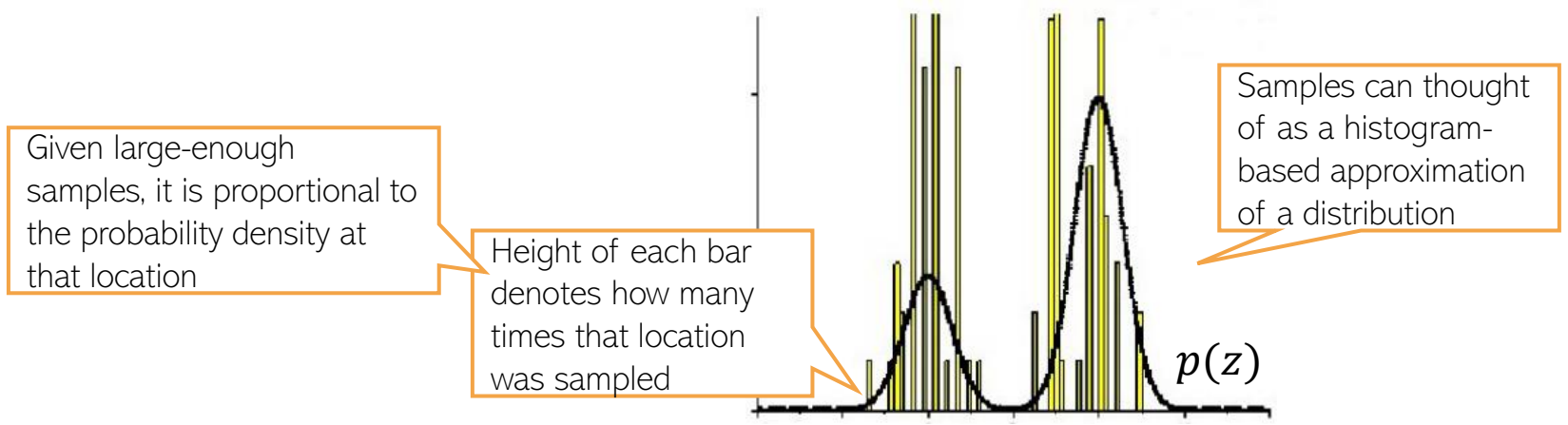
Sampling Methods to Approximate Distributions and Expectations

CS772A: Probabilistic Machine Learning

Piyush Rai

Approximating a Prob. Distribution using Samples

- 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 $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(L)}$, empirical distr. defined by these is

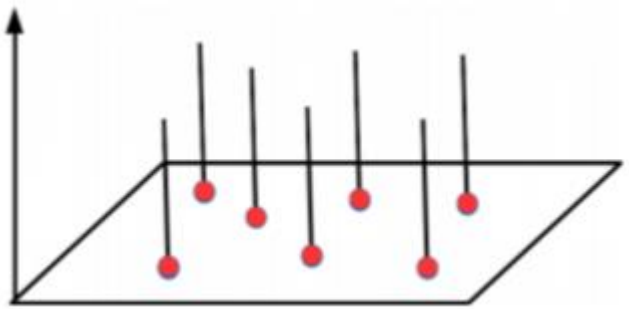
Dirac Distribution with finite support at $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(L)}$

Weights sum to 1

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

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

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



Dirac Distribution

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

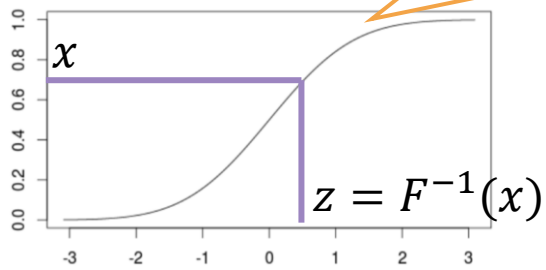
Sampling: Some Basic Methods

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

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 $\mathbf{Unif}(0,1)$, generate (z_1, z_2) from $\mathcal{N}(0, \mathbf{I}_2)$

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

- Transformation Methods are simple but have limitations
  - Mostly limited to standard distributions and/or distributions with very few variables

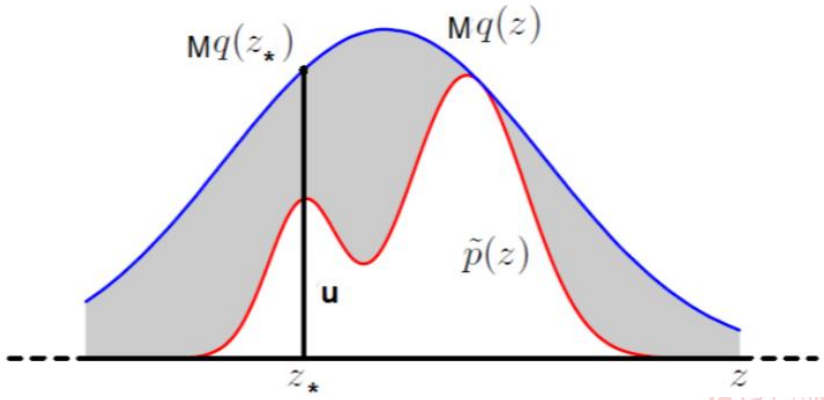
Rejection Sampling

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

Should have the same support as $p(z)$

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

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



- All accepted z_*'s will be random samples from $p(z)$. Proof on next slide

Rejection Sampling

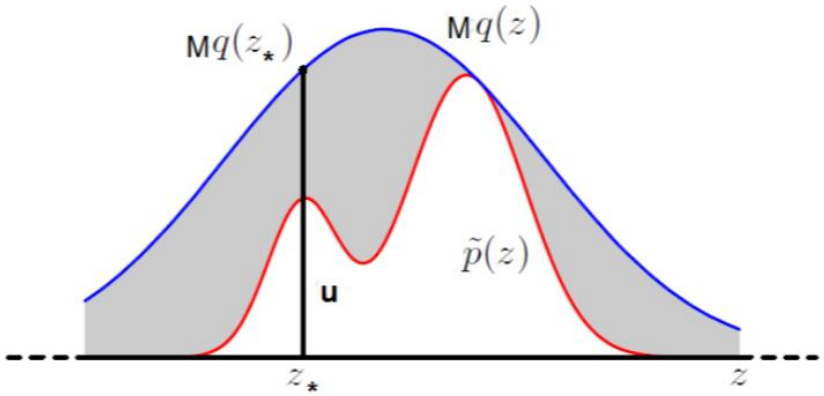
- Why $z \sim q(z)$ + accept/reject rule is equivalent to $z \sim p(z)$?
- Let's look at the pdf of the z's that were accepted, i.e., $p(z|\text{accept})$

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

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

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

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



Computing Expectations via Monte Carlo Sampling

- Often we are interested in computing expectations of the form

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

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

- A simple approx. scheme to compute the above expectation: Monte Carlo integration
  - Generate L independent samples from $p(z)$: $\{z^{(\ell)}\}_{\ell=1}^{L} \sim p(z)$

Assuming we know how to sample from $p(z)$

  - Approximate the expectation by the following empirical average

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

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

Unbiased expectation

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

Variance in our estimate decreases as L increases

Computing Expectations via Importance Sampling

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

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

- This is basically "weighted" Monte Carlo integration
  - $w^{(\ell)} = \frac{p(z^{(\ell)})}{q(z^{(\ell)})}$ denotes the importance weight of each sample $z^{(\ell)}$
- IS works even when we can only evaluate $p(z) = \frac{\tilde{p}(z)}{Z_p}$ up to a prop. constant

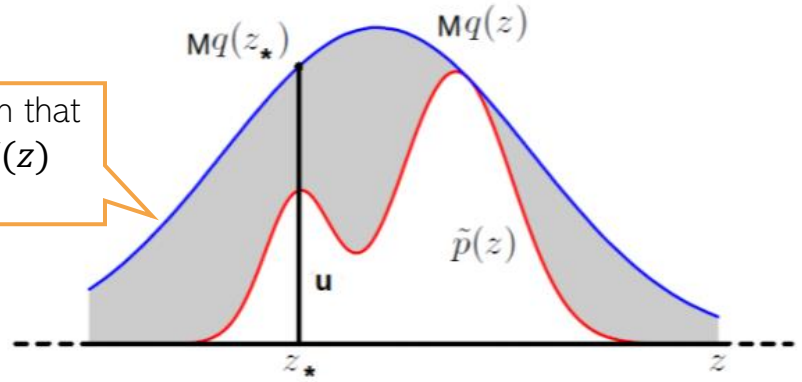
See PRML 11.1.4

- Note: Monte Carlo and Importance Sampling are NOT sampling methods!
  - These are only uses for computing expectations (approximately)

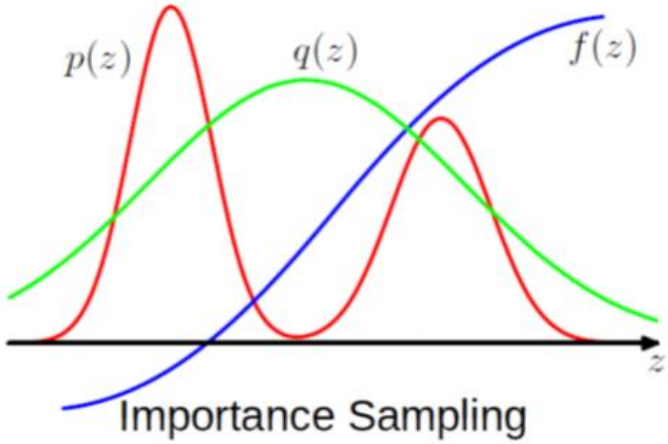
Limitations of the Basic Methods

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

$q(z)$ should be such that $Mq(z)$ envelopes $\tilde{p}(z)$ everywhere



Rejection Sampling



Importance Sampling

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

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)

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

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

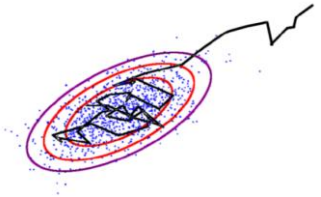
$\boldsymbol{z}$ usually is high-dim

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

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

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

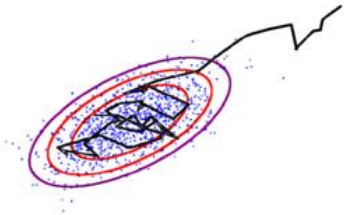
$$\underbrace{\boldsymbol{z}^{(1)} \rightarrow \boldsymbol{z}^{(2)} \rightarrow \boldsymbol{z}^{(3)} \rightarrow}_{\text{initial samples typically garbage}} \ldots \rightarrow \underbrace{\boldsymbol{z}^{(L-2)} \rightarrow \boldsymbol{z}^{(L-1)} \rightarrow \boldsymbol{z}^{(L)}}_{\text{after convergence, actual samples from } p(\boldsymbol{z})}$$



- Given current sample $\boldsymbol{z}^{(\ell)}$ from the chain, MCMC generates the next sample $\boldsymbol{z}^{(\ell+1)}$ as
  - Use a proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\ell)})$ to generate a candidate sample $\boldsymbol{z}_*$
  - Accept/reject $\boldsymbol{z}_*$ as the next sample based on an acceptance criterion (will see later)
  - If accepted, set $\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}_*$. If rejected, set $\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^{(\ell)}$
- Important: The proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\ell)})$ depends on the previous sample $\boldsymbol{z}^{(\ell)}$

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

MCMC: The Basic Scheme



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

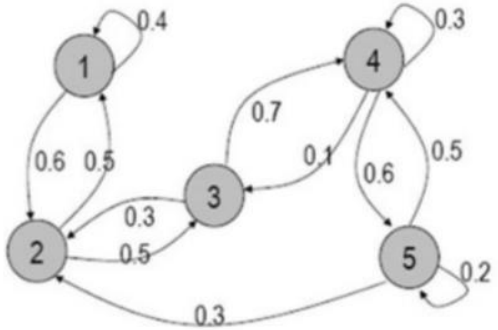
Requirement for Monte Carlo approximation

MCMC: Some Basic Theory

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

Transition probabilities can be defined using a *K*x*K* table if **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 ℓ , i.e., $T_\ell=T$