

# Approximate Inference via Sampling

CS772A: Probabilistic Machine Learning

Piyush Rai

# Sampling for Approximate Inference

- Some typical tasks that we have to solve in probabilistic/fully-Bayesian inference

Posterior distribution  $\rightarrow$

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$$

Posterior predictive distribution  $\rightarrow$

$$p(\mathcal{D}^{new}|\mathcal{D}) = \int p(\mathcal{D}^{new}|\theta)p(\theta|\mathcal{D})d\theta = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^{new}|\theta)]$$

Needed for model selection (and in computing posterior too)  $\rightarrow$  Marginal likelihood  $\rightarrow$

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta)p(\theta|m)d\theta = \mathbb{E}_{p(\theta|m)}[p(\mathcal{D}|\theta)]$$

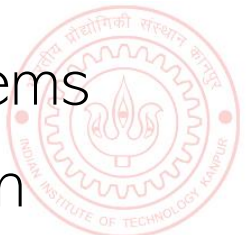
Needed in EM  $\rightarrow$  Expected complete data log-likelihood  $\rightarrow$

$$\text{Exp-CLL} = \int p(\mathbf{z}|\theta, \mathbf{x})p(\mathbf{x}, \mathbf{z}|\theta)d\mathbf{z} = \mathbb{E}_{p(\mathbf{z}|\theta, \mathbf{x})}[p(\mathbf{x}, \mathbf{z}|\theta)]$$

Needed in VI  $\rightarrow$  Evidence lower bound (ELBO)  $\rightarrow$

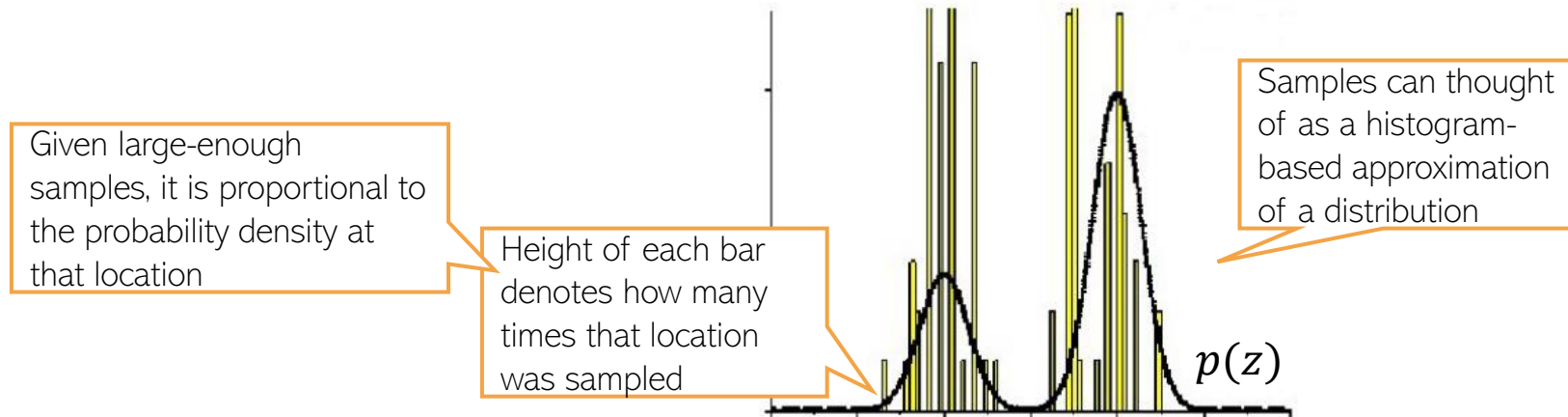
$$\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z})]$$

- Sampling methods provide a general way to (approximately) solve these problems
- More general than VI methods which only approximate the posterior distribution



# Approximating a Prob. Distribution using Samples <sup>3</sup>

- Can approximate any distribution using a set of **randomly drawn samples** from it



- The samples can also be used for computing expectations (Monte-Carlo averaging)
- Usually straightforward to generate samples if it is a simple/standard distribution
- The interesting bit: Even if the distribution is “difficult” (e.g., an intractable posterior), it is often possible to generate random samples from such a distribution, as we will see.



# The Empirical Distribution

- Sampling based approx. can be formally represented using an **empirical distribution**
- Given  $L$  points/samples  $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(L)}$ , empirical distr. defined by these is

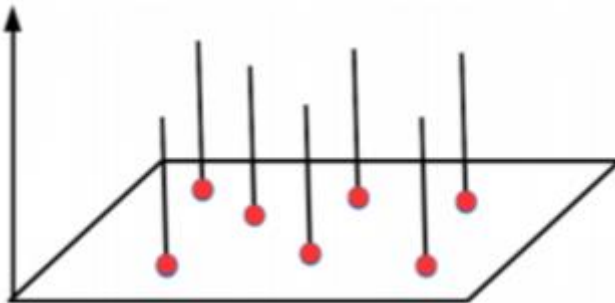
Dirac Distribution with finite support at  $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(L)}$

Weights sum to 1

Weight of point  $\mathbf{z}^{(\ell)}$

$$p_L(A) = \sum_{\ell=1}^L w_{\ell} \delta_{\mathbf{z}^{(\ell)}}(A)$$

Can think of  $A$  as being the area over which we want to evaluate the distribution



Dirac Distribution

$$\delta_{\mathbf{z}}(A) = \begin{cases} 0 & \text{if } \mathbf{z} \notin A \\ 1 & \text{if } \mathbf{z} \in A \end{cases}$$



# Sampling: Some Basic Methods

$$p(z) = q(x) \left| \frac{\partial x}{\partial z} \right|$$

5

Determinant of Jacobian

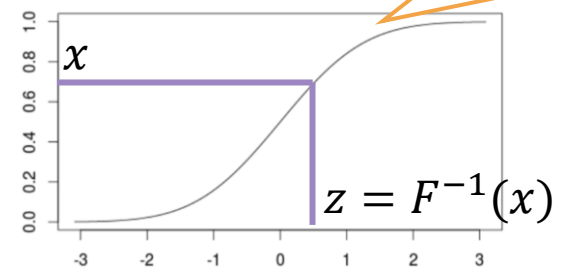
- Most of these basic methods are based on the idea of transformation
  - Generate a random sample  $x$  from a distribution  $q(x)$  which is easy to sample from
  - Apply a transformation on  $x$  to make it random sample  $z$  from a complex distr  $p(z)$

$F(z)$ : CDF of  $p(z)$

- Some popular examples of transformation methods

- Inverse CDF method

$$x \sim \text{Unif}(0, 1) \Rightarrow z = \text{Inv-CDF}_{p(z)}(x) \sim p(z)$$



- Reparametrization method

$$x \sim \mathcal{N}(0, 1) \Rightarrow z = \mu + \sigma x \sim \mathcal{N}(\mu, \sigma^2)$$

- Box-Mueller method: Given  $(x_1, x_2)$  from  $\text{Unif}(-1, +1)$ , generate  $(z_1, z_2)$  from  $\mathcal{N}(0, \mathbf{I}_2)$

$$z_1 = \sqrt{-2 \ln x_1} \cos(2\pi x_2), \quad z_2 = \sqrt{-2 \ln x_1} \sin(2\pi x_2)$$

- Transformation Methods are simple but have limitations

- Mostly limited to standard distributions and/or distributions with very few variables



# Rejection Sampling

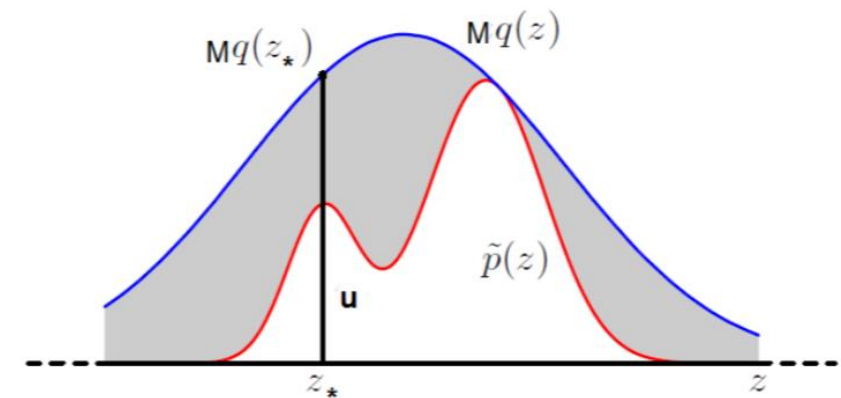
- Goal: Generate a random sample from a distribution of the form  $p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{Z_p}$ , assuming
  - We can only evaluate the value of numerator  $\tilde{p}(\mathbf{z})$  for any  $\mathbf{z}$
  - The denominator (normalization constant)  $Z_p$  is intractable and we don't know its value

Should have the same support as  $p(\mathbf{z})$

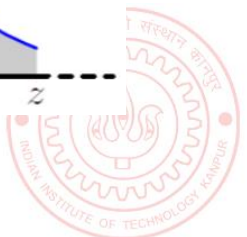
- Assume a **proposal distribution**  $q(\mathbf{z})$  we can generate samples from, and

$$Mq(\mathbf{z}) \geq \tilde{p}(\mathbf{z}) \quad \forall \mathbf{z} \quad (\text{where } M > 0 \text{ is some const.})$$

- Rejection Sampling then works as follows
  - Sample a random variable  $\mathbf{z}_*$  from  $q(\mathbf{z})$
  - Sampling a uniform r.v.  $u \sim \text{Unif}[0, Mq(\mathbf{z}_*)]$
  - If  $u \leq \tilde{p}(\mathbf{z}_*)$  then accept  $\mathbf{z}_*$ , otherwise reject it



- All accepted  $\mathbf{z}_*$ 's will be random samples from  $p(\mathbf{z})$ . Proof on next slide



# Rejection Sampling

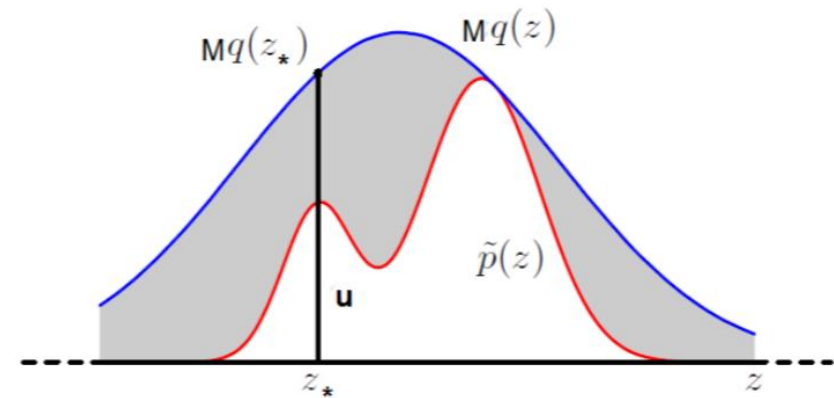
- Why  $z \sim q(z)$  + accept/reject rule is equivalent to  $z \sim p(z)$ ?
- Let's look at the pdf of the  $z$ 's that were accepted, i.e.,  $p(z|\text{accept})$

$$p(\text{accept}|z) = \int_0^{\tilde{p}(z)} \frac{1}{Mq(z)} du = \frac{\tilde{p}(z)}{Mq(z)}$$

$$p(z, \text{accept}) = q(z)p(\text{accept}|z) = \frac{\tilde{p}(z)}{M}$$

$$p(\text{accept}) = \int \frac{\tilde{p}(z)}{M} dz = \frac{Z_p}{M}$$

$$p(z|\text{accept}) = \frac{p(z, \text{accept})}{p(\text{accept})} = \frac{\tilde{p}(z)}{Z_p} = p(z)$$



# Computing Expectations via Monte Carlo Sampling<sup>8</sup>

- Often we are interested in computing expectations of the form

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

where  $f(\mathbf{z})$  is some function of the random variable  $\mathbf{z} \sim p(\mathbf{z})$

- A simple approx. scheme to compute the above expectation: [Monte Carlo integration](#)

- Generate  $L$  independent samples from  $p(\mathbf{z})$ :  $\{\mathbf{z}^{(\ell)}\}_{\ell=1}^L \sim p(\mathbf{z})$  Assuming we know how to sample from  $p(\mathbf{z})$
- Approximate the expectation by the following empirical average

$$\mathbb{E}[f] \approx \hat{f} = \frac{1}{L} \sum_{\ell=1}^L f(\mathbf{z}^{(\ell)})$$

- Since the samples are independent of each other, we can show the following (exercise)

Unbiased expectation

$$\mathbb{E}[\hat{f}] = \mathbb{E}[f]$$

$$\text{and } \text{var}[\hat{f}] = \frac{1}{L} \text{var}[f] = \frac{1}{L} \mathbb{E}[(f - \mathbb{E}[f])^2]$$

Variance in our estimate decreases as  $L$  increases



# Computing Expectations via Importance Sampling <sup>9</sup>

- How to compute Monte Carlo expec. if we don't know how to sample from  $p(\mathbf{z})$ ?
- One way is to use transformation methods or rejection sampling
- Another way is to use **Importance Sampling** (assuming  $p(\mathbf{z})$  can be evaluated at least)
  - Generate  $L$  indep samples from a **proposal**  $q(\mathbf{z})$  we know how sample from:  $\{\mathbf{z}^{(\ell)}\}_{\ell=1}^L \sim q(\mathbf{z})$
  - Now approximate the expectation as follows

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

- This is basically “weighted” Monte Carlo integration
  - $w^{(\ell)} = \frac{p(\mathbf{z}^{(\ell)})}{q(\mathbf{z}^{(\ell)})}$  denotes the **importance weight** of each sample  $\mathbf{z}^{(\ell)}$

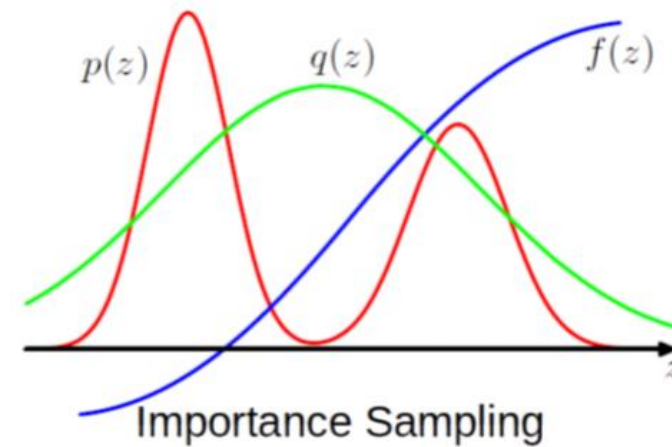
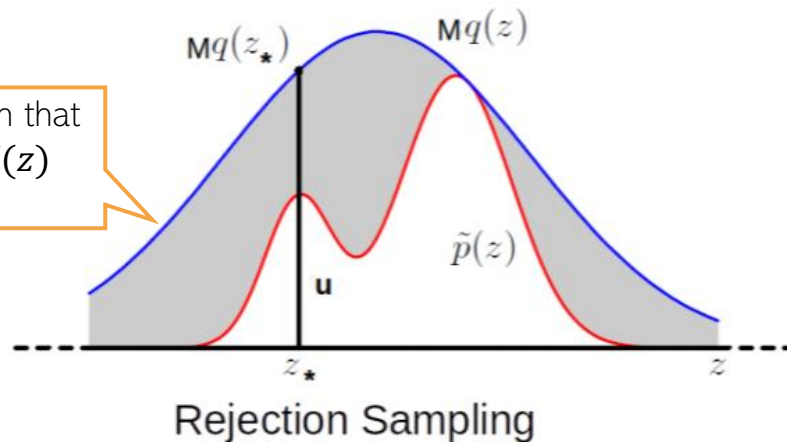
See PRML 11.1.4

- IS works even when we can only evaluate  $p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{Z_p}$  up to a prop. constant
- Note: Monte Carlo and Importance Sampling are NOT sampling methods!
  - These are only uses for computing expectations (approximately)



# Limitations of the Basic Methods

- Transformation based methods: Usually limited to drawing from standard distributions
- Rejection Sampling and Importance Sampling: Require good proposal distributions



$$\mathbb{E}[f] \approx \frac{1}{L} \sum_{\ell=1}^L f(z^{(\ell)}) \frac{p(z^{(\ell)})}{q(z^{(\ell)})}$$

Ideally, would like  $q(z)$  to give samples from where  $p(z)$  is large or  $f(z)p(z)$  is large

Difficult to guarantee so if  $z$  is high-dimensional

- In general, difficult to find good prop. distr. especially when  $z$  is high-dim
- More sophisticated sampling methods like MCMC work well in such high-dim spaces



# Markov Chain Monte Carlo (MCMC)

If the target is a posterior, it will be conditioned on data, i.e.,  $p(\mathbf{z}|\mathbf{x})$

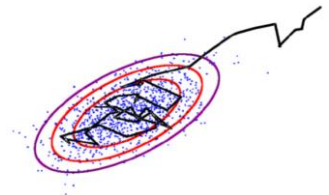
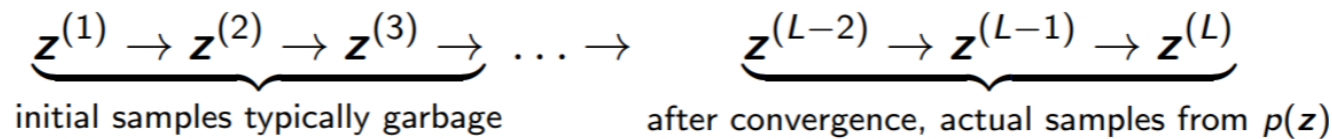
- Goal: Generate samples from some target distribution  $p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{Z_p}$

$\mathbf{z}$  usually is high-dim

- Assume we can evaluate  $p(\mathbf{z})$  at least up to a proportionality constant

Means we can at least evaluate  $\tilde{p}(\mathbf{z})$

- MCMC uses a **Markov Chain** which, when converged, starts giving samples from  $p(\mathbf{z})$



- Given current sample  $\mathbf{z}^{(\ell)}$  from the chain, MCMC generates the next sample  $\mathbf{z}^{(\ell+1)}$  as

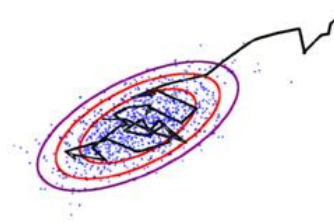
- Use a **proposal distribution**  $q(\mathbf{z}|\mathbf{z}^{(\ell)})$  to generate a candidate sample  $\mathbf{z}_*$
- **Accept/reject**  $\mathbf{z}_*$  as the next sample based on an **acceptance criterion** (will see later)
- If accepted, set  $\mathbf{z}^{(\ell+1)} = \mathbf{z}_*$ . If rejected, set  $\mathbf{z}^{(\ell+1)} = \mathbf{z}^{(\ell)}$

Should also have the same support as  $p(\mathbf{z})$

- Important: The proposal distribution  $q(\mathbf{z}|\mathbf{z}^{(\ell)})$  depends on the previous sample  $\mathbf{z}^{(\ell)}$



# MCMC: The Basic Scheme



- The chain run infinitely long (i.e., upon convergence) will give ONE sample from  $p(\mathbf{z})$
- But we usually require **several samples** to approximate  $p(\mathbf{z})$
- This is done as follows
  - Start the chain at an initial  $\mathbf{z}^{(0)}$
  - Using the proposal  $q(\mathbf{z}|\mathbf{z}^{(\ell)})$ , run the chain long enough, say  $T_1$  steps
  - Discard the first  $T_1 - 1$  samples (called “**burn-in**” **samples**) and take last sample  $\mathbf{z}^{(T_1)}$
  - Continue from  $\mathbf{z}^{(T_1)}$  up to  $T_2$  steps, discard intermediate samples, take last sample  $\mathbf{z}^{(T_2)}$ 
    - This discarding (called “**thinning**”) helps ensure that  $\mathbf{z}^{(T_1)}$  and  $\mathbf{z}^{(T_2)}$  are **uncorrelated**
  - Repeat the same for a total of  $S$  times
  - In the end, we now have  $S$  *approximately independent* samples from  $p(\mathbf{z})$
- Note: Good choices for  $T_1$  and  $T_i - T_{i-1}$  (thinning gap) are usually based on heuristics

MCMC is exact in theory but approximate in practice since we can't run the chain for infinitely long in practice



Thus we say that the samples are approximately from the target distribution

Will treat it as our first sample from  $p(\mathbf{z})$

Requirement for Monte Carlo approximation

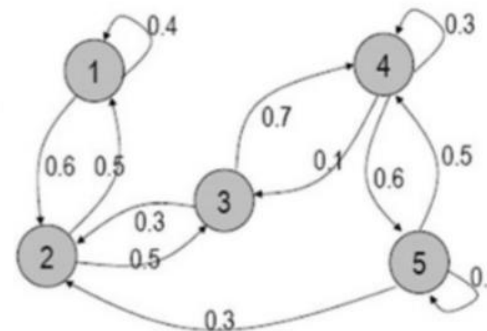


# MCMC: Some Basic Theory

- A first order Markov Chain assumes  $p(\mathbf{z}^{(\ell+1)} | \mathbf{z}^{(1)}, \dots, \mathbf{z}^{(\ell)}) = p(\mathbf{z}^{(\ell+1)} | \mathbf{z}^{(\ell)})$
- A 1st order Markov Chain  $\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, \dots, \mathbf{z}^{(L)}$  is a sequence of r.v.'s and is defined by
  - An initial state distribution  $p(\mathbf{z}^{(0)})$
  - A Transition Function (TF):  $T_\ell(\mathbf{z}^{(\ell)} \rightarrow \mathbf{z}^{(\ell+1)}) = p(\mathbf{z}^{(\ell+1)} | \mathbf{z}^{(\ell)})$
- TF is a distribution over the values of next state given the value of the current state
- Assuming a  $K$ -dim discrete state-space, TF will be  $K \times K$  probability table

Transition probabilities  
can be defined using a  
 $K \times K$  table if  $\mathbf{z}$  is a discrete  
r.v. with  $K$  possible values

	1	2	3	4	5
1	0.4	0.6	0.0	0.0	0.0
2	0.5	0.0	0.5	0.0	0.0
3	0.0	0.3	0.0	0.7	0.0
4	0.0	0.0	0.1	0.3	0.6
5	0.0	0.3	0.0	0.5	0.2



- Homogeneous Markov Chain: The TF is the same for all  $\ell$ , i.e.,  $T_\ell = T$

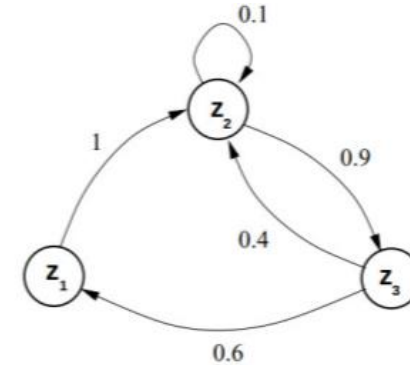


# MCMC: Some Basic Theory

- Consider the following Markov Chain with a  $K = 3$  discrete state-space

$$p(\mathbf{z}^{(0)}) = p(z_1^{(0)}, z_2^{(0)}, z_3^{(0)}) \\ = [0.5, 0.2, 0.3]$$

$$T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}$$



$$p(\mathbf{z}^{(1)}) = p(\mathbf{z}^{(0)}) \times T = [0.2, 0.6, 0.2] \quad (\text{rounded to single digit after decimal})$$

After doing it a few more  
(say some  $m$ ) times

Stationary/Invariant Distribution  
 $p(\mathbf{z})$  of this Markov Chain

$p(\mathbf{z})$  is multinoulli with  $\pi = [0.2, 0.4, 0.4]$

$$p(\mathbf{z}^{(0)}) \times T^m = [0.2, 0.4, 0.4] \quad (\text{rounded to single digit after decimal})$$

- $p(\mathbf{z})$  being Stationary means no matter what  $p(\mathbf{z}^{(0)})$  is, we will reach  $p(\mathbf{z})$
- A Markov Chain has a stationary distribution if  $T$  has the following properties
  - Irreducibility:  $T$ 's graph is connected (ensures reachability from anywhere to anywhere)
  - Aperiodicity:  $T$ 's graph has no cycles (ensures that the chain isn't trapped in cycles)



# MCMC: Some Basic Theory

- A Markov Chain with transition function  $T$  has stationary distribution  $p(\mathbf{z})$  if  $T$  satisfies

Known as the Detailed Balance condition

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

Here  $T(b|a)$  denotes the transition probability of going from state  $a$  to state  $b$

- Integrating out (or summing over) detailed balanced condition on both sides w.r.t.  $\mathbf{z}'$

Thus  $p(\mathbf{z})$  is the stationary distribution of this Markov Chain

$$p(\mathbf{z}) = \int p(\mathbf{z}')T(\mathbf{z}|\mathbf{z}')d\mathbf{z}'$$

- Thus a Markov Chain with detailed balance always converges to a stationary distribution
- Detailed Balance ensures reversibility
- Detailed balance is sufficient but not necessary condition for having a stationary distr.



# Coming Up Next

- MCMC algorithms
  - Metropolis Hastings (MH)
  - Gibbs sampling (special case of MH)





# Some MCMC Algorithms



# Metropolis-Hastings (MH) Sampling (1960)

- Suppose we wish to generate samples from a target distribution  $p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{Z_p}$
- Assume a suitable proposal distribution  $q(\mathbf{z}|\mathbf{z}^{(\tau)})$ , e.g.,  $\mathcal{N}(\mathbf{z}|\mathbf{z}^{(\tau)}, \sigma^2 \mathbf{I})$
- In each step, draw  $\mathbf{z}^*$  from  $q(\mathbf{z}|\mathbf{z}^{(\tau)})$  and accept  $\mathbf{z}^*$  with probability

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

Favors acceptance of  $\mathbf{z}^*$  if it is more probable than  $\mathbf{z}^{(\tau)}$  (under  $p(\mathbf{z})$ )

Also "unfavor"  $\mathbf{z}^*$  if its generation was favored too much by the proposal distribution

Favor acceptance of  $\mathbf{z}^*$  if it had very low chance of being generated by the proposal but it does have high probability  $\tilde{p}(\mathbf{z}^*)$  under the target

- Transition function of this Markov Chain:  $T(\mathbf{z}^*|\mathbf{z}^{(\tau)}) = A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) q(\mathbf{z}^*|\mathbf{z}^{(\tau)})$
- Exercise: Show that  $T(\mathbf{z}^*|\mathbf{z}^{(\tau)})$  satisfies the detailed balance property

$$p(\mathbf{z})T(\mathbf{z}^{(\tau)}|\mathbf{z}) = p(\mathbf{z}^{(\tau)})T(\mathbf{z}|\mathbf{z}^{(\tau)})$$



# The MH Sampling Algorithm

- Initialize  $\mathbf{z}^{(1)}$  randomly
- For  $\ell = 1, 2, \dots, L$ 
  - Sample  $\mathbf{z}^* \sim q(\mathbf{z}^* | \mathbf{z}^{(\ell)})$  and  $u \sim \text{Unif}(0, 1)$
  - Compute acceptance probability

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

- If  $A(\mathbf{z}^*, \mathbf{z}^{(\ell)}) > u$

$$\mathbf{z}^{(\ell+1)} = \mathbf{z}^*$$

Meaning accepting  $\mathbf{z}^*$  with probability  $A(\mathbf{z}^*, \mathbf{z}^{(\ell)})$

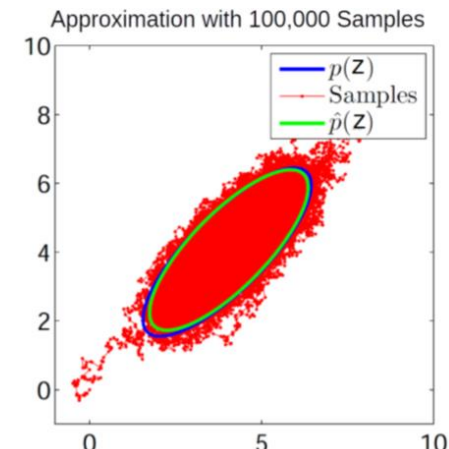
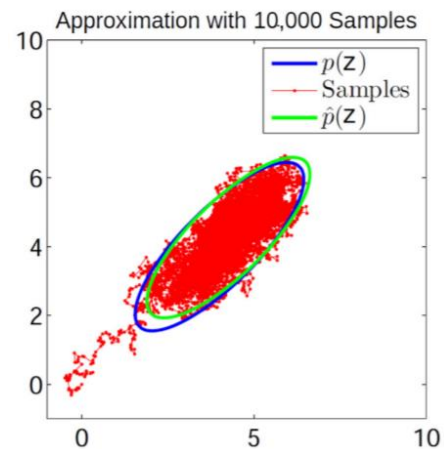
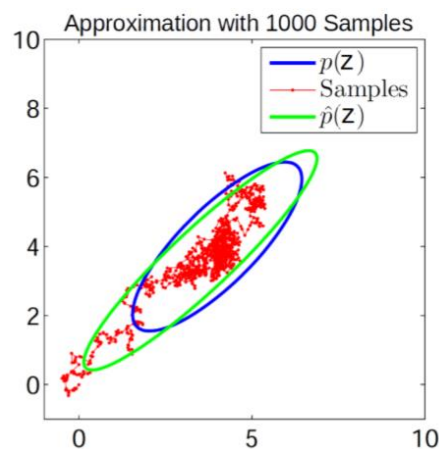
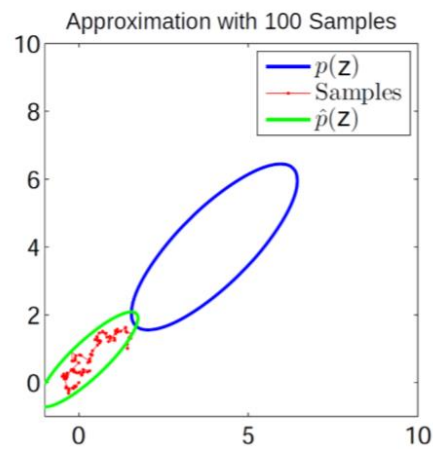
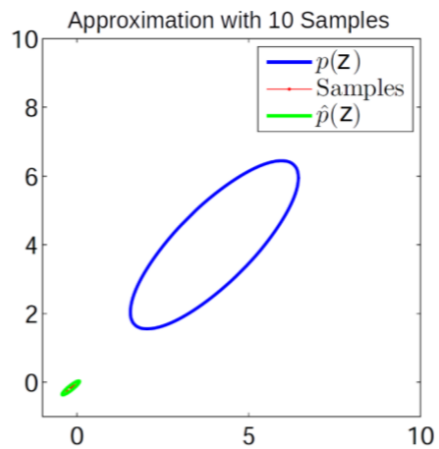
- Else

$$\mathbf{z}^{(\ell+1)} = \mathbf{z}^{(\ell)}$$



# MH Sampling in Action: A Toy Example..

- Target distribution  $p(\mathbf{z}) = \mathcal{N} \left( \begin{bmatrix} 4 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix} \right)$
- Proposal distribution  $q(\mathbf{z}^{(t)} | \mathbf{z}^{(t-1)}) = \mathcal{N} \left( \mathbf{z}^{(t-1)}, \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \right)$

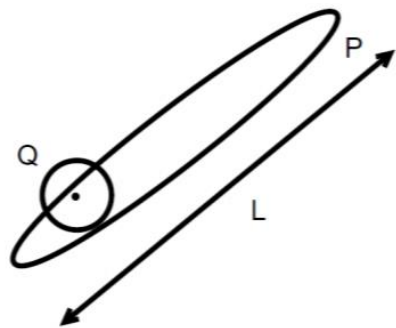


# MH Sampling: Some Comments

- If prop. distrib. is symmetric, we get [Metropolis Sampling](#) algo (Metropolis, 1953) with

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

- Some limitations of MH sampling
  - Can sometimes have very slow convergence (also known as slow “mixing”)



$$Q(\mathbf{z}|\mathbf{z}^{(\tau)}) = \mathcal{N}(\mathbf{z}|\mathbf{z}^{(\tau)}, \sigma^2 \mathbf{I})$$

$\sigma$  large  $\Rightarrow$  many rejections

$\sigma$  small  $\Rightarrow$  slow diffusion

$\sim \left(\frac{L}{\sigma}\right)^2$  iterations required for convergence

- Computing acceptance probability can be expensive\*, e.g., if  $p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{Z_p}$  is some target posterior then  $\tilde{p}(\mathbf{z})$  would require computing likelihood on all the data points (expensive)



# Gibbs Sampling (Geman & Geman, 1984)

- Goal: Sample from a joint distribution  $p(\mathbf{z})$  where  $\mathbf{z} = [z_1, z_2, \dots, z_M]$
- Suppose we can't sample from  $p(\mathbf{z})$  but can sample from each conditional  $p(z_i | \mathbf{z}_{-i})$ 
  - In Bayesian models, can be done easily if we have a locally conjugate model
- For Gibbs sampling, the proposal is the conditional distribution  $p(z_i | \mathbf{z}_{-i})$
- Gibbs sampling samples from these conditionals in a cyclic order
- Gibbs sampling is equivalent to MH sampling with acceptance prob. = 1

Hence no need to compute it

$$A(\mathbf{z}^*, \mathbf{z}) = \frac{p(\mathbf{z}^*)q(\mathbf{z}|\mathbf{z}^*)}{p(\mathbf{z})q(\mathbf{z}^*|\mathbf{z})} = \frac{p(z_i^*|\mathbf{z}_{-i}^*)p(\mathbf{z}_{-i}^*)p(z_i|\mathbf{z}_{-i}^*)}{p(z_i|\mathbf{z}_{-i})p(\mathbf{z}_{-i})p(z_i^*|\mathbf{z}_{-i})} = 1$$

where we use the fact that  $\mathbf{z}_{-i}^* = \mathbf{z}_{-i}$

Since only one component is changed at a time



# Gibbs Sampling: Sketch of the Algorithm

- $M$ : Total number of variables,  $T$ : number of Gibbs sampling iterations

1. Initialize  $\{z_i : i = 1, \dots, M\}$  Assuming  $\mathbf{z} = [z_1, z_2, \dots, z_M]$

2. For  $\tau = 1, \dots, T$ :

– Sample  $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}, \dots, z_M^{(\tau)})$ .

– Sample  $z_2^{(\tau+1)} \sim p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)}, \dots, z_M^{(\tau)})$ .

⋮

– Sample  $z_j^{(\tau+1)} \sim p(z_j | z_1^{(\tau+1)}, \dots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \dots, z_M^{(\tau)})$ .

⋮

– Sample  $z_M^{(\tau+1)} \sim p(z_M | z_1^{(\tau+1)}, z_2^{(\tau+1)}, \dots, z_{M-1}^{(\tau+1)})$ .

CP of each component of  $\mathbf{z}$  uses the most recent values (from this or the previous iteration) of all the other components

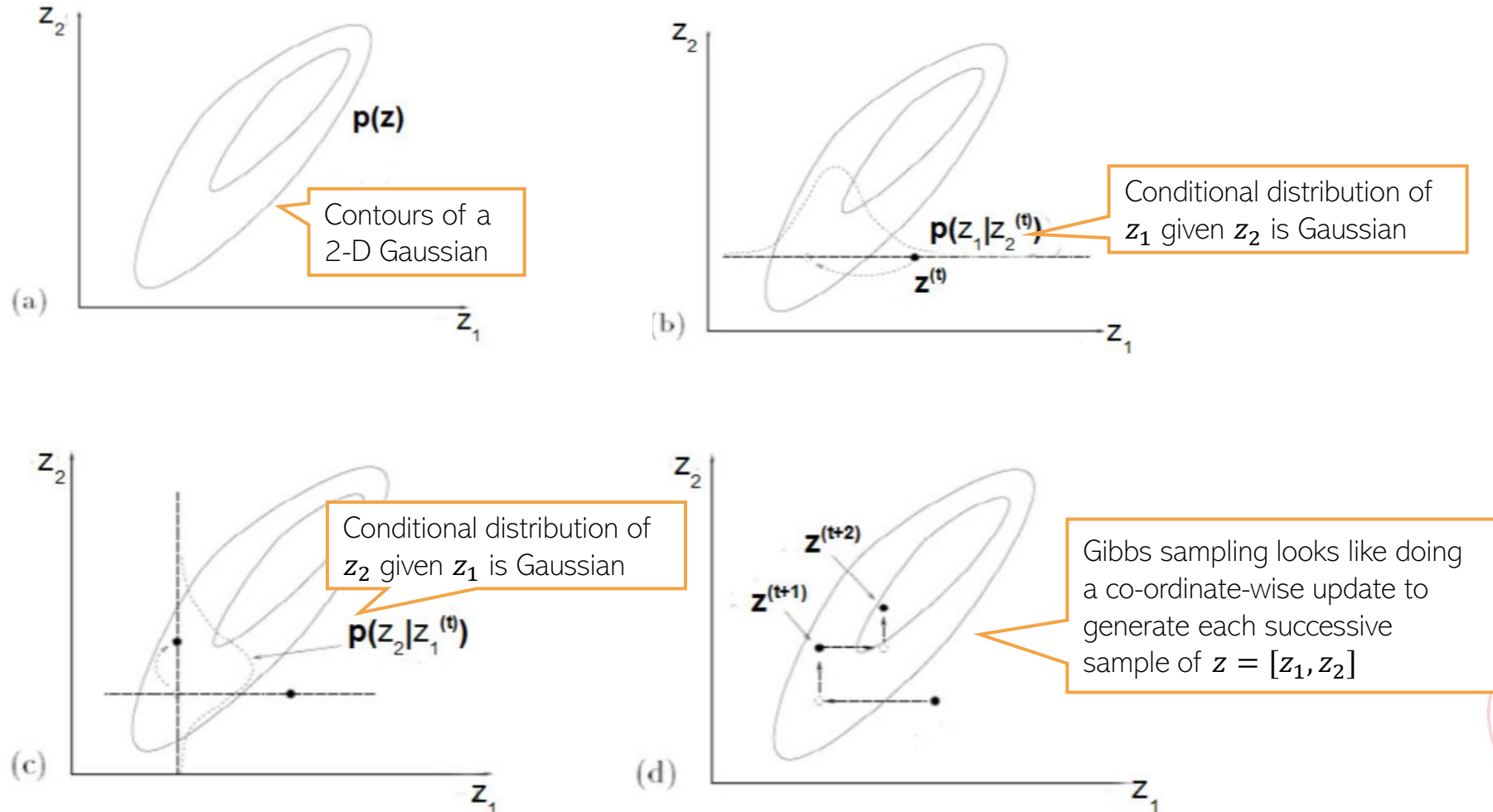
Each iteration will give us one sample  $\mathbf{z}^{(\tau)}$  of  $\mathbf{z} = [z_1, z_2, \dots, z_M]$

- Note: Order of updating the variables usually doesn't matter (but see "Scan Order in Gibbs Sampling: Models in Which it Matters and Bounds on How Much" from NIPS 2016)



# Gibbs Sampling: A Simple Example

- Can sample from a 2-D Gaussian using 1-D Gaussians





# Gibbs Sampling: Some Comments

- One of the most popular MCMC algorithms
- Very easy to derive and implement for locally conjugate models
- Many variations exist, e.g.,
  - **Blocked Gibbs**: sample more than one component jointly (sometimes possible)
  - **Rao-Blackwellized Gibbs**: Can collapse (i.e., integrate out) the unneeded components while sampling. Also called “collapsed” Gibbs sampling
  - **MH within Gibbs**: If CPs are not easy to sample distributions
- Instead of sampling from CPs, an alternative is to use the mode of the CPs
  - Called the “**Iterative Conditional Mode**” (ICM) algorithm
  - ICM doesn't give the posterior though – it's more like ALT-OPT to get (approx) MAP estimate



# Coming Up Next

- Using posterior's gradient info in sampling algorithms
- Online MCMC algorithms
- Recent advances in MCMC
- Some other practical issues (convergence etc)

