1

Machine Learning

Chapter 11: Sampling Methods

林嘉文 (Chia-Wen Lin)
清華大學電機系
cwlin@ee.nthu.edu.tw

5/30/2022

2

Goal of Sampling Methods

- Fundamental problem: find the expectation of some function $f(\mathbf{z})$ with respect to a probability distribution $p(\mathbf{z})$

$$\mathbb{E}(f) = \int p(\mathbf{z}) f(\mathbf{z}) d\mathbf{z}$$

- Idea: If we obtain a set of samples $\mathbf{z}^{(l)}, l = 1, \ldots, L$ drawn independently from $p(\mathbf{z})$, the expectation may be approximated by

$$\hat{f} = \frac{1}{L} \sum_{l=1}^{L} f\left(\mathbf{z}^{(l)}\right)$$

- New problem: How can we obtain independent samples from a distribution $p(\mathbf{z})$ we do not know how to sample from?

5/30/2022

3

Basic Sampling: The Transformation Method

- We aim at sampling from $p(\cdot)$
- Suppose that we have available samples z uniformly distributed over the interval $(0,1)$ (i.e., $p(z) = 1$, for $z \in (0,1)$)
- Let $y = f(z)$, then the PDF of y: $p(y) = p(z)\left|\frac{dz}{dy}\right|$
- Transforming z into y using $y = h^{-1}(z)$ where h is defined as the cumulative distribution function of $p(\cdot)$:

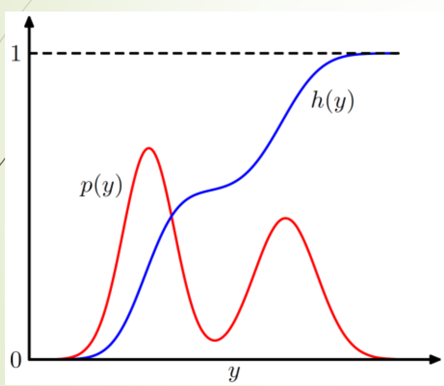
$$z = h(y) \equiv \int_{-\infty}^{y} p(\hat{y})d\hat{y}$$

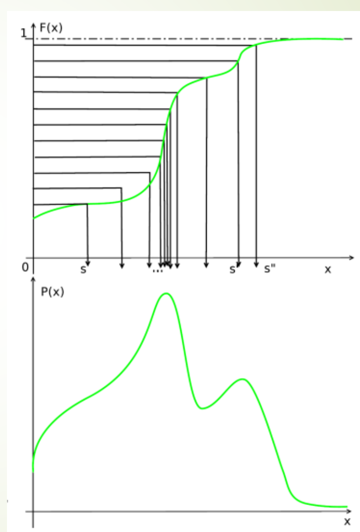
- Then $y = h^{-1}(z)$ are independent samples drawn from $p(\cdot)$
  - e.g., exponential distribution: $p(y) = \lambda e^{-\lambda y}$, $z = h(y) = 1 - e^{-\lambda y} \Rightarrow y = -\lambda^{-1}\ln(1-z)$
- The success of the transformation method depends on the ability to calculate and then invert the CDF (only feasible for simple functions)

5/30/2022

4

Basic Sampling: The Transformation Method





5/30/2022

5

Rejection Sampling

- Rejection sampling allows us to sample from relatively complex distributions, subject to certain constraints
- Assumption 1: Sampling directly from $p(\mathbf{z})$ is difficult but we are able to evaluate $p(\mathbf{z})$ for any value of $\mathbf{z}$ up to an unknown normalizing constant Z_p

$$p(z) = \frac{1}{Z_p}\tilde{p}(z)$$

- Assumption 2: We know how to sample from a proposal distribution $q(z)$, and there exists a constant k such that

$$kq(z) \geq \tilde{p}(z)$$

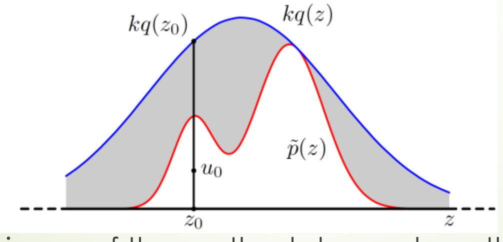
- Then we know how to obtain independent samples from $p(\cdot)$ from samples drawn from $q(z)$ based on $kq(z)$ (comparison function)

5/30/2022

6

Rejection Sampling

- Generate a number z_0 from $q(\cdot)$
- Generate a number u_0 from the uniform distribution over $[0, kq(z_0)]$
- If $u_0 > \tilde{p}(z_0)$ then the sample is rejected, otherwise z_0 is kept (with a probability of $\tilde{p}(z)/kq(z)$)
- The set of kept z are distributed according to $p(\cdot)$



$$p(\text{accept}) = \int \{\tilde{p}(z)/kq(z)\}\, q(z)dz$$

$$= \frac{1}{k}\int \tilde{p}(z)dz$$

- Efficiency of the method depends on the ratio between the grey area and the white area → the proposal distribution $q(\cdot)$ should be as close as possible to $p(\cdot)$

5/30/2022

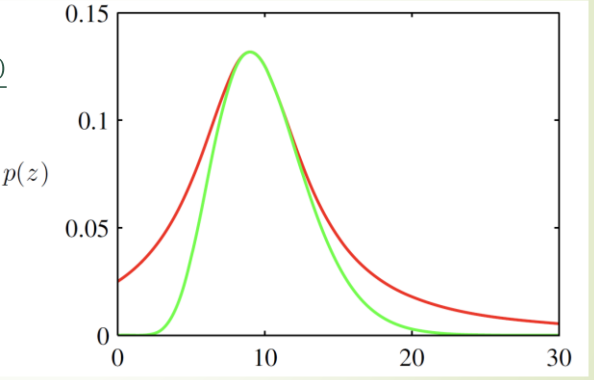
7

Rejection Sampling: Example

$$p(z) = \mathrm{Gam}(z|a,b) = \frac{b^a z^{a-1}\exp(-bz)}{\Gamma(a)}$$

Proposal (envelop) distribution:

$$q(z) = \frac{k}{1+(z-c)^2/b^2}$$

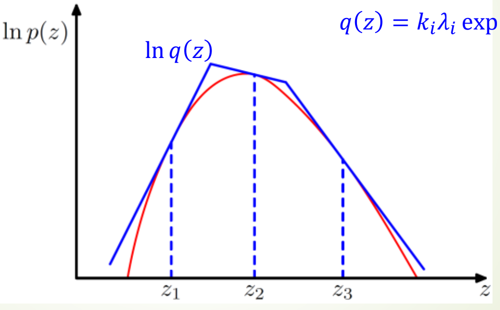


5/30/2022

8

Adaptive Rejection sampling

- The proposal distribution $q(\cdot)$ may be constructed on the fly
- If p is log-concave (i.e., $\ln p(z)$ is concave), use of the derivative of ln p at some given grid points



$$q(z) = k_i \lambda_i \exp\left(-\lambda_i(z - z_{i-1})\right), \ z_{i-1} < z \le z_i$$

- In any cases, rejection sampling methods are inefficient if sampling in high dimension (exponential decrease of acceptance rate with dimensionality)

5/30/2022

9

Importance Sampling

- Provides a framework for approximating expectations $\mathbb{E}(f) = \int p(\mathbf{z}) f(\mathbf{z}) d\mathbf{z} \approx \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{z}^{(l)})$ directly (but not for drawing samples from $p(\mathbf{z})$)
- Technique again based on the use of a proposal distribution $q(\cdot)$

$$\mathbb{E}(f) = \int f(\mathbf{z}) \frac{p(\mathbf{z})}{q(\mathbf{z})} q(\mathbf{z}) d\mathbf{z} \approx \frac{1}{L}\sum_{l=1}^{L} \frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})} f(\mathbf{z}^{(l)})$$

- Let $p(z) = \tilde{p}(z)/Z_p$ where Z_p is unknown, Similarly, $q(z) = \tilde{q}(z)/Z_q$,

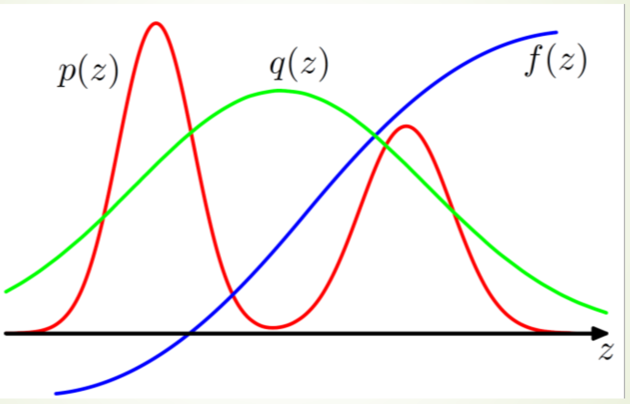
$$\mathbb{E}(f) = \frac{Z_q}{Z_p} \int f(\mathbf{z}) \frac{\tilde{p}(\mathbf{z})}{\tilde{q}(\mathbf{z})} q(\mathbf{z}) d\mathbf{z} \approx \frac{Z_q}{Z_p} \frac{1}{L} \sum_{l=1}^{L} \tilde{r}_l f(\mathbf{z}^{(l)}) \approx \sum_{l=1}^{L} w_l f(\mathbf{z}^{(l)})$$

where $w_l = \frac{\tilde{r}_l}{\sum_m \tilde{r}_m} = \frac{\tilde{p}(\mathbf{z}^{(l)})/q(\mathbf{z}^{(l)})}{\sum_m \tilde{p}(\mathbf{z}^{(m)})/q(\mathbf{z}^{(m)})}$, and $\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int \tilde{p}(\mathbf{z}) d\mathbf{z} = \int \frac{\tilde{p}(\mathbf{z})}{\tilde{q}(\mathbf{z})} q(\mathbf{z}) d\mathbf{z} \approx \frac{1}{L}\sum_m \tilde{r}_m$

- The importance weights $\frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}$ correct the bias introduced by sampling from a wrong distribution. w_l are the normalized importance weights
- All the generated samples are retained

10

Importance Sampling



Again, the success of the method depend on how well the proposal $q(\cdot)$ fits the desired distribution $p(\cdot)$. In particular, $q(z)$ should not be small or zero in regions where $p(z)$ may be significant.

11

Sampling-Importance-Resampling (SIR)

- Importance sampling is an alternative to rejection sampling
- Technique based on the use of a proposal distribution $q(\cdot)$ the assumption on the existence of a constant k is relaxed

1. Draw L samples $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \cdots, \mathbf{z}^{(L)}$ from $q(\mathbf{z})$
2. Calculate the importance weights $\frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}, \forall l = 1, \cdots, L$
3. Normalize the weights to obtain $w_1, \cdots, w_L$ $\qquad w_l = \frac{\tilde{p}(\mathbf{z}^{(l)})/q(\mathbf{z}^{(l)})}{\sum_m \tilde{p}(\mathbf{z}^{(m)})/q(\mathbf{z}^{(m)})}$
4. Draw a second set of L samples from the discrete distribution $(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \cdots, \mathbf{z}^{(L)})$ with probabilities $(w_1, \cdots, w_L)$

- The resulting L samples are distributed according to $p(\mathbf{z})$ if $L \to \infty$

12

Sampling and the EM algorithm

Monte Carlo EM algorithm

- Use some Monte Carlo methods to approximate the expectation of the E step
- The expected complete-data log likelihood, given by ($\mathbf{Z}$ hidden ; $\mathbf{X}$ observed ; $\boldsymbol{\theta}$ parameters):

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \int p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{Z}, \mathbf{X}|\boldsymbol{\theta})\, d\mathbf{Z}$$

may be approximate by (where $\mathbf{Z}^{(l)}$ are drawn from $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$)

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) \approx \frac{1}{L} \sum_{l=1}^{L} \ln p(\mathbf{Z}^{(l)}, \mathbf{X}|\boldsymbol{\theta})$$

Stochastic EM algorithm

- Considering a finite mixture model, only one sample $\mathbf{Z}$ may be drawn at each E step (meaning a hard assignment of each data point to one of the components in the mixture)

(Metropolis and Ulam, 1949)

13

Markov Chain Monte Carlo (MCMC)

- MCMC : general strategy which allows sampling from a large class of distributions
- MCMC scales well with the dimensionality of the sample space ≠ importance sampling / rejection sampling
- MCMC uses the mechanism of Markov chains
- Goal: to generate a set of samples from $p(\mathbf{z})$
- Assumption: we know how to evaluate $\tilde{p}(\mathbf{z})$

$$\tilde{p}(\mathbf{z}) = Z_p p(\mathbf{z})$$

5/30/2022

14

MCMC: Idea

- Goal: to generate a set of samples from $p(\mathbf{z})$
- Idea: to generate samples from a Markov Chain whose invariant distribution is $p(\mathbf{z})$
  1. Knowing the current sample is $\mathbf{z}^{(\tau)}$, generate a candidate sample $\mathbf{z}^*$ from a proposal distribution $q(\mathbf{z}|\mathbf{z}^{(\tau)})$ we know how to sample from
  2. Accept the sample according to an appropriate criterion
  3. If the candidate sample is accepted then $\mathbf{z}^{(\tau+1)} = \mathbf{z}^*$; otherwise $\mathbf{z}^{(\tau+1)} = \mathbf{z}^{(\tau)}$
  - The proposal distribution depends on the current state
  - Samples $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \cdots$ form a Markov chain

5/30/2022

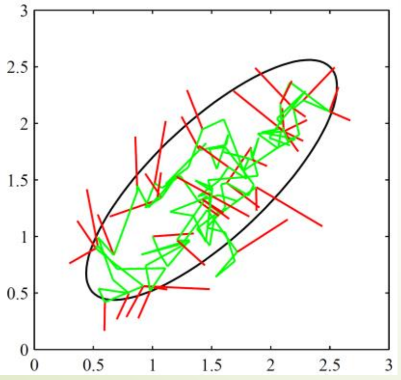
(Metropolis *et al.*, 1953)

15

Metropolis Algorithm

- the proposal distribution is symmetric: $q(\mathbf{z}_A|\mathbf{z}_B) = q(\mathbf{z}_B|\mathbf{z}_A)$
- The candidate sample is accepted with probability

$$A\left(\mathbf{z}^*, \mathbf{z}^{(\tau)}\right) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})}\right)$$



Use of Metropolis algorithm to sample from a Gaussian distribution. The proposal distribution is an isotropic Gaussian whose $\sigma = 0.2$. Accepted steps in green, rejected steps in red. 150 candidate samples, 43 rejected.

5/30/2022

16

MCMC: Why Is It Working?

- Idea: to generate samples from a Markov Chain whose invariant distribution is $p(\mathbf{z})$
- Why is it working ? Under what circumstances will a Markov chain converge to the desired distribution
  - First order Markov chain: series of random variables $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \cdots, \mathbf{z}^{(M)}$ such that for $m \in \{1, \cdots, M-1\}$
  $$p\left(\mathbf{z}^{(m+1)} \middle| \mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(m)}\right) = p\left(\mathbf{z}^{(m+1)} \middle| \mathbf{z}^{(m)}\right)$$
  - A Markov chain is specified by $p(\mathbf{z}^{(0)})$ and the transition probabilities
  $$T_m\left(\mathbf{z}^{(m)}, \mathbf{z}^{(m+1)}\right) \equiv p\left(\mathbf{z}^{(m+1)} \middle| \mathbf{z}^{(m)}\right)$$

5/30/2022

17

MCMC: Why Is It Working?

- A Markov chain is called homogeneous is the transition probabilities are the same for all m
- A distribution $p^*(\mathbf{z})$ is said to be invariant/stationary w.r.t a Markov chain if each transition leaves the distribution invariant

$$p^*(\mathbf{z}) = \sum_{\mathbf{z}'} T(\mathbf{z}', \mathbf{z}) p^*(\mathbf{z}')$$

- A sufficient condition for ensuring $p^*(\mathbf{z})$ to be invariant is to choose the transition probabilities to satisfy the property of detailed balance:

$$p^*(\mathbf{z}) T(\mathbf{z}, \mathbf{z}') = T(\mathbf{z}', \mathbf{z}) p^*(\mathbf{z}')$$

- A Markov chain that respects the detailed balance is said to be reversible
- A Markov chain is said ergodic if it converges to the invariant distribution irrespective of the choice of the initial distribution

5/30/2022

18

MCMC: Why Is It Working?

- Goal: to generate a set of samples from $p(\mathbf{z})$
- Idea: to generate samples from a Markov Chain whose invariant distribution is $p(\mathbf{z})$
- How: choose the transition probability $T(\mathbf{z}, \mathbf{z}^*)$ to satisfy the property of detailed balance for $p(\mathbf{z})$
- Remark: $T(\mathbf{z}, \mathbf{z}^*)$ can be a mixture distribution

$$T(\mathbf{z}, \mathbf{z}^*) = \sum_{k=1}^{K} \alpha_k B_k(\mathbf{z}, \mathbf{z}^*)$$

5/30/2022

(Hasting, 1970)

19

The Metropolis-Hasting algorithm

- Generalization of the Metropolis algorithm
- The proposal distribution q is no longer symmetric
- Knowing the current sample is $\mathbf{z}^{(\tau)}$, generate a candidate sample $\mathbf{z}^*$ from a proposal distribution $q_k(\mathbf{z}|\mathbf{z}^{(\tau)})$ (k: the members of the set of possible transitions being considered)
- Accept it with probability

$$A_k(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)q_k(\mathbf{z}^{(\tau)}|\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})q_k(\mathbf{z}^*|\mathbf{z}^{(\tau)})}\right)$$

- This reduces to the standard Metropolis criterion if $q_k(\mathbf{z}^{(\tau)}|\mathbf{z}^*) = q_k(\mathbf{z}^*|\mathbf{z}^{(\tau)})$ (i.e., symmetric)

20

The Metropolis-Hasting algorithm

- Transition probability of this chain: $T(\mathbf{z}, \mathbf{z}') = q(\mathbf{z}'|\mathbf{z})A(\mathbf{z}', \mathbf{z})$
- To prove that $p(\mathbf{z})$ is the invariant distribution of the chain, it is sufficient to prove the property of detailed balance

$$p(\mathbf{z})T(\mathbf{z}, \mathbf{z}') = T(\mathbf{z}', \mathbf{z})p(\mathbf{z}')$$

21

The Metropolis-Hasting algorithm

- Transition probability of this chain: $T(\mathbf{z},\mathbf{z}') = q(\mathbf{z}'|\mathbf{z})A(\mathbf{z}',\mathbf{z})$
- To prove that $p(\mathbf{z})$ is the invariant distribution of the chain, it is sufficient to prove the property of detailed balance

$$p(\mathbf{z})T(\mathbf{z},\mathbf{z}') = T(\mathbf{z}',\mathbf{z})p(\mathbf{z}')$$

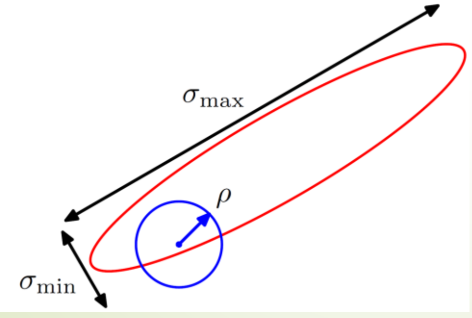
$$A(\mathbf{z}',\mathbf{z}) = \min\left(1, \frac{p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')}{p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})}\right)$$

$$\begin{aligned} & p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})A(\mathbf{z}',\mathbf{z}) \\ &= \min\big(p(\mathbf{z})q(\mathbf{z}'|\mathbf{z}), p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')\big) \\ &= \min(p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}'), p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})) \\ &= p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}') \min\left(1, \frac{p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})}{p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')}\right) \\ &= p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')A(\mathbf{z},\mathbf{z}') \end{aligned}$$

22

The Metropolis-Hasting algorithm

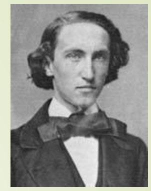
- Common choice for q: Gaussian centered on the current state
  - Small variance → high rate of acceptation but slow exploration of the state space + non independent samples
  - Large variance → high rate of rejection



Use of an isotropic Gaussian proposal (blue circle), to sample from a Gaussian distribution (red). The scale ρ of the proposal should be on the order of $\sigma_{\min}$, but the algorithm may have low convergence (low to explore the state space in the other direction)

23

Gibbs Sampling


Josiah Willard Gibbs

- Special case of the Metropolis-Hasting algorithm
- Objective law: $p(\mathbf{z}) = p(z_1, \cdots, z_M)$
- $p(z_i | \mathbf{z}_{\setminus i})$ known
  - Each step of the Gibbs sampling procedure involves replacing the value of one of the variables z_i by a value drawn from $p(z_i | \mathbf{z}_{\setminus i})$, the distribution of z_i conditioned on the values of the remaining variables
  - The procedure is repeated either by cycling through the variables in some order, or by choosing the variable to be updated at each step from some distribution

5/30/2022

24

Gibbs Sampling – Example

- The objective law is $p\left(z_1^{(i)}, z_2^{(i)}, z_3^{(i)}\right)$
- At step i, we have selected values $z_1^{(i)}, z_2^{(i)}, z_3^{(i)}$
- Then we obtain $z_1^{(i+1)}, z_2^{(i+1)}, z_3^{(i+1)}$ with

$$z_1^{(i+1)} \sim p\left(z_1 \middle| z_2^{(i)}, z_3^{(i)}\right)$$
$$z_2^{(i+1)} \sim p\left(z_2 \middle| z_1^{(i+1)}, z_3^{(i)}\right)$$
$$z_3^{(i+1)} \sim p\left(z_3 \middle| z_1^{(i+1)}, z_2^{(i+1)}\right)$$

- If, instead of drawing a sample from the conditional distribution, we replace the variable by the maximum of the conditional distribution, we obtain the iterated conditional modes (ICM) algorithm

5/30/2022

25

Gibbs Sampling

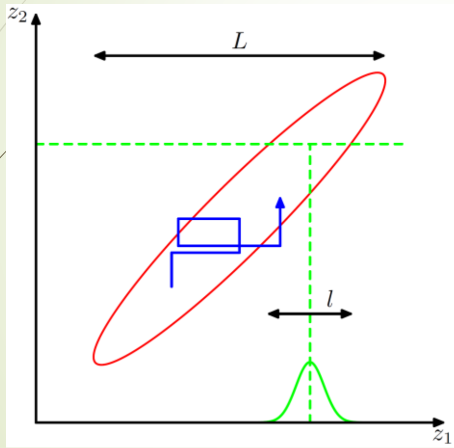


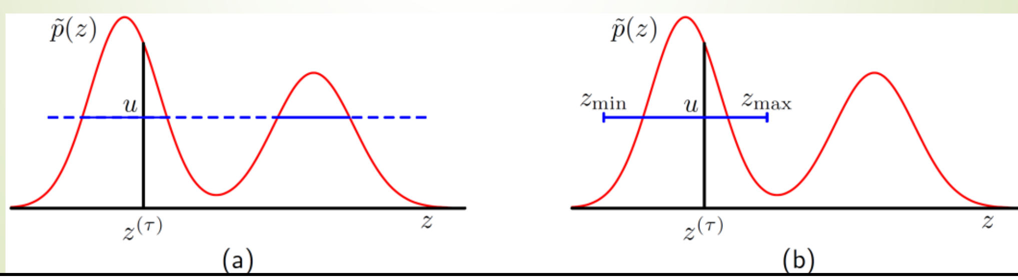
Illustration of Gibbs sampling, by alternate updates of two variables (blue steps) whose distribution is a correlated Gaussian (red). The conditional distributions are Gaussian (green curve).
Elise Arnaud,

5/30/2022

26

Slice Sampling

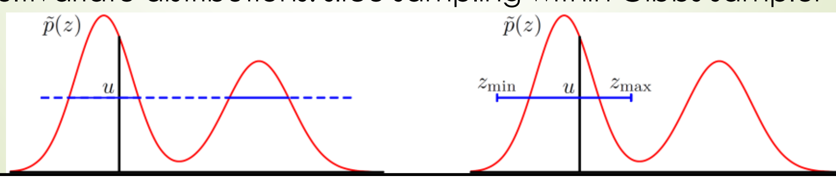
- Problem of Metropolis algorithm (proposal $q(\mathbf{z}|\mathbf{z}') = q(\mathbf{z}'|\mathbf{z})$)
  - Step size too small, slow convergence (random walk behavior)
  - Step size too large, high estimator variance (high rejection rate)
- Idea: adapt the step size automatically to a suitable value
- Technique: introduce variable u and sample (u, z) jointly
  - Ignoring u leads to the desired samples of $p(z)$



27

Slice Sampling

- Sample z and u uniformly from area under the distribution
  - Fix z, sample u uniform from $[0, \tilde{p}(z)]$
  - Fix u, sample z uniform from slice : $\{z: \tilde{p}(z) > u\}$
- How to sample z from the slice? [Neal, 2003]
  - Start with region of width w containing $z^{(\tau)}$
  - If end point in slice, then extend region by w in that direction
  - Sample z' uniform from region
  - If z' in slice, then accept as $z^{(\tau+1)}$
  - If not: make z' new end point of the region, and resample z'
- Multivariate distributions: slice sampling within Gibbs sampler



28

Hybrid Monte Carlo

- Problem of Metropolis algorithm is the step size trade-off
- Hybrid Monte Carlo suitable in continuous state spaces
  - able to make large jumps in state space
  - low rejection rate
  - based on dynamical systems
  - need to evaluate gradient of log-prob. w.r.t. state $\mathbf{z}$
  - based on Hamiltonian Dynamics
- Goal is to sample from

$$p(\mathbf{z}) = \frac{1}{Z_p} \exp(-E(\mathbf{z}))$$

  where $E(\mathbf{z})$ is interpreted as potential energy of system in $\mathbf{z}$
- Hamiltonian dynamical system over energy landscape $E(\mathbf{z})$

29

Hybrid Monte Carlo: Hamiltonian Dynamics

- Evolution of state variable $\mathbf{z} = \{z_i\}$ under continuous time
- Momentum variables correspond to rate of change of state

$$r_i = \frac{dz_i}{d\tau}$$

- Joint $(\mathbf{z}, \mathbf{r})$ space is called phase space
- Rate of change of momentum (acceleration, applied force):

$$\frac{dr_i}{d\tau} = -\frac{dE(\mathbf{z})}{dz_i}$$

Hamiltonian is constant under dynamical system evolution

$$H(\mathbf{z}, \mathbf{r}) = E(\mathbf{z}) + K(\mathbf{r}), \text{ with } K(\mathbf{r}) = \frac{1}{2}\sum_i r_i^2$$

5/30/2022

30

Hybrid Monte Carlo: Distribution over Phase Space

- Let Hamiltonian define a distribution over phase space:
- Momentum variables correspond to rate of change of state

$$p_H(\mathbf{z}, \mathbf{r}) = \frac{1}{Z_H}\exp\big(-H(\mathbf{z}, \mathbf{r})\big) = \frac{1}{Z_H}\ \exp\big(-E(\mathbf{z}) - K(\mathbf{r})\big)$$

  - Sate $\mathbf{z}$ and momentum $\mathbf{r}$ are independently distributed
- Hamiltonian dynamics leave this distribution invariant!
  - If $(\mathbf{z}, \mathbf{r}) \sim p_H$, and evolve in time τ to $(\mathbf{z}^*, \mathbf{r}^*)$, then also $(\mathbf{z}^*, \mathbf{r}^*) \sim p_H$
  - volume and H are constant under Hamiltonian dynamics
- Hamiltonian evolution is not an ergodic sampler of p_H
  - Resampling of r using $p_H(\mathbf{r}|\mathbf{z}) = p_H(\mathbf{r})$
  - Gibbs sampling step

5/30/2022

31

Hybrid Monte Carlo

- Combination of Metropolis algorithm & Hamitonian Dynamics
- Markov chain that alternates
  - stochastic update of the momentum variables $\mathbf{r}$
  - Hamiltonian dynamical updates with acceptance probability

$$\min(1, \exp\{H(\mathbf{z}, \mathbf{r}) - H(\mathbf{z}^*, \mathbf{r}^*)\})$$

  to compensate for numerical errors: if $H(\mathbf{z}, \mathbf{r}) \neq H(\mathbf{z}^*, \mathbf{r}^*)$
- Direction of time is chosen randomly to have detailed balance
- Hybrid Monte Carlo generates independent samples faster
  - the use of gradient information causes this difference

5/30/2022

32

Estimating the Partition Function

- Most sampling algorithms require distribution up to the constant partition function Z_E:

$$p_E(\mathbf{z}) = \frac{1}{Z_E} \exp\left(-E(\mathbf{z})\right)$$

$$Z_E = \sum_{\mathbf{z}} \exp\left(-E(\mathbf{z})\right)$$

- Partition function is useful for model comparison
  - it expresses the probability of observed data in

$$p(\text{hidden}|\text{observed}) = \frac{1}{p(\text{observed})} p(\text{hidden}, \text{observed})$$

- For model comparison we need the ratio of partition functions
- Often intractable to compute due to sum over many terms

5/30/2022

33

Estimating the Partition Function: Strategy 1

- Use importance sampling from proposal p_G with energy $G(z)$

$$\frac{Z_E}{Z_G} = \frac{\sum_{\mathbf{z}} \exp(-E(\mathbf{z}))}{\sum_{\mathbf{z}} \exp(-G(\mathbf{z}))} = \frac{\sum_{\mathbf{z}} \exp(-E(\mathbf{z}) + G(\mathbf{z})) \exp(-G(\mathbf{z}))}{\sum_{\mathbf{z}} \exp(-G(\mathbf{z}))}$$

$$= \mathbb{E}_{PG}[\exp(-E(\mathbf{z}) + G(\mathbf{z}))] \approx \frac{1}{L} \sum_{l=1}^{L} \exp\left(-E(\mathbf{z}^{(l)}) + G(\mathbf{z}^{(l)})\right)$$

- $z^{(l)}$ are sampled from p_G
- If Z_G is easy to compute we can estimate Z_E
- For this approach to work p_E needs to match p_G well.

5/30/2022

34

Estimating the Partition Function: Strategy 1

- Problem: how to come up with a p_G that matches p_E?
- Idea: we can use samples $z^{(l)}$ from p_E from a Markov chain:

$$p_G(\mathbf{z}) = \frac{1}{L} \sum_{l=1}^{L} T(\mathbf{z}^{(l)}, \mathbf{z})$$

where T gives the transition probabilities of the chain

- If Z_G is easy to compute we can estimate Z_E
- We now define $G(\mathbf{z}) = -\log p_G(\mathbf{z})$

$$\frac{Z_E}{Z_G} \approx \frac{1}{L} \sum_{l=1}^{L} \exp\left(-E(\mathbf{z}^{(l)}) - \log p_G(\mathbf{z}^{(l)})\right)$$

5/30/2022

35

Estimating the Partition Function: Chaining

- Partition function ratio estimation requires matching distributions
- Problematic when estimating absolute value of part. Function
  - Partition function Z_G needs to be evaluated exactly
  - Only simple, poor matching, distributions allow this
- Use set of distributions between simple p_1 and complex p_M

$$\frac{Z_M}{Z_1} = \frac{Z_2}{Z_1}\frac{Z_3}{Z_2}\cdots\frac{Z_M}{Z_{M-1}}$$

- The intermediate distributions interpolate from E_1 to E_M

$$E_\alpha(\mathbf{z}) = (1-\alpha)E_1(\mathbf{z}) + \alpha E_M(\mathbf{z})$$

- Now each term can be reasonably approximated

5/30/2022