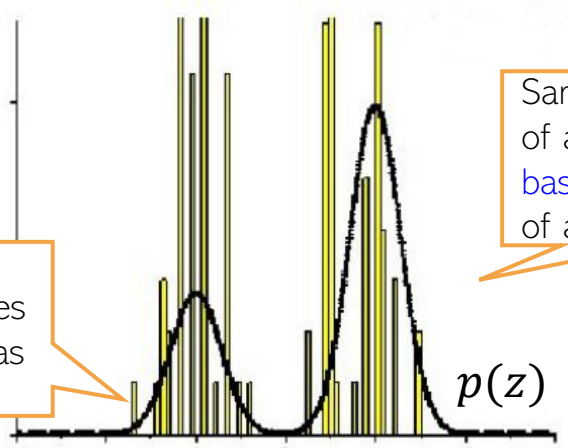
Approximating Distributions via Sampling

CS772A: Probabilistic Machine Learning

Piyush Rai

Approximating a Prob. Distribution using Samples

- A distribution can be approximated using a set of randomly drawn samples from it



Given large-enough samples, it is proportional to the probability density at that location

Height of each bar denotes how many times that location/interval was sampled

Samples can thought of as a histogram-based approximation of a distribution

- Using the S samples $\boldsymbol{Z}^{(1)}, \boldsymbol{Z}^{(2)}, \ldots, \boldsymbol{Z}^{(S)}$, our approx. $p(\boldsymbol{Z}) \approx \frac{1}{S} \sum_{s=1}^{S} \delta_{\boldsymbol{Z}^{(s)}}(\boldsymbol{Z})$

Dirac Distribution $\delta_z(\boldsymbol{A}) = 1$ if $\boldsymbol{z} \in \boldsymbol{A}$ and 0 otherwise

- 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.

Using samples to compute expectations/predictions

- Given S i.i.d. samples $\boldsymbol{Z}^{(1)}, \boldsymbol{Z}^{(2)}, \ldots, \boldsymbol{Z}^{(S)}$ from a distribution $p(\boldsymbol{Z})$

$$\mathbb{E}_{p(\boldsymbol{Z})}[f(\boldsymbol{Z})] = \int f(\boldsymbol{Z}) p(\boldsymbol{Z}) d\boldsymbol{Z} \approx \frac{1}{S} \sum_{s=1}^{S} f(\boldsymbol{Z}^{(s)})$$

- For Bayesian lin. reg., assuming $\boldsymbol{w}, \beta, \lambda$ to be unknown, PPD approximation will be

$$\int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta) p(\boldsymbol{w}, \beta, \lambda|\boldsymbol{X}, \boldsymbol{y}) d\boldsymbol{w} d\beta d\lambda \approx \frac{1}{S} \sum_{s=1}^{S} p\left(y_* \middle| \boldsymbol{x}_*, \boldsymbol{w}^{(s)}, \beta^{(s)}\right)$$

Joint posterior over all unknowns

Thus, in this case, the PPD is a sum of S Gaussians

Can also think of it as an ensemble consisting of S members

Sampling based approximation of PPD

Mean: $\mathbb{E}[\boldsymbol{w}^\top \boldsymbol{x}_*] \approx \frac{1}{S} \sum_{s=1}^{S} \boldsymbol{w}^{(s)\top} \boldsymbol{x}_*$

Variance: Exercise! Use definition of variance and use Monte-Carlo approximation

- Sampling based approx. for PPD of other models can also be obtained likewise

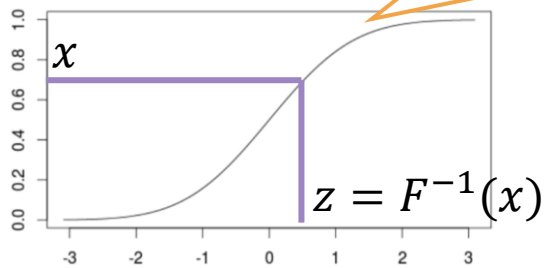
Sampling: Some Basic Methods

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

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

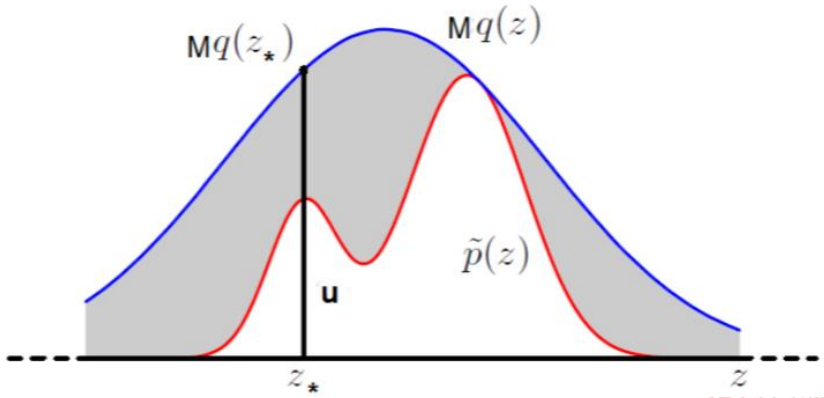
Rejection Sampling

- 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

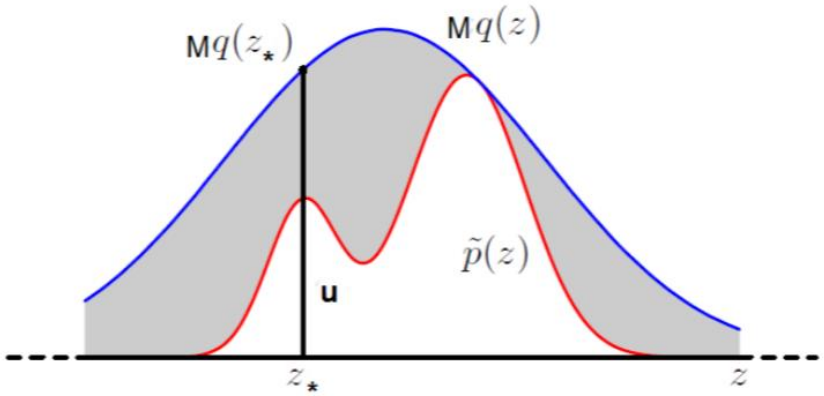
- Why $z \sim q(z)$ + accept/reject rule is equivalent to $z \sim p(z)$?
- Let's look at the pdf of the z's that were accepted, i.e., $p(z|\text{accept})$

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

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

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

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



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

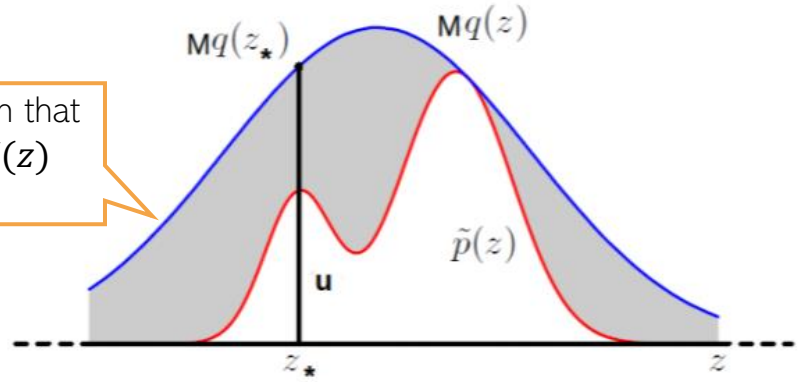
See PRML 11.1.4

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

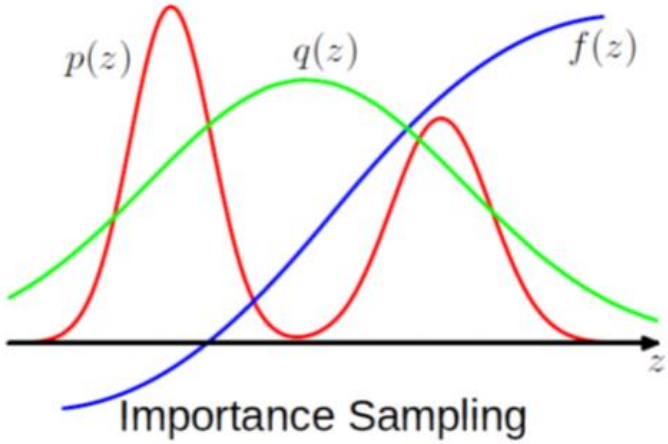
Limitations of the Basic Methods

- 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., $p(\boldsymbol{z}|\boldsymbol{x})$

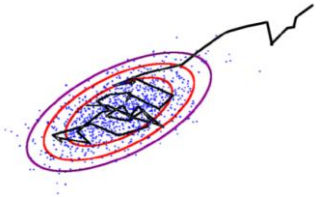
$\boldsymbol{z}$ usually is high-dim

- 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 $p(\boldsymbol{z})$

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

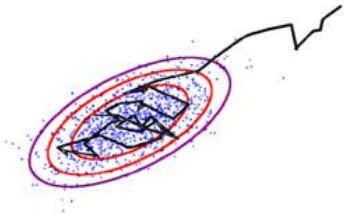


- Given current sample $\boldsymbol{z}^{(\ell)}$ from the chain, MCMC generates the next sample $\boldsymbol{z}^{(\ell+1)}$ as
  - Use a proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\ell)})$ to generate a candidate sample $\boldsymbol{z}_*$
  - Accept/reject $\boldsymbol{z}_*$ as the next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

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

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

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 $p(z)$

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

- 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 $z^{(T_1)}$

Will treat it as our first sample from $p(z)$

  - 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 $p(z)$

Requirement for Monte Carlo approximation

- Note: Good choices for T_1 and $T_i - T_{i-1}$ (thinning gap) are usually based on heuristics

Some MCMC Algorithms

Metropolis-Hastings (MH) Sampling (1960)

- Suppose we wish to generate samples from a target distribution $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$
- Assume a suitable proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\tau)})$, e.g., $\mathcal{N}(\boldsymbol{z}|\boldsymbol{z}^{(\tau)}, \sigma^2\boldsymbol{I})$
- Suppose the most recently accepted sample in the chain be $\boldsymbol{z}^{(\tau)}$
- Draw the next candidate $\boldsymbol{z}^*$ from $q(\boldsymbol{z}|\boldsymbol{z}^{(\tau)})$ and compute its acceptance probability

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

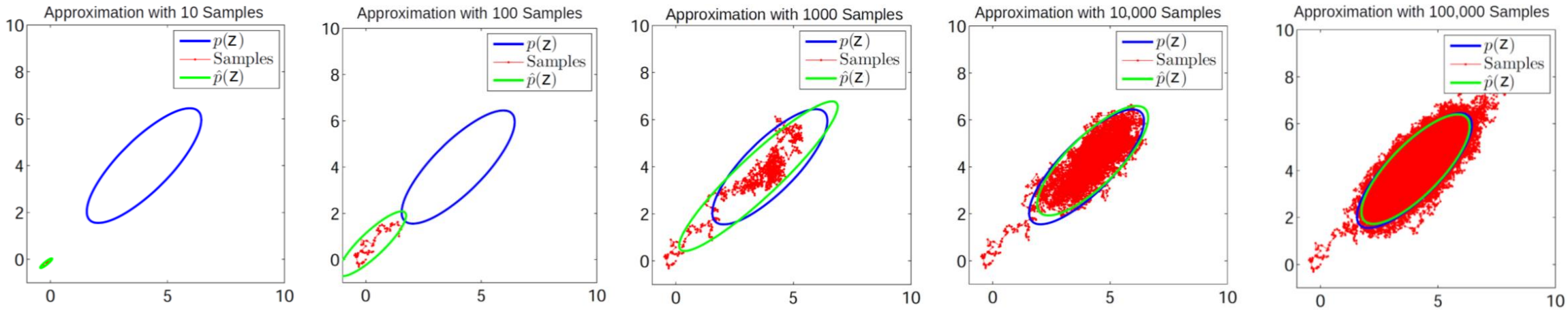
$$A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\boldsymbol{z}^*)q(\boldsymbol{z}^{(\tau)}|\boldsymbol{z}^*)}{\tilde{p}(\boldsymbol{z}^{(\tau)})q(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)})}\right)$$

Downweight the probability of acceptance of $\boldsymbol{z}^*$ if the proposal itself favors its generation (i.e., if $q(z^*|z^{(\tau)})$ is high), and upweight if it unfavors the generation

- Draw $u \sim \text{Unif}(0,1)$. If $A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)}) > u$, then accept $\boldsymbol{z}^*$, set $z^\tau = \boldsymbol{z}^*$; otherwise reject $\boldsymbol{z}^*$
  - This means we are accepting the sample $\boldsymbol{z}^*$ with probability $A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)})$
- This will generate a chain $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(T)}$ of samples

MH Sampling in Action: A Toy Example..

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

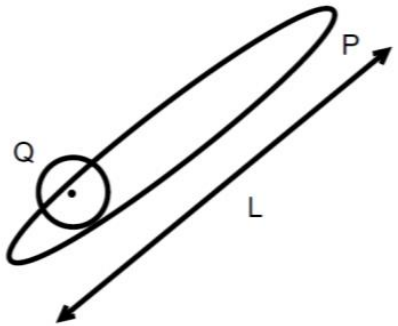


MH Sampling: Some Comments

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

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

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



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

σ large $\Rightarrow$ many rejections

σ small $\Rightarrow$ slow diffusion

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

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

*Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget (Korattikara et al, 2014), Firefly Monte Carlo: Exact MCMC with Subsets of Data {(Maclaurin and Adams, 2015)