# Sampling Methods

Slides adapted from CS772A, Piyush Rai, IITK

# Sampling for Approximate Inference

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

Posterior distribution

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

Posterior predictive distribution

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

Needed for model selection (and in computing posterior too)

Marginal likelihood

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

Needed in EM

Expected complete data log-likelihood

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

Needed in VI

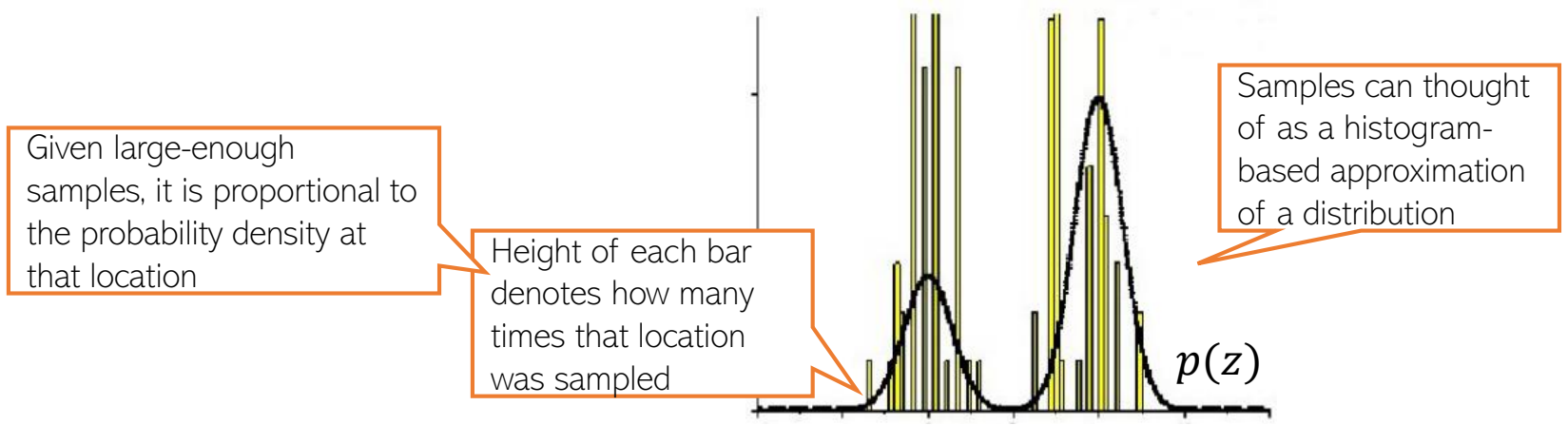
Evidence lower bound (ELBO)

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

- 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

- 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

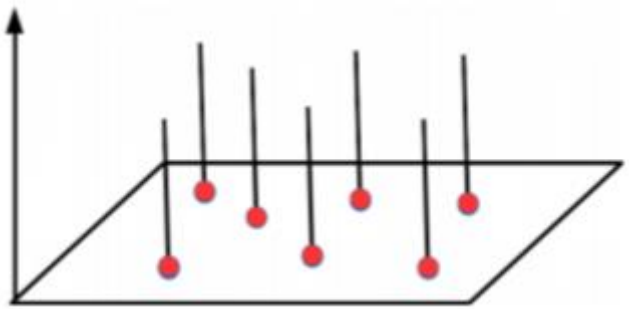
Dirac Distribution with finite support at $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(L)}$

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

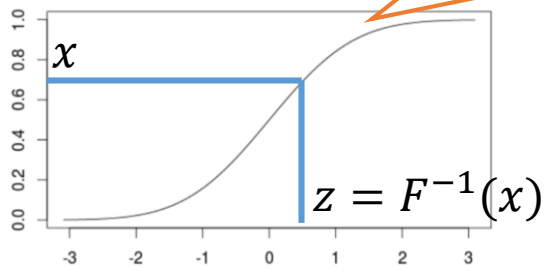


$$p(z) = q(x)\left|\frac{\partial x}{\partial z}\right|$$

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

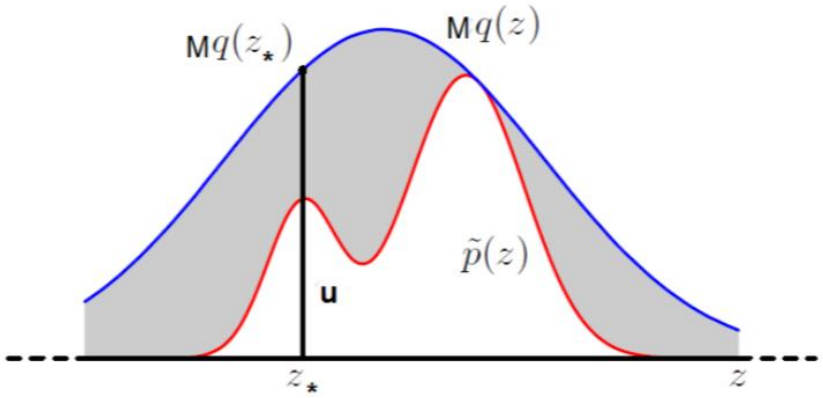
# Rejection Sampling

- 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. $u \sim \mathbf{Unif}[0, Mq(z_*)]$
  - 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

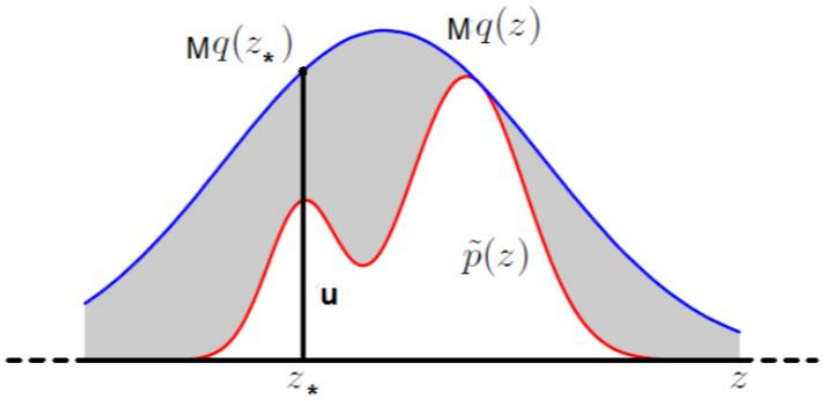
- Why $z \sim q(z)$ + accept/reject rule is equivalent to $z \sim p(z)$?
- Let's look at the pdf of the $z$'s that were accepted, i.e., $p(z|\text{accept})$

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

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

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

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



# 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
- 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

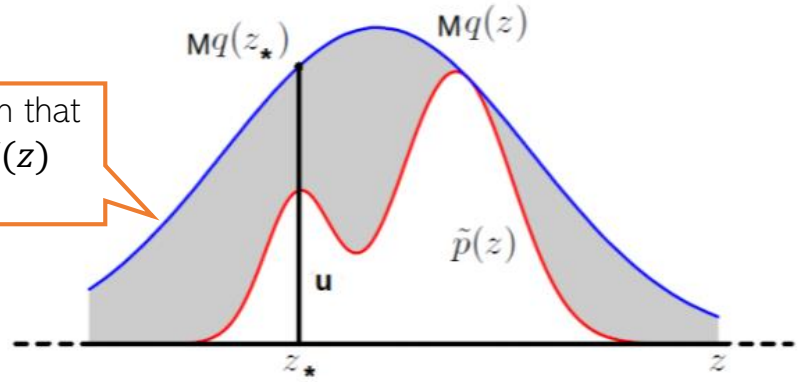
See PRML 11.1.4

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

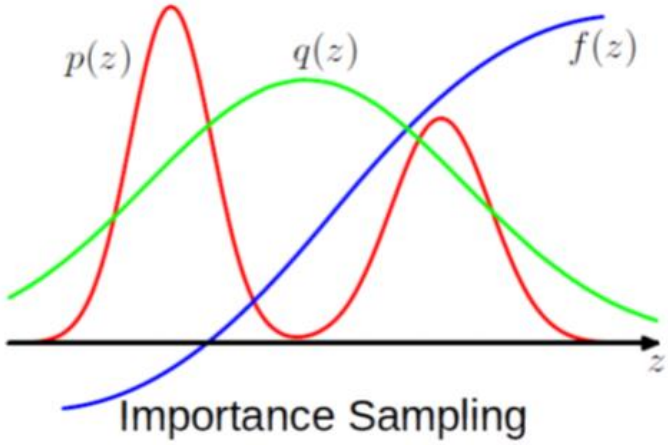
# Limitations of the Basic Methods

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

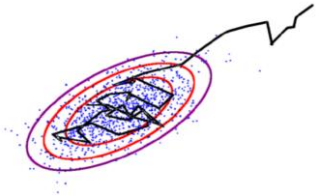
$\boldsymbol{z}$ usually is high-dim

- 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 $p(\boldsymbol{z})$

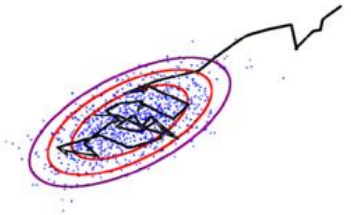
$$\underbrace{\boldsymbol{z}^{(1)} \to \boldsymbol{z}^{(2)} \to \boldsymbol{z}^{(3)} \to}_{\text{initial samples typically garbage}} \ldots \to \underbrace{\boldsymbol{z}^{(L-2)} \to \boldsymbol{z}^{(L-1)} \to \boldsymbol{z}^{(L)}}_{\text{after convergence, actual samples from } p(\boldsymbol{z})}$$



- 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 $\boldsymbol{z}^{(\ell)}$

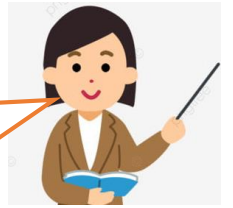
Should also have the same support as $p(\boldsymbol{z})$

# 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 $p(z)$

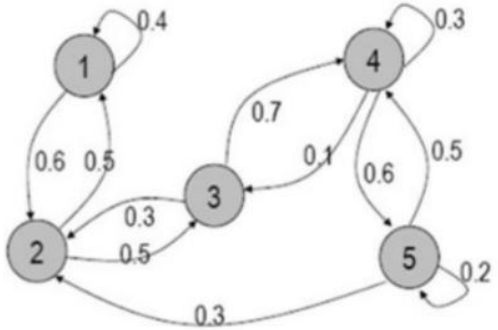
Requirement for Monte Carlo approximation

# 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 $\ell$ , i.e., $T_\ell=T$

# MCMC: Some Basic Theory

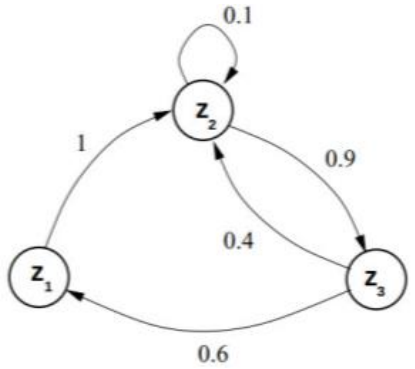
- Consider the following Markov Chain to sample a discrete r.v. $\boldsymbol{z}$ with 3 possible values

The initial state distribution for $\boldsymbol{z}$

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

Probabilities of the initial state taking each of the 3 possible values

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




Distribution of $\boldsymbol{z}$ after taking the first step

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

After doing it a few more (say some $\boldsymbol{m}$) times

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

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

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

- $p(\boldsymbol{z})$ being Stationary means no matter what $p(\boldsymbol{z}^{(0)})$ is, we will reach $p(\boldsymbol{z})$
- Such transition functions are desirable in MCMC

# MCMC: Some Basic Theory

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

Known as the Detailed Balance condition

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

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

- Detailed Balance ensures "reversibility"
- Integrating out (or summing over) detailed balanced condition on both sides w.r.t. $\boldsymbol{z}'$

Thus $p(z)$ is the stationary distribution of this Markov Chain

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

- Thus a Markov Chain with detailed balance always converges to a stationary distribution

- Detailed balance is sufficient but not necessary condition for having a stationary distr.

# 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., $\mathcal{N}(\boldsymbol{z}|\boldsymbol{z}^{(\tau)}, \sigma^2\boldsymbol{I})$
- In each step, draw $\boldsymbol{z}^*$ from $q(\boldsymbol{z}|\boldsymbol{z}^{(\tau)})$ and accept $\boldsymbol{z}^*$ with probability

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

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

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

- Transition function of this Markov Chain
  - $T(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)}) = A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)})q(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)})$ if state changed
  - $T(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)}) = q(\boldsymbol{z}^{(\tau)}|\boldsymbol{z}^{(\tau)}) + \sum_{\boldsymbol{z}^* \neq \boldsymbol{z}^{(\tau)}}(1 - A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)}))q(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)})$ otherwise

# The MH Sampling Algorithm

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

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

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

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

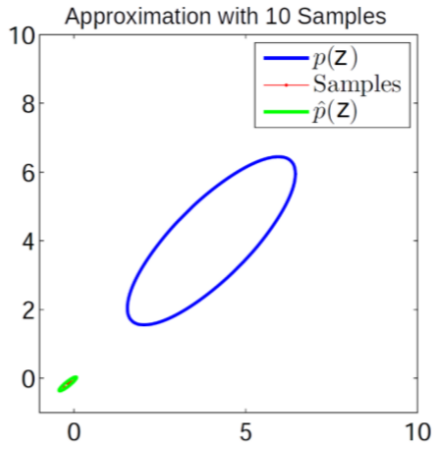
Meaning accepting $\boldsymbol{z}^*$ with probability $A(\boldsymbol{z}^*, \boldsymbol{z}^{(\ell)})$

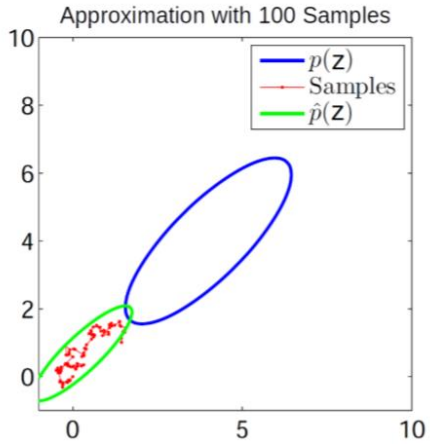
  - Else

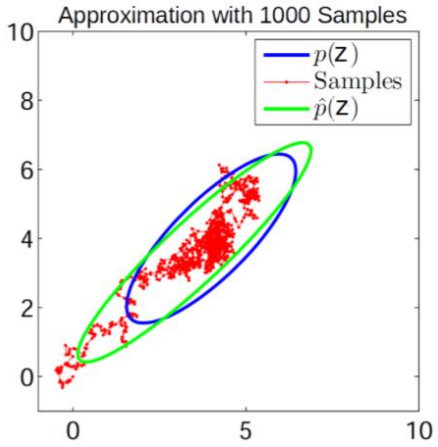
$$\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^{(\ell)}$$

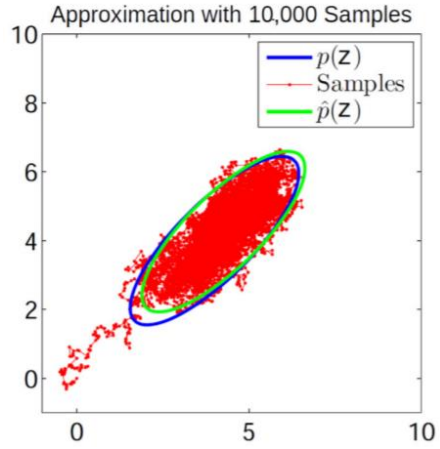
# MH Sampling in Action: A Toy Example..

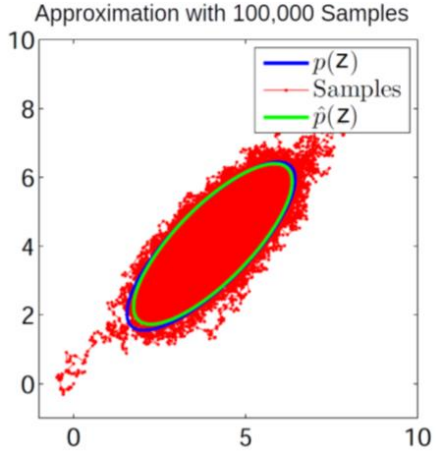
- Target distribution $p(\boldsymbol{z}) = \mathcal{N}\left(\begin{bmatrix}4\\4\end{bmatrix}, \begin{bmatrix}1 & 0.8\\0.8 & 1\end{bmatrix}\right)$
- Proposal distribution $q(\boldsymbol{z}^{(t)}|\boldsymbol{z}^{(t-1)}) = \mathcal{N}\left(\boldsymbol{z}^{(t-1)}, \begin{bmatrix}0.01 & 0\\0 & 0.01\end{bmatrix}\right)$



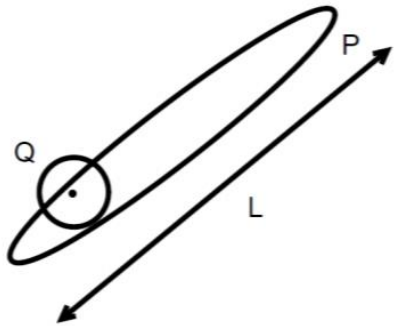



# MH Sampling: Some Comments

- 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")




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

$\sigma$ large ⇒ many rejections

$\sigma$ small ⇒ 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)

# Gibbs Sampling (Geman & Geman, 1984)

- Goal: Sample from a joint distribution $p(\boldsymbol{z})$ where $\boldsymbol{z} = [z_1, z_2, \ldots, z_M]$
- Suppose we can't sample from $p(\boldsymbol{z})$ but can sample from each conditional $p(z_i|\boldsymbol{z}_{-i})$
  - In Bayesian models, can be done easily if we have a locally conjugate model
- For Gibbs sampling, the proposal is the conditional distribution $p(z_i|\boldsymbol{z}_{-i})$
- 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

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

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

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 $\{z_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
   - Sample $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - Sample $z_2^{(\tau+1)} \sim p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - $\vdots$
   - Sample $z_j^{(\tau+1)} \sim p(z_j | z_1^{(\tau+1)}, \ldots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - $\vdots$
   - Sample $z_M^{(\tau+1)} \sim p(z_M | z_1^{(\tau+1)}, z_2^{(\tau+1)}, \ldots, z_{M-1}^{(\tau+1)})$.

Assuming $\boldsymbol{z} = [z_1, z_2, \ldots, z_M]$

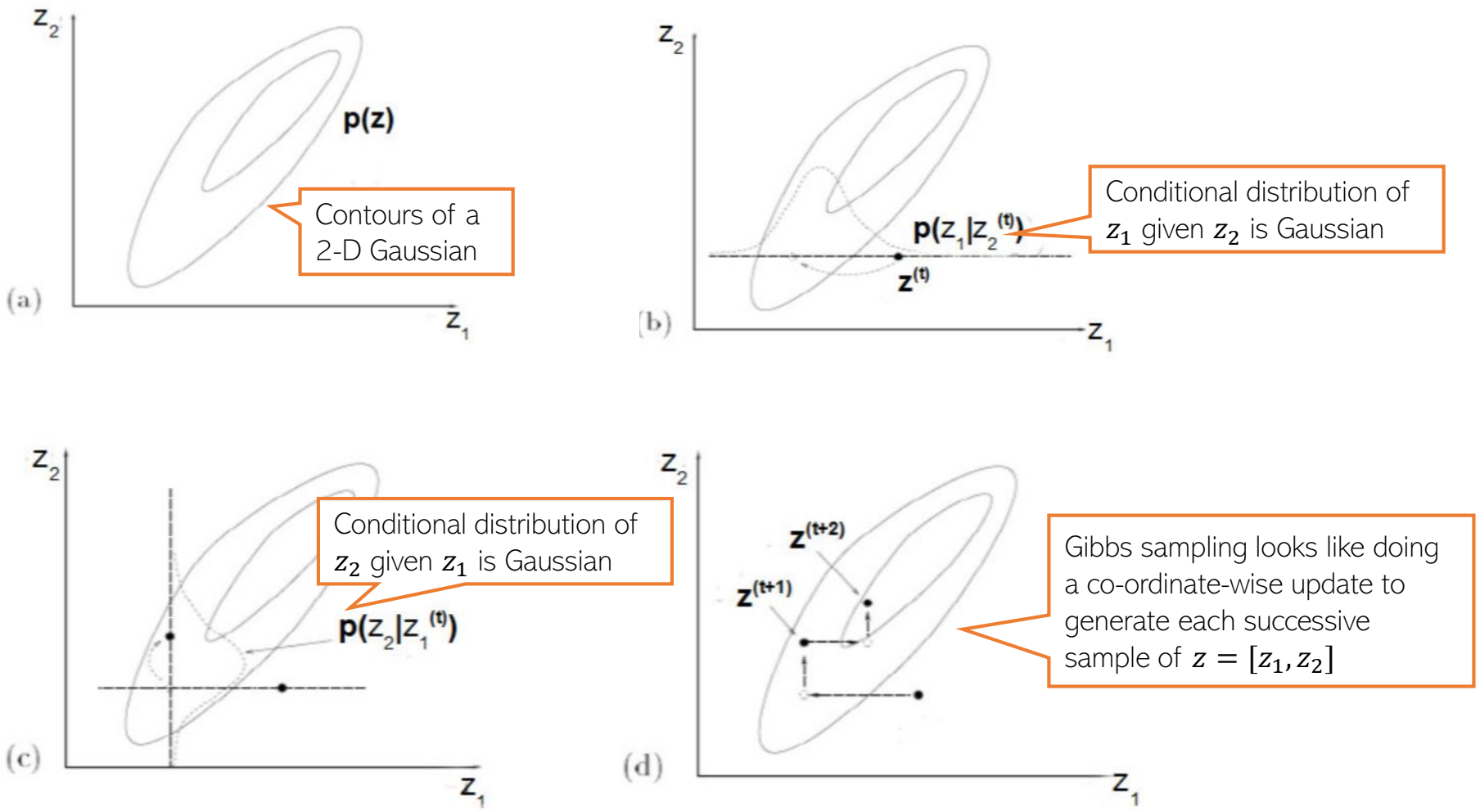
CP of each component of $\boldsymbol{z}$ uses the most recent values (from this or the previous iteration) of all the other components

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

- Note: Order of updating the variables usually doesn't matter

# Gibbs Sampling: A Simple Example

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



# Gibbs Sampling: Another Simple Example

- Bayesian linear regression: $p(y_n|\boldsymbol{x}_n, \boldsymbol{w}, \beta) = \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1})$, $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \lambda^{-1}I)$, $p(\lambda) = \mathbf{Gamma}(\lambda|a, b)$, $p(\beta) = \mathbf{Gamma}(\beta|c, d)$. Gibbs sampler for $p(w, \lambda, \beta|\boldsymbol{X}, \boldsymbol{y})$ will be
- Initialize $\lambda, \beta$ as $\lambda^{(0)}, \beta^{(0)}$. For iteration $t = 1,2, \ldots, T$
  - Generate a random sample of $\boldsymbol{w}$ by sampling from its CP as

$$\boldsymbol{w}^{(t)} \sim \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})$$

where

$$\boldsymbol{\Sigma}^{(t-1)} = \left(\beta^{(t-1)}\boldsymbol{X}^\top\boldsymbol{X} + \lambda^{(t-1)}\right)^{-1}$$

$$\boldsymbol{\mu}^{(t-1)} = \left(\boldsymbol{X}^\top\boldsymbol{X} + \frac{\lambda^{(t-1)}}{\beta^{(t-1)}}\right)^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

  - Generate a random sample of $\lambda$ by sampling from its CP as

$$\lambda^{(t)} \sim \mathbf{Gamma}\left(\lambda|a + \frac{D}{2}, b + \frac{\boldsymbol{w}^{(t)\top}\boldsymbol{w}^{(t)}}{2}\right)$$

  - Generate a random sample of $\beta$ by sampling from its CP as

$$\beta^{(t)} \sim \mathbf{Gamma}\left(\beta|c + \frac{N}{2}, d + \frac{\left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(t)}\right\|^2}{2}\right)$$

Note: Assuming these are post-burnin samples and thinning (if any) is also considered

- The posterior's approximation is the set of collected samples $\left\{\boldsymbol{w}^{(t)}, \lambda^{(t)}, \beta^{(t)}\right\}_{t=1}^{T}$

# Using MCMC samples to make predictions

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

- Any expectation that depends on $p(\boldsymbol{Z})$ can be approximated as

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

- For Bayesian lin. reg., assuming $\boldsymbol{w}, \beta, \lambda$ to be unknown, the PPD approx. 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: $\mathbb{E}[\boldsymbol{w}^{\mathsf{T}} \boldsymbol{x}_*] \approx \frac{1}{S} \sum_{s=1}^{S} \boldsymbol{w}^{(s)\mathsf{T}} \boldsymbol{x}_*$

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

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

# Approximate Inference: VI vs Sampling

- VI approximates a posterior distribution $p(\boldsymbol{Z}|\boldsymbol{X})$ by another distribution $q(\boldsymbol{Z}|\boldsymbol{\phi})$
- Sampling uses $\boldsymbol{S}$ samples $\boldsymbol{Z}^{(1)}, \boldsymbol{Z}^{(2)}, \ldots, \boldsymbol{Z}^{(S)}$ to approximate $p(\boldsymbol{Z}|\boldsymbol{X})$
- Sampling can be used within VI (ELBO approx using Monte-Carlo)
- In terms of "comparison" between VI and sampling, a few things to be noted
  - Convergence: VI only has local convergence, sampling (in theory) can give exact posterior
  - Storage: Sampling based approx needs to store all samples, VI only needs var. params $\boldsymbol{\phi}$
  - Prediction Cost: Sampling always requires Monte-Carlo averaging for posterior predictive; with VI, sometimes we can get closed form posterior predictive

**PPD if using sampling:** $$p(x_*|X) = \int p(x_*|Z)p(Z|X)dZ \approx \frac{1}{S}\sum_{s=1}^{S} p\left(x_*|Z^{(s)}\right)$$

**PPD if using VI:** $$p(x_*|X) = \int p(x_*|Z)p(Z|X)dZ \approx \int p(x_*|Z)q(Z|\phi)dZ$$

Closed form if integral is tractable (otherwise Monte Carlo avg still needed for PPD)

# Inference Methods: Summary

- MLE/MAP: Straightforward for differentiable models (can even use automatic diff.)
- Conjugate models with one "main" parameter: Straightforward posterior updates
- MLE-II/MAP-II: Often useful for estimating the hyperparameters
- EM: If we want to do MLE/MAP for models with latent variables
  - Very general algorithm, can also be made online
  - Used when we want point estimates for some unknowns and posterior over others
  - Can use it for hyperparameter estimation as well
  - Often better than using direct gradient methods
- VI and sampling methods can be used to get full posterior for complex models
  - Quite easy if we have local conjugacy (VI has closed form updates, Gibbs sampler is easy to derive)
  - In other cases, we have general VI with Monte-Carlo gradients, MH sampling
  - MCMC can also make use of gradient info (LD/SGLD)

# References

- Chapter 11-12, Kevin Murphy, Probabilistic Machine Learning: Advanced Topics