data.

We still have to use random values for x_3 through x_n since we haven' t sampled from their relevant marginal distributions just yet.

Gibbs Sampling

Implementation [edit]

Gibbs sampling, in its basic incarnation, is a special case of the Metropolis–Hastings algorithm. The point of Gibbs sampling is that given a multivariate distribution it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. Suppose we want to obtain k samples of $\mathbf{X} = (x_1, \ldots, x_n)$ from a joint distribution $p(x_1, \ldots, x_n)$. Denote the ith sample by $\mathbf{X}^{(i)} = \left(x_1^{(i)}, \ldots, x_n^{(i)}\right)$. We proceed as follows:

1. We begin with some initial value $\mathbf{X}^{(0)}$.
2. We want the next sample. Call this next sample $\mathbf{X}^{(i+1)}$. Since $\mathbf{X}^{(i+1)} = \left(x_1^{(i+1)}, x_2^{(i+1)}, \ldots, x_n^{(i+1)}\right)$ is a vector, we sample each component of the vector, $x_j^{(i+1)}$, from the distribution of that component conditioned on all other components sampled so far. But there is a catch: we condition on $\mathbf{X}^{(i+1)}$'s components *up to* $x_{j-1}^{(i+1)}$, and thereafter condition on $\mathbf{X}^{(i)}$'s components, starting from $x_{j+1}^{(i)}$ to $x_n^{(i)}$. To achieve this, we sample the components in order, starting from the first component. More formally, to sample $x_j^{(i+1)}$, we update it according to the distribution specified by $\boxed{p\left(x_j^{(i+1)} | x_1^{(i+1)}, \ldots, x_{j-1}^{(i+1)}, x_{j+1}^{(i)}, \ldots, x_n^{(i)}\right).}$ We use the value that the $(j+1)$th component had in the ith sample, not the $(i+1)$th sample.
3. Repeat the above step k times.

Gibbs Sampling

As we go through all the random variables in order, it becomes obvious that we will no longer be using randomly initialized values at one point.

Specifically, to generate k-th sample by update element on dimension m:

$$x_m^{'} \sim p(x_m | x_1^{k-1}, x_2^{k-1}, \dots, x_{m-1}^{k-1}, x_{m+1}^{k-1}, \dots, x_n^{k-1})$$

$$(x_1^{k-1}, x_2^{k-1}, \dots, x_{m-1}^{k-1}, {\color{red}x_m^{'}}, x_{m+1}^{k-1}, \dots, x_n^{k-1})$$

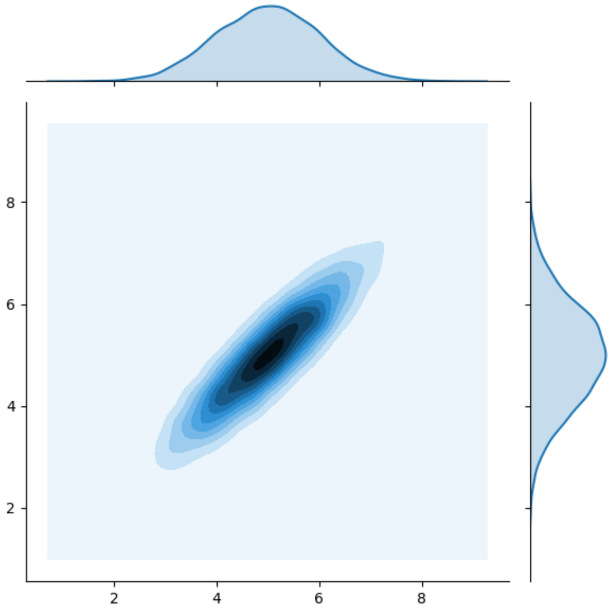
Gibbs Sampling

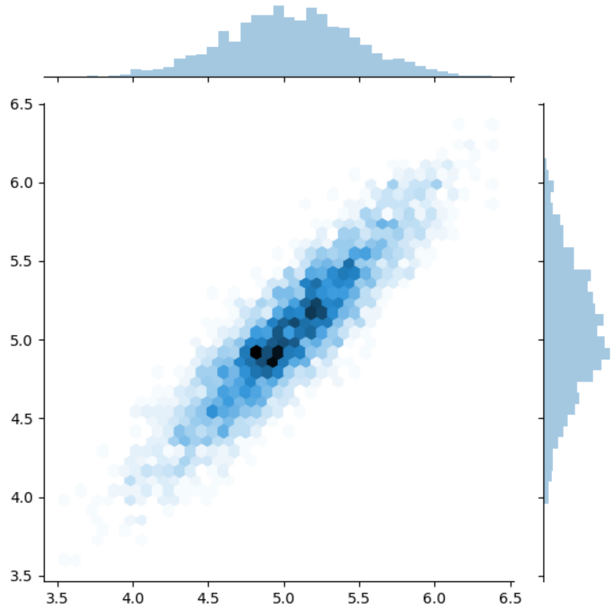
- Sample from a multivariate distribution $p(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \ldots, x_d)$
- Marginal distribution $p(x_i \mid \mathbf{x}_{\neg i})$, $\mathbf{x}_{\neg i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$
- Given the current $\mathbf{x}$
  - Randomly choose a coordinate i
  - Sample y_i based on $p(x_i \mid \mathbf{x}_{\neg i})$
  - Set next sample $\mathbf{y}$ as $(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_d)$

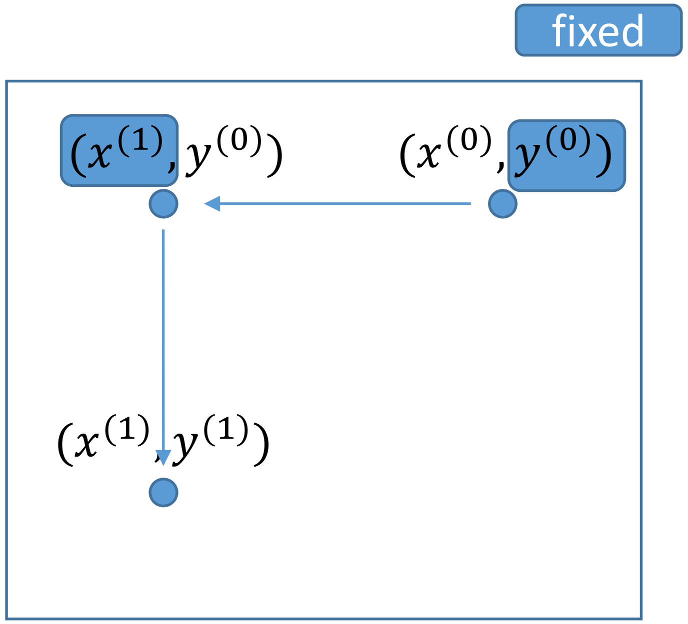
$(x^{(0)}, y^{(0)})$

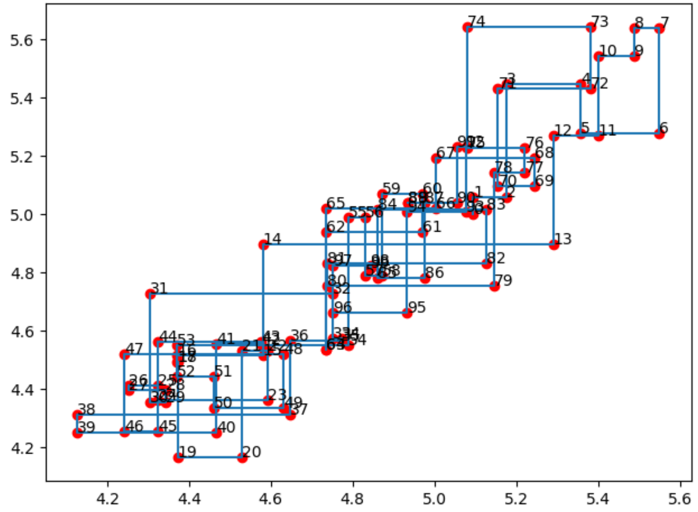
Gibbs Sampling

An Example: Generate samples from 2D-Gaussian distribution









Thank you!