

10707

Deep Learning

Russ Salakhutdinov

Machine Learning Department

rsalakhu@cs.cmu.edu

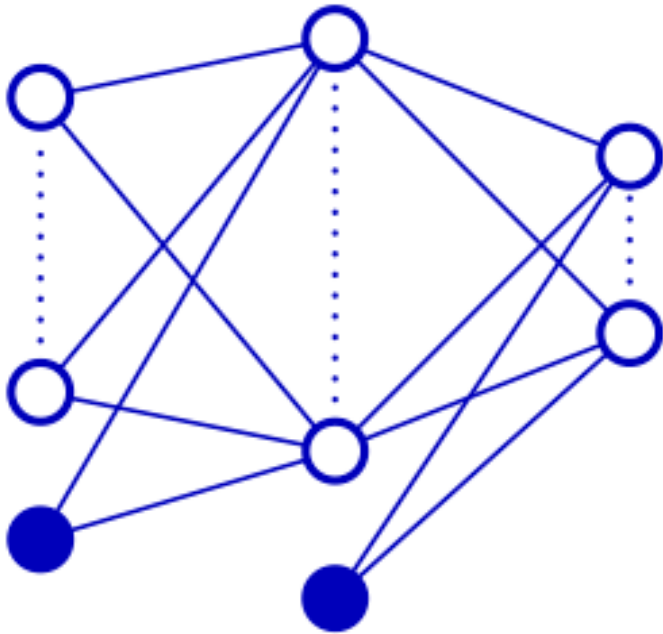
Markov Chain Monte Carlo

Approximate Inference

- When using probabilistic graphical models, we will be interested in evaluating the **posterior distribution** $p(\mathbf{Z}|\mathbf{X})$ of the latent variables \mathbf{Z} given the observed data \mathbf{X} .
- For example, in the EM algorithm, we need to evaluate the **expectation of the complete-data log-likelihood** with respect to the **posterior distribution** over the latent variables.
- For more complex models, it may be infeasible to evaluate the posterior distribution, or compute expectations with respect to this distribution.
- Last class we looked at **variational approximations**, including mean-field. .
- We now consider **sampling-based methods**, known as Monte Carlo techniques.

Bayesian Neural Networks

- Example: consider **Bayesian neural nets**, that often give state-of-the-art results for a range of regression problems.
- **Regression problem**: We are given a set of i.i.d. observations $\mathbf{X} = \{x^1, \dots, x^N\}$ with corresponding targets $\mathbf{T} = \{t^1, \dots, t^N\}$.



- Likelihood:

$$p(\mathbf{T}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^N \mathcal{N}(t^n | y(\mathbf{x}^n, \mathbf{w}), \beta^2)$$

- The mean is given by the **output of the neural network**:

$$y_k(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^M w_{kj}^{(2)} \sigma \left(\sum_{i=1}^D w_{ji}^{(1)} x_i \right)$$

where $\sigma(x)$ is the sigmoid function.

- We place **Gaussian prior over model parameters**: $p(\mathbf{w}) = \mathcal{N}(0, \alpha I)$.

Bayesian Neural Networks

- We therefore have:

Nonlinear function of inputs.

- Likelihood:

$$p(\mathbf{T}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^N \mathcal{N}(t^n | y(\mathbf{x}^n, \mathbf{w}), \beta^2)$$

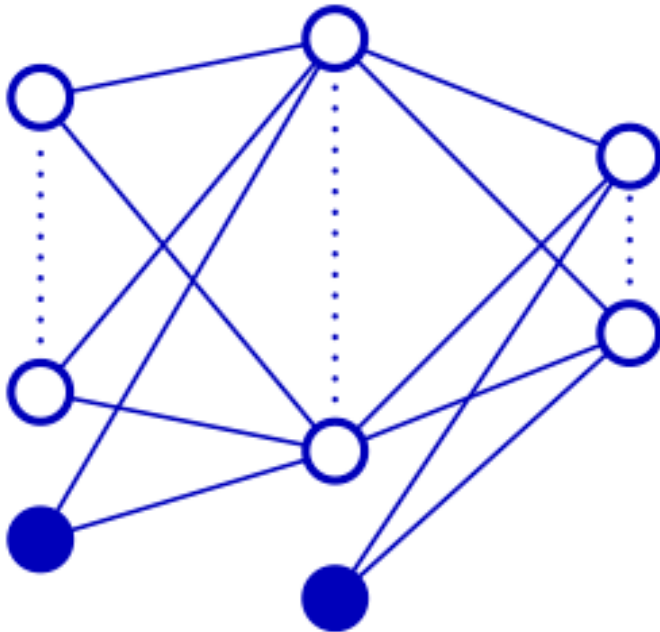
- Gaussian prior over parameters:

$$p(\mathbf{w}) = \mathcal{N}(0, \alpha I).$$

- The posterior is analytically intractable:

$$p(\mathbf{w}|\mathbf{T}, \mathbf{X}) = \frac{p(\mathbf{T}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{T}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}}$$

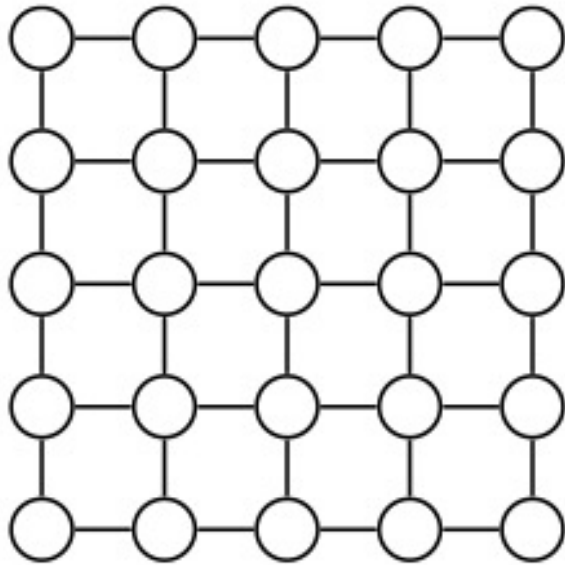
Cannot analytically compute normalizing constant.



- We need the posterior to compute predictive distribution for t given a new input x .

Undirected Graphical Models

- Let \mathbf{x} be a binary random vector with $x_i \in \{-1, 1\}$:



$$P_{\theta}(\mathbf{x}) = \frac{1}{Z(\theta)} \exp \left(\sum_{ij \in E} x_i x_j \theta_{ij} + \sum_{i \in V} x_i \theta_i \right)$$

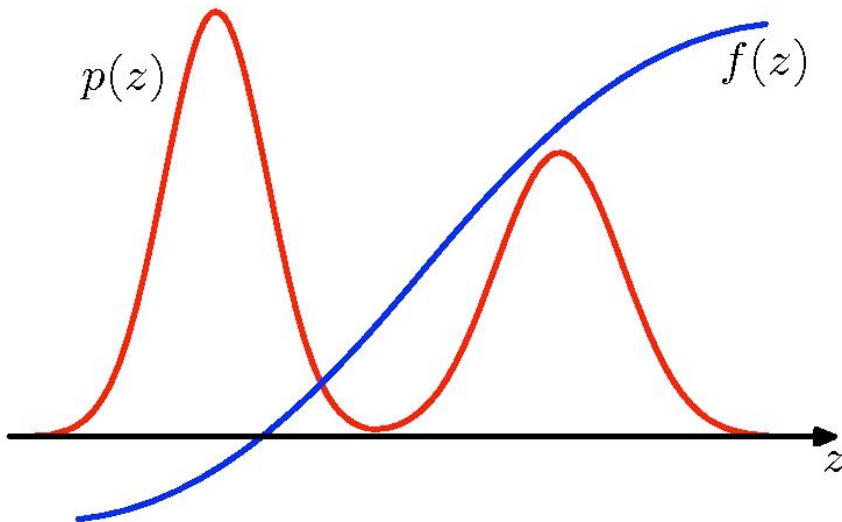
where $Z(\theta)$ is a **normalizing constant** (also known as partition function):

$$Z(\theta) = \sum_{\mathbf{x}} \exp \left(\sum_{ij \in E} x_i x_j \theta_{ij} + \sum_{i \in V} x_i \theta_i \right).$$

- If \mathbf{x} is 100-dimensional, we need to sum over 2^{100} terms.
- The sum might decompose, which would be the case for the **tree structured graphical models** (or models with low tree-width). Otherwise, we need to approximate.

Notation

- For most situations, we will be interested in **evaluating expectations** (for example in order to make predictions):



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

where the integral will be replaced with summation in case of discrete variables.

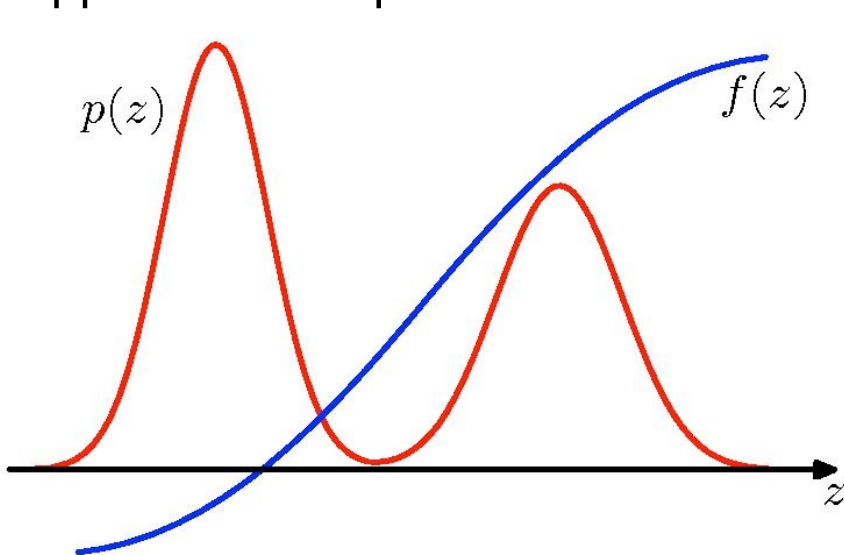
- We will make use of the following notation: $p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{\mathcal{Z}}$.
- We can evaluate $\tilde{p}(\mathbf{z})$ **pointwise** but **cannot evaluate** \mathcal{Z} .

- **Posterior distribution:** $p(\theta|\mathcal{D}) = \frac{1}{p(\mathcal{D})}p(\mathcal{D}|\theta)p(\theta)$.

- **Markov Random Fields:** $p(\mathbf{x}) = \frac{1}{\mathcal{Z}} \exp(-E(\mathbf{x}))$.

Simple Monte Carlo

- **General Idea:** Draw independent samples $\{z^1, \dots, z^n\}$ from distribution $p(z)$ to approximate expectation:



$$\mathbb{E}[f] = \int f(z)p(z)dz \approx \frac{1}{N} \sum_{n=1}^N f(z^n) = \hat{f}.$$

Note that:

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

so the estimator has correct mean (**unbiased**).

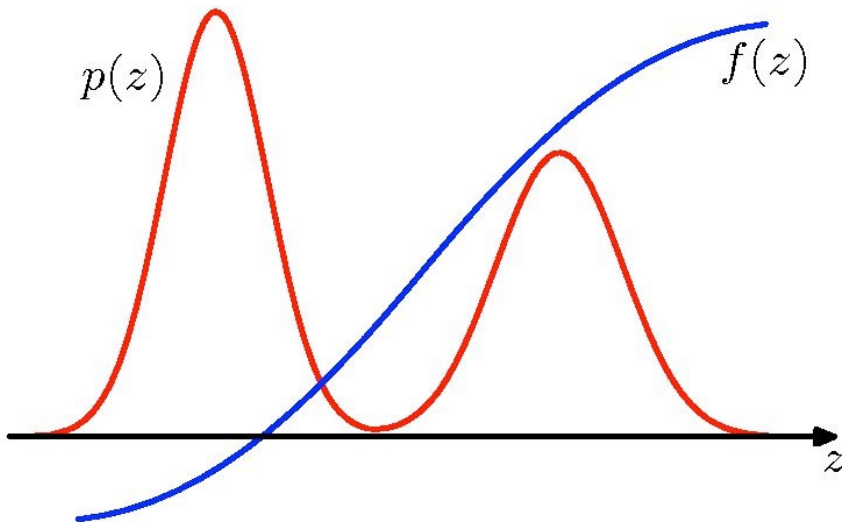
- The **variance**:
$$\text{var}[\hat{f}] = \frac{1}{N} \mathbb{E}[(f - \mathbb{E}[f])^2].$$

- Variance decreases as $1/N$.

- **Remark:** The accuracy of the estimator **does not depend on dimensionality** of z .

Simple Monte Carlo

- High accuracy may be achieved with a **small number N of independent samples** from distribution $p(z)$.



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

- **Problem 1:** we may **not be able to draw independent samples**.

- **Problem 2:** if $f(z)$ is large in regions where $p(z)$ is small (and vice versa), then the expectations may be dominated by **regions of small probability**. Need larger sample size.

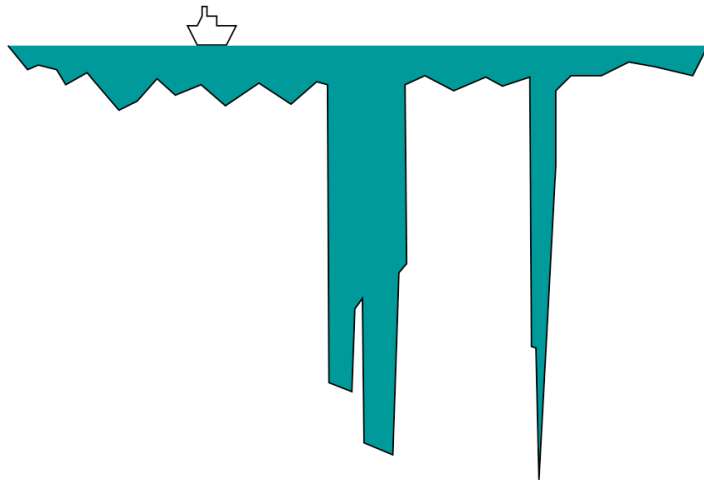
Simple Monte Carlo

- In general:

$$\mathbb{E}[f] = \int f(z)p(z)dz \approx \frac{1}{N} \sum_{n=1}^N f(z^n), \quad z^n \sim p(z).$$

- Predictive distribution:

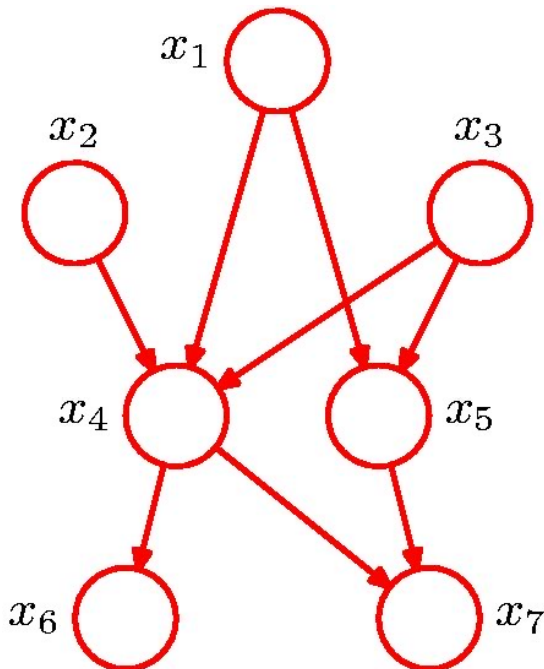
$$p(x^*|\mathcal{D}) = \int p(x^*|\theta, \mathcal{D})p(\theta|\mathcal{D})d\theta$$
$$\approx \frac{1}{N} \sum_{n=1}^N p(x^*|\theta^n), \quad \theta^n \sim p(\theta|\mathcal{D}).$$



- **Problem:** It is hard to draw exact samples from $p(z)$.

Directed Graphical Models

- For many distributions, the joint distribution can be conveniently specified in terms of a **graphical model**.



- For directed graphs **with no observed variables**, sampling from the joint is simple:

$$p(\mathbf{x}) = \prod_{k=1}^K p(x_k | \text{pa}_k)$$

$$\hat{x}_1 \sim p(x_1)$$

$$\hat{x}_2 \sim p(x_2)$$

$$\hat{x}_3 \sim p(x_3)$$

$$\hat{x}_4 \sim p(x_4 | \hat{x}_1, \hat{x}_2, \hat{x}_3)$$

$$\hat{x}_5 \sim p(x_5 | \hat{x}_1, \hat{x}_3)$$

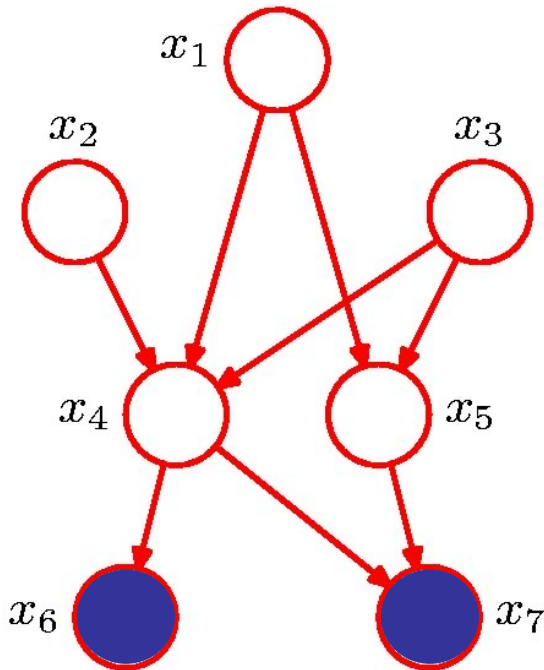
The parent variables
are set to their
sampled values



- After one pass through the graph, we obtain a sample from the joint.

Directed Graphical Models

- Consider the case when **some of the nodes are observed**.



- Naive idea: Sample from the joint.

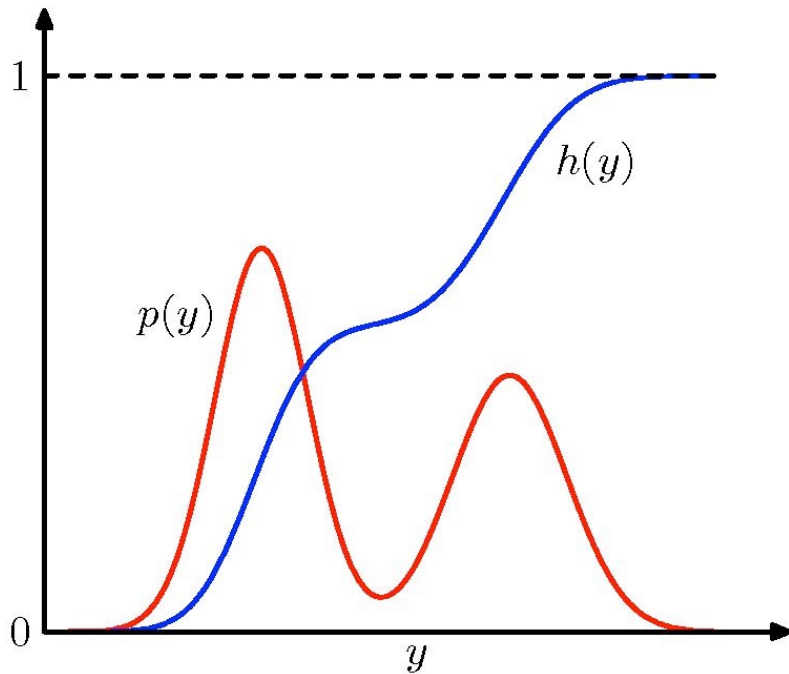
$$p(\mathbf{x}) = \prod_{k=1}^K p(x_k | \text{pa}_k)$$

- If the sampled **values agree with the observed values**, we retain the sample.
- Otherwise, we **disregard the whole sample**.

- The algorithm samples correctly from the posterior.
- The overall probability of accepting the sample from the posterior **decreases rapidly** as the number of observed variables increases.
- Rarely used in practice.

Basic Sampling Algorithm

- How can we generate samples from **simple non-uniform distributions** assuming we can generate samples from uniform distribution.



- Define:

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

- Sample:

$$z \sim \text{U}[0, 1]$$

- Then

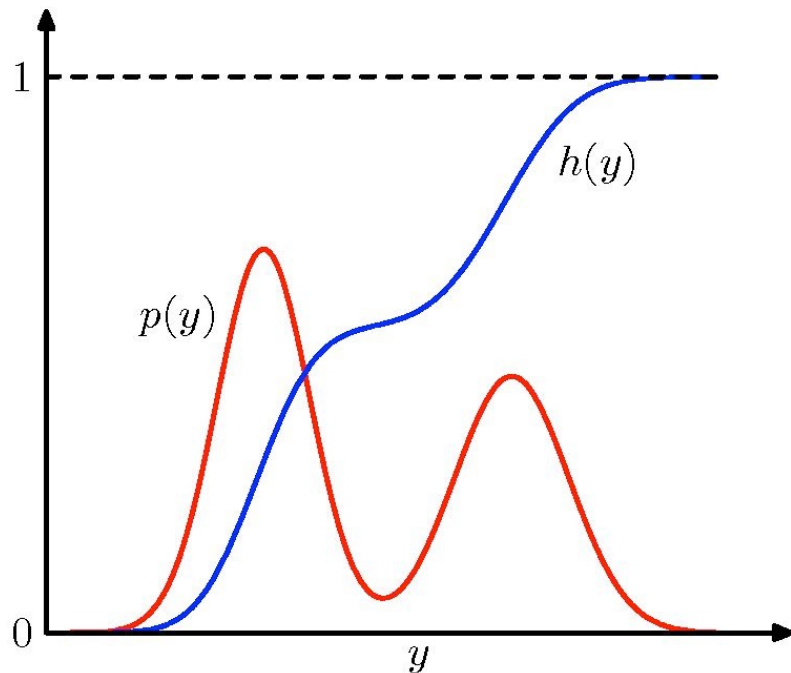
$$y = h^{-1}(z)$$

is a sample from $p(y)$.

Basic Sampling Algorithm

- For example, consider the [exponential distribution](#):

$$p(y) = \lambda \exp(-\lambda y).$$



- In this case:

$$h(y) = \int_0^y p(\hat{y}) d\hat{y} = 1 - \exp(-\lambda y).$$

- Sample:

$$z \sim \text{U}[0, 1]$$

- Then

$$y = h^{-1}(z) = -\lambda^{-1} \ln(1 - z)$$

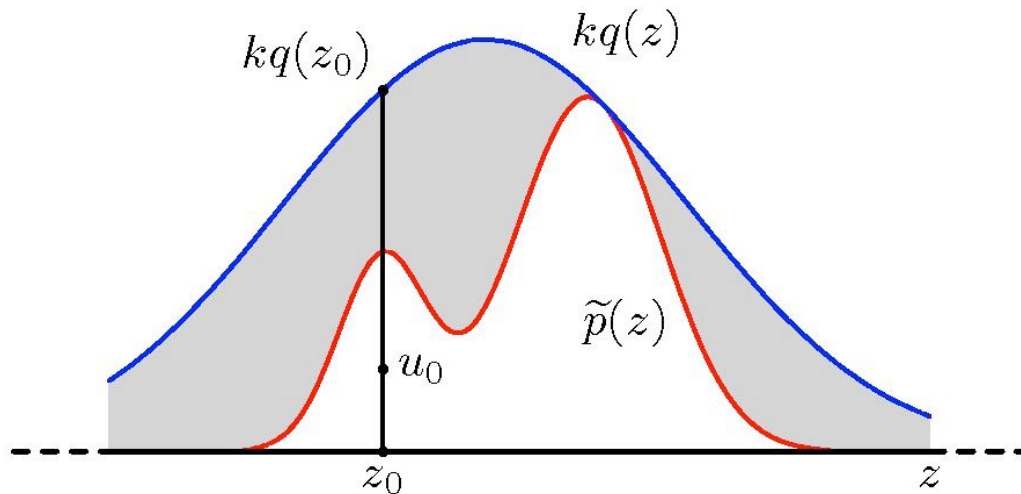
is a sample from $p(y)$.

- **Problem:** Computing $h(y)$ is **just as hard!**

Rejection Sampling

- Sampling from the **target distribution** $p(z) = \tilde{p}(z)/\mathcal{Z}_p$ is difficult. Suppose we have an **easy-to-sample proposal distribution** $q(z)$, such that:

$$kq(z) \geq \tilde{p}(z), \quad \forall z.$$



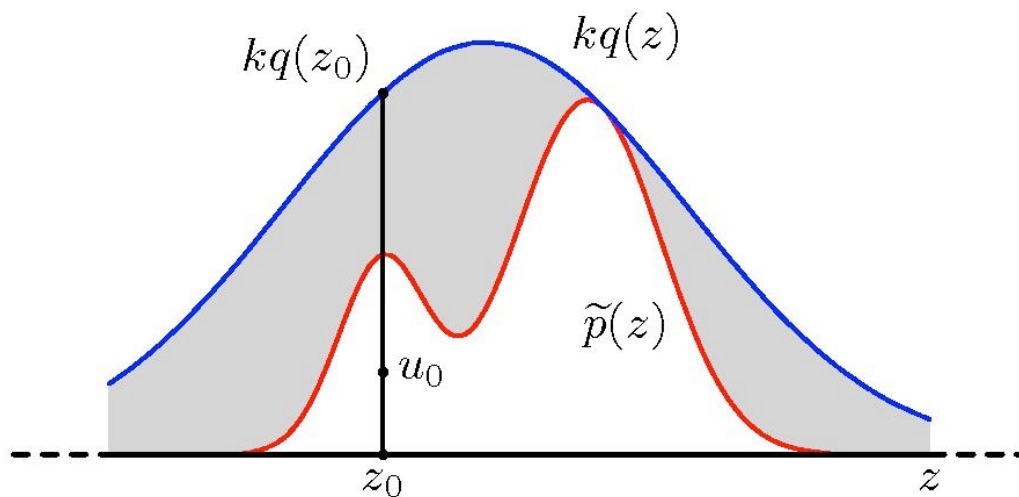
- Sample:
 $z_0 \sim q(z),$
- Sample:
 $u_0 \sim \text{Uniform}[0, kq(z_0)].$

- Sample (z_0, u_0) has **uniform distribution** under the curve of $kq(z)$.
- If $u_0 > \tilde{p}(z_0)$, the sample is **rejected**.

Rejection Sampling

- Probability that a sample is accepted is calculated as:

$$\begin{aligned} p(\text{accept}) &= \int \frac{\tilde{p}(z)}{kq(z)} q(z) dz \\ &= \frac{1}{k} \int \tilde{p}(z) dz. \end{aligned}$$

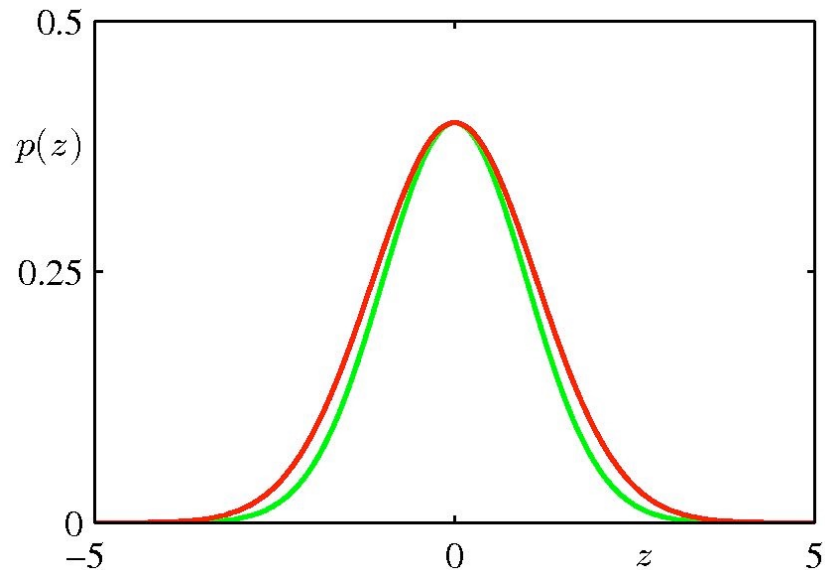


- The fraction of accepted samples depends on the ratio of the area under $\tilde{p}(z)$ and $kq(z)$.

- It is often hard to find $q(z)$ with optimal k .

Rejection Sampling

- Consider the following simple problem:



- Target distribution:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \sigma_p^2 I),$$

- Proposal distribution:

$$q(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \sigma_q^2 I).$$

- We must have:

$$\sigma_q^2 \geq \sigma_p^2.$$

- The optimal k is given by: $k = \left(\frac{\sigma_q}{\sigma_p} \right)^D$.

- Hence the **acceptance rate diminishes exponentially!**

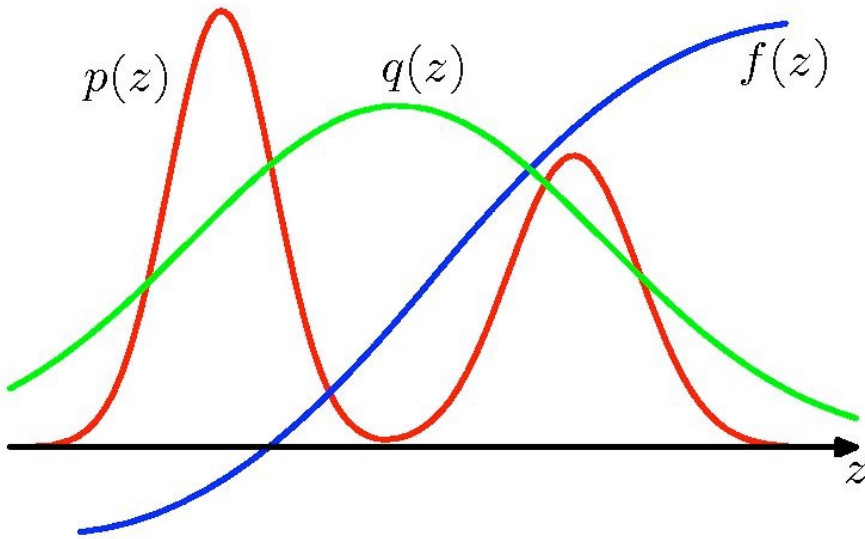
- Useful technique in one or two dimensions. Typically applies as a **subroutine in more advanced techniques.**

Importance Sampling

- Suppose we have an **easy-to-sample proposal distribution** $q(z)$, such that

$$q(z) > 0 \text{ if } p(z) > 0. \quad \mathbb{E}[f] = \int f(z)p(z)dz$$

$$= \int f(z) \frac{p(z)}{q(z)} q(z) dz$$
$$\approx \frac{1}{N} \sum_n \frac{p(z^n)}{q(z^n)} f(z^n), \quad z^n \sim q(z).$$



- The quantities

$$w^n = p(z^n)/q(z^n)$$

are known as **importance weights**.

- Unlike rejection sampling **all samples are retained**.

- But wait: we cannot compute $p(z) = \frac{\tilde{p}(z)}{\mathcal{Z}}$.

Importance Sampling

- Let our proposal be of the form: $q(z) = \tilde{q}(z) / \mathcal{Z}_q$.

$$\begin{aligned}\mathbb{E}[f] &= \int f(z)p(z)dz = \int f(z)\frac{p(z)}{q(z)}q(z)dz = \frac{\mathcal{Z}_q}{\mathcal{Z}_p} \int f(z)\frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz \\ &\approx \frac{\mathcal{Z}_q}{\mathcal{Z}_p} \frac{1}{N} \sum_n \frac{\tilde{p}(z^n)}{\tilde{q}(z^n)} f(z^n) = \frac{\mathcal{Z}_q}{\mathcal{Z}_p} \frac{1}{N} \sum_n w^n f(z^n),\end{aligned}$$

- But we can use the same weights to approximate $\mathcal{Z}_q / \mathcal{Z}_p$:

$$\frac{\mathcal{Z}_p}{\mathcal{Z}_q} = \frac{1}{\mathcal{Z}_q} \int \tilde{p}(z)dz = \int \frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz \approx \frac{1}{N} \sum_n \frac{\tilde{p}(z^n)}{\tilde{q}(z^n)} = \frac{1}{N} \sum_n w^n.$$

- Hence:

$$\mathbb{E}[f] \approx \sum_{n=1}^N \frac{w^n}{\sum_{m=1}^N w^m} f(z^n), \quad z^n \sim q(z).$$

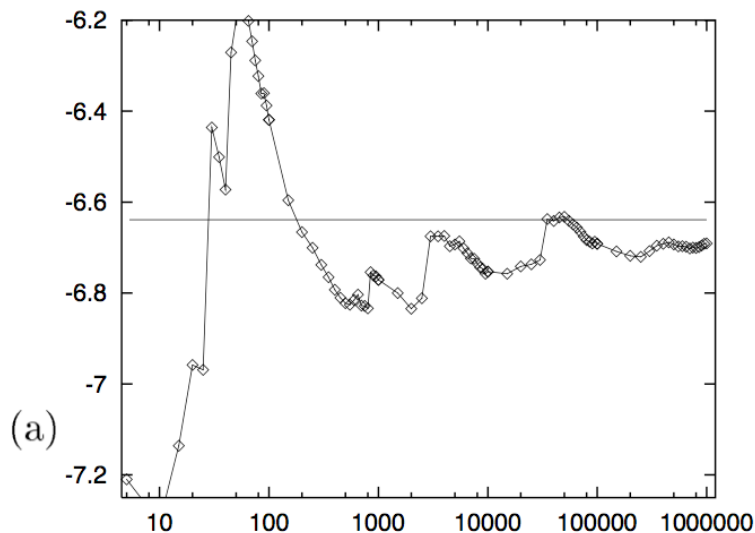
Consistent but biased²⁰

Importance Sampling: Example

- With importance sampling, it is hard to estimate how **reliable the estimator** is:

$$\hat{f} = \frac{\sum_{n=1}^N w^n f(z^n)}{\sum_{m=1}^N w^m}, \quad \mathbb{E}[f] = \int f(z) \frac{p(z)}{q(z)} q(z) dz$$

- **Huge variance** if the proposal density $q(z)$ is small in a region where $|f(z)p(z)|$ is large



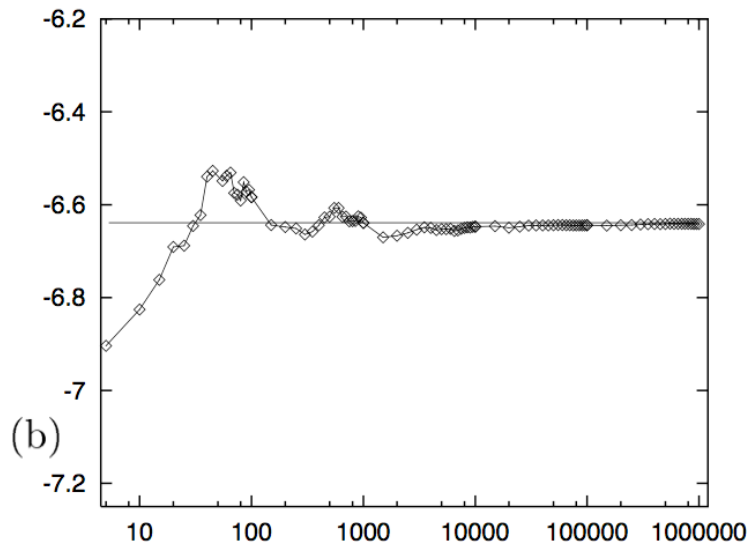
- Example of using Gaussian distribution as a proposal distribution (1-d case).
- Even **after 1 million samples**, the estimator has not converged to the true value.

Importance Sampling: Example

- With importance sampling, it is hard to estimate how reliable the estimator:

$$\hat{f} = \frac{\sum_{n=1}^N w^n f(z^n)}{\sum_{m=1}^N w^m}, \quad \mathbb{E}[f] = \int f(z) \frac{p(z)}{q(z)} q(z) dz$$

- Huge variance if the proposal density $q(z)$ is small in a region where $|f(z)p(z)|$ is large



- Example of using **Cauchy distribution** as a proposal distribution (1-d case).
- After 500 samples, the estimator appears to converge
- Proposal distribution **should have heavy tails.**

Summary so Far

- If our proposal distribution $q(z)$ poorly matches our target distribution $p(z)$ then:
 - **Rejection sampling**: almost always rejects
 - **Importance Sampling**: has large, possibly infinite, variance (unreliable estimator).
- For high-dimensional problems, finding good proposal distributions is very hard.
What can we do?
- Markov Chain Monte Carlo.

Markov Chains

- A **first-order Markov chain**: a series of random variables $\{z^1, \dots, z^N\}$, such that the following conditional independence property holds for $n \in \{1, \dots, N-1\}$:

$$p(z^{n+1} | z^1, \dots, z^n) = p(z^{n+1} | z^n).$$



- We can specify Markov chain:
 - Probability distribution for **initial state** $p(z^1)$.
 - **Conditional probability** for subsequent states in the form of transition probabilities:

$$T(z^{n+1} \leftarrow z^n) = p(z^{n+1} | z^n).$$

- $T(z^{n+1} \leftarrow z^n)$ is often called a **transition kernel**.

Markov Chains

- A **marginal probability** of a particular state can be computed as:

$$p(z^{n+1}) = \sum_{z^n} T(z^{n+1} \leftarrow z^n) p(z^n).$$

- A distribution $\pi(z)$ is said to be **invariant** or **stationary** with respect to a Markov chain if each step in the chain leaves $\pi(z)$ invariant:

$$\pi(z) = \sum_{z'} T(z \leftarrow z') \pi(z').$$

- A given Markov chain may have **many stationary distributions**.
- For example:

$$T(z \leftarrow z') = I(z = z')$$

is the identity transformation. Then **any distribution is invariant**.

Detailed Balance

- A sufficient (but not necessary) condition for ensuring that $\pi(z)$ is invariant is to choose a transition kernel that satisfies a **detailed balance property**:

$$\pi(z')T(z \leftarrow z') = \pi(z)T(z' \leftarrow z). \quad T(z' \leftarrow z) = p(z'|z).$$

- A transition kernel that satisfies detailed balance **will leave that distribution invariant**:

$$\begin{aligned} \sum_{z'} \pi(z')T(z \leftarrow z') &= \sum_{z'} \pi(z)T(z' \leftarrow z) \\ &= \pi(z) \sum_{z'} T(z' \leftarrow z) = \pi(z). \end{aligned}$$

- A Markov chain that satisfies detailed balance is said to be **reversible**.

Example

- Discrete example:

$$P^* = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} \quad T = \begin{pmatrix} 2/3 & 1/2 & 1/2 \\ 1/6 & 0 & 1/2 \\ 1/6 & 1/2 & 0 \end{pmatrix} \quad T_{ij} = T(x_i \leftarrow x_j)$$

- In this case P^* is **invariant distribution** of T since $TP^* = P^*$, or:

$$\sum_{z'} P^*(z') T(z \leftarrow z') = P^*(z).$$

- P^* is also the **equilibrium distribution** of T since:

$$T^{100} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} = P^*$$

Markov Chains

- We want to sample from the **target distribution** (e.g. posterior distribution, or a Markov Random Field):

$$\pi(z) = \tilde{\pi}(z) / \mathcal{Z}.$$

- Obtaining independent samples is difficult.
 - Set up a Markov chain with transition kernel $T(z' \leftarrow z)$ that **leaves our target distribution** $\pi(z)$ invariant.
 - If the **chain is ergodic**, then the chain will converge to this unique equilibrium distribution.
 - We obtain **dependent samples drawn approximately** from $\pi(z)$ by simulating a Markov chain for some time.

- **Ergodicity** requires: There exists K , such that for any starting z , we have

$$T^K(z' \leftarrow z) > 0 \text{ for all } \pi(z') > 0.$$

Combining Transition Operators

- In practice, we often construct the transition probabilities from a set of “**base**” transition operators B_1, \dots, B_K .
- One option is to consider a **mixture distribution** of the form:

$$T(z' \leftarrow z) = \sum_{k=1}^K \alpha_k B_k(z' \leftarrow z),$$

where mixing coefficients satisfy: $\alpha_k \geq 0$, $\sum_k \alpha_k = 1$.

- Another option is to **combine through successive application**:

$$T(z' \leftarrow z) = \sum_{z^1} \dots \sum_{z^{n-1}} B_1(z' \leftarrow z^1) \dots B_K(z^{K-1} \leftarrow z).$$

- If a distribution is **invariant** with respect to each of the base transitions, then it will also be **invariant** with respect to $T(z' \leftarrow z)$.

Combining Transition Operators

- For the case of the mixture:

$$T(z' \leftarrow z) = \sum_{k=1}^K \alpha_k B_k(z' \leftarrow z),$$

If each of the base distributions satisfies the detailed balance, then the mixture transition T will also satisfy detailed balance.

- For the case of using composite transition probabilities:

$$T(z' \leftarrow z) = \sum_{z^1} \dots \sum_{z^{n-1}} B_1(z' \leftarrow z^1) \dots B_K(z^{K-1} \leftarrow z).$$

this **does not hold**.

- A simple idea is to **symmetrize** the order of application of the base transitions:

$$B_1, B_2, \dots, B_K, B_K, \dots, B_2, B_1.$$

- A common example of using composite transition probabilities is where **each base transition changes only a subset of variables**.

Metropolis-Hasting Algorithm

• A **Markov chain transition operator** from the current state z to a new state z' is defined as follows:

- A new “candidate” state z^* is proposed according to some **proposal distribution** $q(z^*|z)$.
- A candidate z^* is **accepted with probability**:

$$\min \left(1, \frac{\tilde{\pi}(z^*) q(z|z^*)}{\tilde{\pi}(z) q(z^*|z)} \right).$$

- If accepted, set $z' = z^*$. Otherwise $z = z'$, or the next state is the copy of the current state.

• Note: there **is no need to compute normalizing constant**.

• For symmetric proposals, e.g. $N(z, \sigma^2)$, the acceptance probability reduces to:

$$\min \left(1, \frac{\tilde{\pi}(z^*)}{\tilde{\pi}(z)} \right).$$

Metropolis-Hasting Algorithm

- We can show that M-H transition kernel leaves $\pi(z)$ **invariant** by showing that it satisfies **detailed balance**:

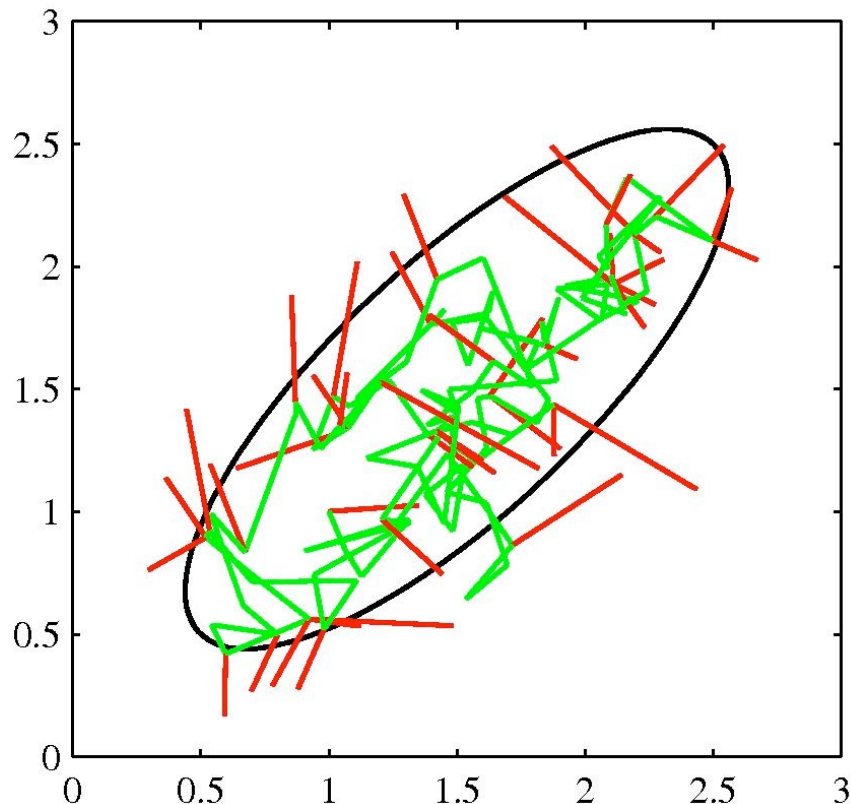
$$\begin{aligned}\pi(z)T(z' \leftarrow z) &= \pi(z)q(z'|z) \min \left(1, \frac{\pi(z')}{\pi(z)} \frac{q(z|z')}{q(z'|z)} \right) \\ &= \min (\pi(z)q(z'|z), \pi(z')q(z|z')) \\ &= \pi(z')q(z|z') \min \left(\frac{\pi(z)}{\pi(z')} \frac{q(z'|z)}{q(z|z')}, 1 \right) \\ &= \pi(z')T(z \leftarrow z').\end{aligned}$$

- Note that **whether the chain is ergodic** will depend on the particulars of the stationary distribution π and proposal distribution q .

Metropolis-Hasting Algorithm

- Using Metropolis algorithm to sample from Gaussian distribution with proposal

$$q(z'|z) = \mathcal{N}(z, 0.04).$$



- **accepted** (green), **rejected** (red).
- 150 samples were generated and 43 were rejected.
- Note that **generated samples are not independent**.

Random Walk Behaviour

- Consider a state-space consisting of integers with

$$\begin{aligned}p(z^{t+1} = z^t) &= 0.5 \\p(z^{t+1} = z^t + 1) &= 0.25 \\p(z^{t+1} = z^t - 1) &= 0.25\end{aligned}$$

- If the initial state is $z^1 = 0$, then by symmetry:

$$\mathbb{E}[z^t] = 0,$$

- and

$$\mathbb{E}[(z^t)^2] = t/2.$$

- Hence after t steps, the random walk traveled a distance that is on average **proportional to the square root** of t .
- This square root dependence is typical of random walk behavior.
- Ideally, we would want to design MCMC methods that **avoid random walk** behavior.

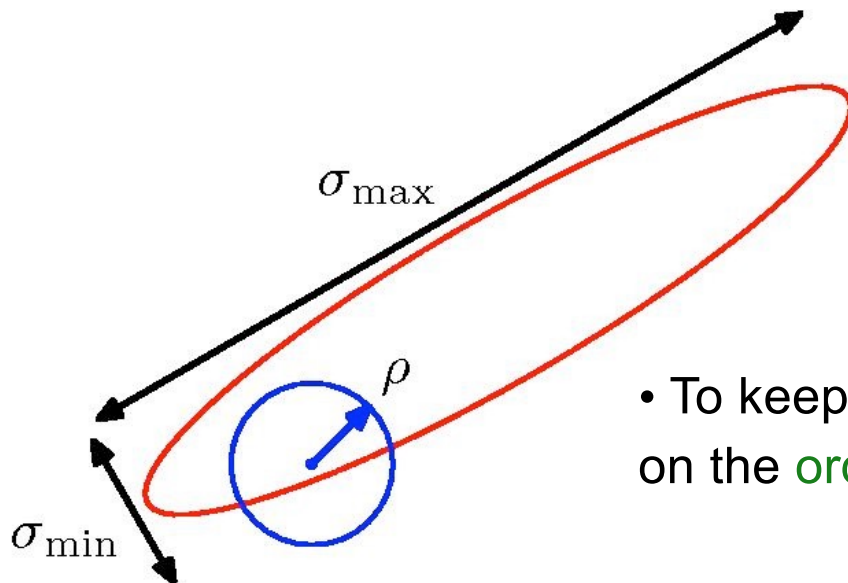
Choice of Proposal

- Suppose that our goal is to sample from the correlated multivariate Gaussian distribution.
- Consider a Gaussian proposal: centered on the current state:

$$q(z'|z) = \mathcal{N}(z, \rho^2 I)$$

- ρ large -- many rejections
- ρ small -- chain moves too slowly.

- To keep the rejection rate low, the scale ρ should be on the order of the smallest standard deviation σ_{min} .

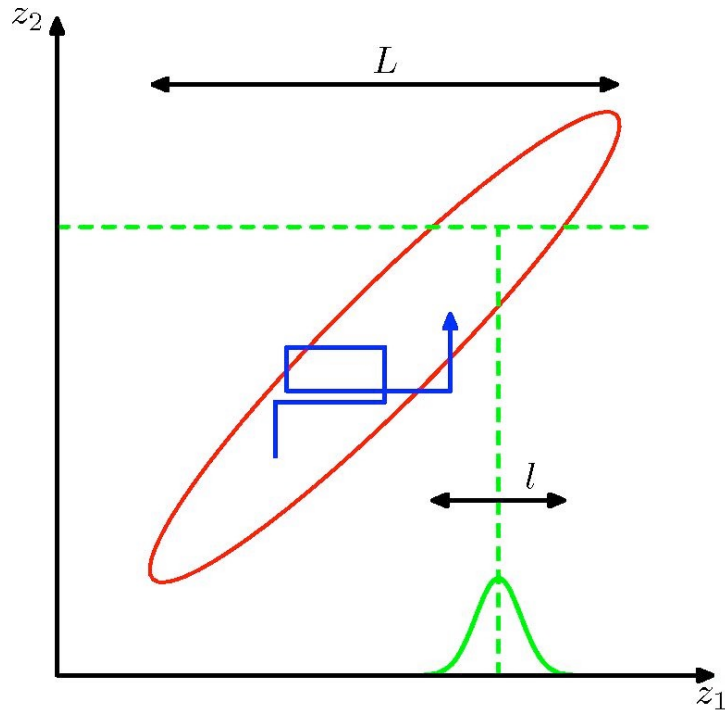


- **Random walk behaviour:** The number of steps separating states that are approximately independent is of order: $(\sigma_{max}/\sigma_{min})^2$.

- The specific choice of proposal can greatly affect the performance of the algorithm.

Gibbs Sampler

- Consider sampling from $p(z_1, \dots, z_N)$:



- Initialize $z_i, i=1, \dots, N$.

- For $t=1:T$

- Sample: $z_1^{t+1} \sim p(z_1 | z_2^t, \dots, z_N^t)$
- Sample: $z_2^{t+1} \sim p(z_2 | z_1^{t+1}, z_3^t, \dots, z_N^t)$
- ...
- Sample: $z_N^{t+1} \sim p(z_N | z_1^{t+1}, z_2^{t+1}, \dots, z_{N-1}^{t+1})$

- This procedure samples from the required distribution $p(\mathbf{z})$.

- When sampling $p(z_n | \mathbf{z}_{-n})$ the marginal distribution $p(\mathbf{z}_{-n})$ is clearly **invariant**, as it does not change.

- Each step samples from the correct conditional, hence the **joint distribution is itself invariant**.

Gibbs Sampler

- Applicability of the Gibbs sampler depends on how easy it is to sample from conditional probabilities $p(z_n | \mathbf{z}_{-n})$.
- For discrete random variables with a few discrete settings:

$$p(z_n | \mathbf{z}_{-n}) = \frac{p(z_n, \mathbf{z}_{-n})}{\sum_{z_n} p(z_n, \mathbf{z}_{-n})},$$

where the sum can be performed analytically.

- For continuous random variables:

$$p(z_n | \mathbf{z}_{-n}) = \frac{p(z_n, \mathbf{z}_{-n})}{\int p(z_n, \mathbf{z}_{-n}) dz_n},$$

- The integral is univariate and is often analytically tractable or amenable to standard sampling methods.

Gibbs Sampler

- Gibbs sampler is a particular instance of M-H algorithm with proposals:

$$q_n(\mathbf{z}^* | \mathbf{z}) = p(z_n^* | \mathbf{z}_{-n}).$$

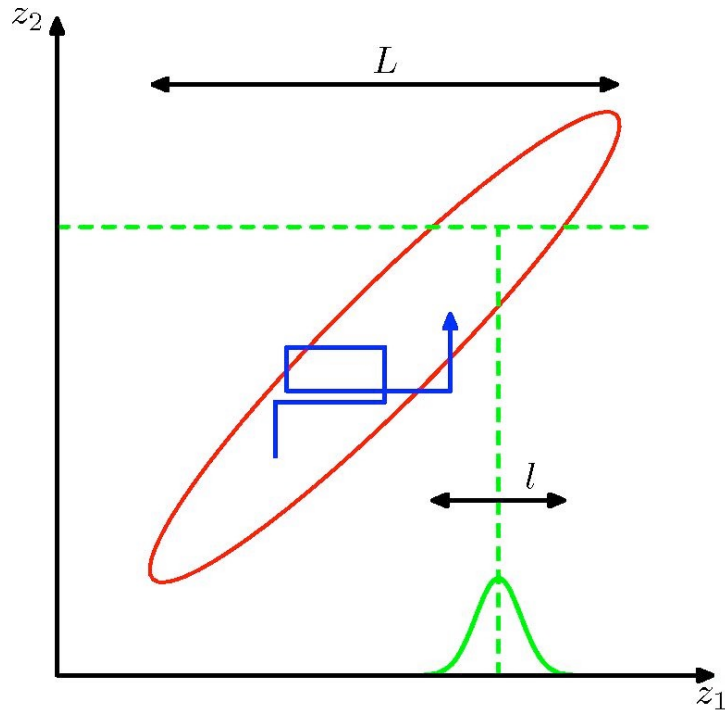
- Note that $\mathbf{z}_{-n}^* = \mathbf{z}_{-n}$ because these components are unchanged by the sampling step.
- Let us look at the factor that determines acceptance probability in M-H.

$$\begin{aligned} A(\mathbf{z}^*, \mathbf{z}) &= \frac{p(\mathbf{z}^*)}{p(\mathbf{z})} \times \frac{q_n(\mathbf{z} | \mathbf{z}^*)}{q_n(\mathbf{z}^* | \mathbf{z})} \\ &= \frac{p(z_n^* | \mathbf{z}_{-n}^*) p(\mathbf{z}_{-n}^*)}{p(z_n | \mathbf{z}_{-n}) p(\mathbf{z}_{-n})} \times \frac{p(z_n | \mathbf{z}_{-n}^*)}{p(z_n^* | \mathbf{z}_{-n})} = 1. \end{aligned}$$

- Thus MH steps **are always accepted**.
- Let us look at the behavior of Gibbs.

Gibbs Sampler

- As with MH, we can get some insight into the behavior of Gibbs sampling.



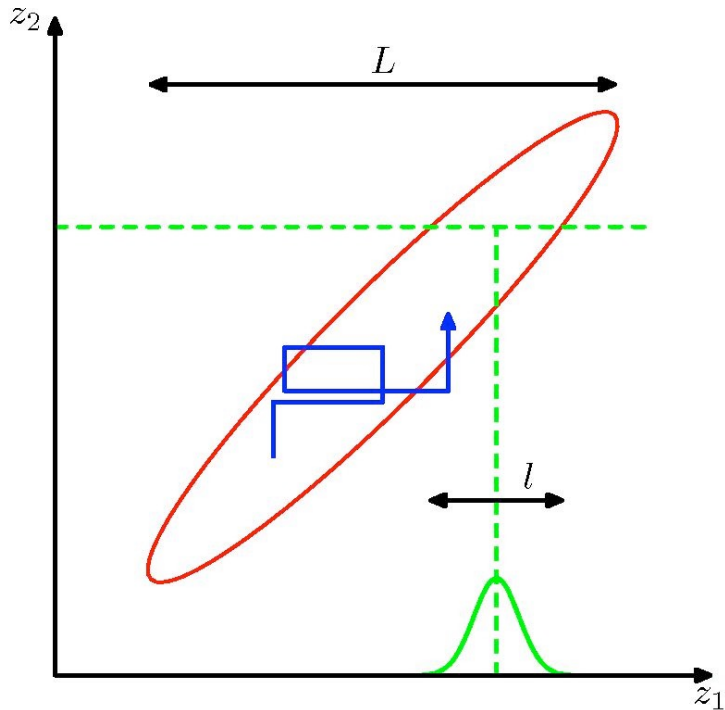
- Consider a correlated Gaussian having conditional distributions of width l and marginal distributions of width L .
- **Random walk behavior:** The typical step size is governed by the conditional and will be of order l .
- The number of steps separating states that are **approximately independent** is of order:

$$O((L/l)^2).$$

- If the Gaussian distribution were uncorrelated, then the Gibbs sampling would be optimally efficient.

Over-Relaxation

- One approach to reducing random walk behavior is called **over-relaxation**:



- Consider conditional distributions that are Gaussian.

- At each step of the Gibbs sampler, the conditional distribution for z_i is:

$$p(z_n | \mathbf{z}_{-n}) = \mathcal{N}(z_n | \mu_n, \sigma_n^2).$$

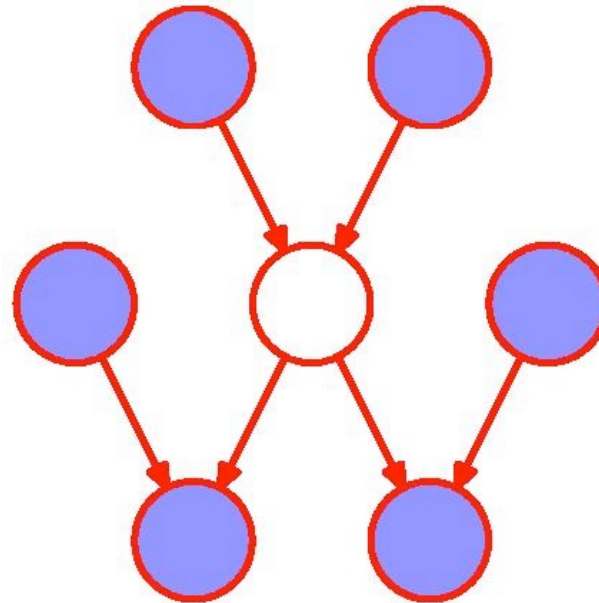
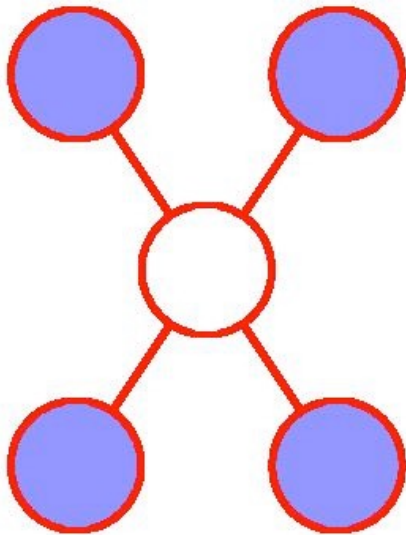
- In the **over-relaxed framework**, the value of z_n is replaced with:

$$z'_n = \mu_n + \alpha(z_n - \mu_n) + \sigma_n(1 - \alpha^2)^{1/2}\nu, \\ \nu \sim \mathcal{N}(0, 1).$$

- Setting $\alpha = 0$, we **recover standard Gibbs**.
- The step leaves the **desired distribution invariant** because if z_n has mean μ_n and standard deviation σ_n , then so does z'_n .
- This encourages directed motion through the state space when the **variables are high correlated**.

Graphical Models

- For graphical models, the conditional distribution is a function only of the states of the nodes in the **Markov blanket**.



- **Block Gibbs:** Choose blocks of variables (not necessarily disjoint) and then sample jointly from the variables in each block in turn, conditioned on the remaining variables.

Using MCMC in Practice

- The samples we obtain from MCMC are not independent. Should we thin, i.e. only keep every K th sample?
- We often start MCMC from arbitrary starting points. Should we discard a burn-in period?
- Should we perform multiple runs as opposed to one long run?
- How do we know whether we have run our chain for long enough?

