Computer Science

CSC580: Principles of Machine Learning

Monte Carlo Methods

Prof. Jason Pacheco

Some material from:
Prof. Erik Sudderth & Prof. Kobus Barnard

Outline

- Monte Carlo Estimation
- Markov Chain Monte Carlo

Outline

- Monte Carlo Estimation
- Markov Chain Monte Carlo

Motivation for Monte Carlo Methods

- Real problems are typically complex and high dimensional.

- Suppose that we *could* generate samples from a distribution that is proportional to one we are interested in.

- Typically we want posterior samples,

$$p(z \mid \mathcal{D}) = \frac{p(z)p(\mathcal{D} \mid z)}{p(\mathcal{D})} \propto \widetilde{p}(z)$$

Unnormalized posterior

Don't know marginal likelihood / normalizer

- Typically, $\widetilde{p}(z)$ is easier to evaluate (though not always)

Motivation for Monte Carlo Methods

- Generally, Z lives in a very high dimensional space.
- Generally, regions of high $\tilde{p}(z)$ is very little of that space.
- IE, the probability mass is very localized.
- Watching samples from $\tilde{p}(z)$ should provide a good maximum (one of our inference problems)

Motivation for Monte Carlo Methods

- Now consider computing the expectation of a function $f(z)$ w.r.t $p(z)$.
- Recall that this looks like $E_{p(z)}[f] = \int_z f(z)p(z)dz$
- How can we approximate or estimate E[f]?

A bad plan…

Discretize the space where z lives into L blocks

Then compute $E_{p(z)}[f] \cong \frac{1}{L}\sum_{l=1}^{L} p(z)f(z)$

Scales poorly with dimension of Z

A better plan…

Given independant samples $z^{(l)}$ from $\tilde{p}(z)$

Estimate $E_{p(z)}[f] \cong \frac{1}{L}\sum_{l=1}^{L} f(z)$

Challenges for Monte Carlo Methods

- In real problems sampling $p(z)$ is very difficult
- Typically don't know normalization, so need to use $\widetilde{p}(z)$ instead
- Even if we **can** sample $p(z)$, it can be hard to know if/when they are "good" and if we have enough (e.g. to approximate E[f] well)
- Sometimes evaluating $\widetilde{p}(z)$ can also be hard

Inference (and related) Tasks

- Simulation: $x \sim p(x) = \frac{1}{Z} f(x)$
- Compute expectations: $\mathbb{E}[\phi(x)] = \int p(x)\phi(x)\,dx$
- Optimization: $x^* = \arg\max_x f(x)$
- Compute normalizer / marginal likelihood: $Z = \int f(x)\,dx$

Inference (and related) Tasks

- Simulation: $x \sim p(x) = \frac{1}{Z} f(x)$

- Compute expectations: $\mathbb{E}[\phi(x)] = \int p(x)\phi(x)\, dx$

- Optimization: $x^* = \arg\max_x f(x)$

- Compute normalizer / marginal likelihood: $Z = \int f(x)\, dx$

Basic Sampling (so far…)

- Uniform sampling (everything builds on this)
- Sampling from simple discrete distributions
  - Multinomial / categorical
  - Binomial / Bernoulli
  - Etc.
- Sampling for selected continuous distributions (e.g., Gaussian)
  - At least, Matlab and Numpy / Scipy know how to do it.
- Ancestral sampling

Sampling Continuous RVs

Recall that the CDF is the integral of the PDF and (left) tail probability,

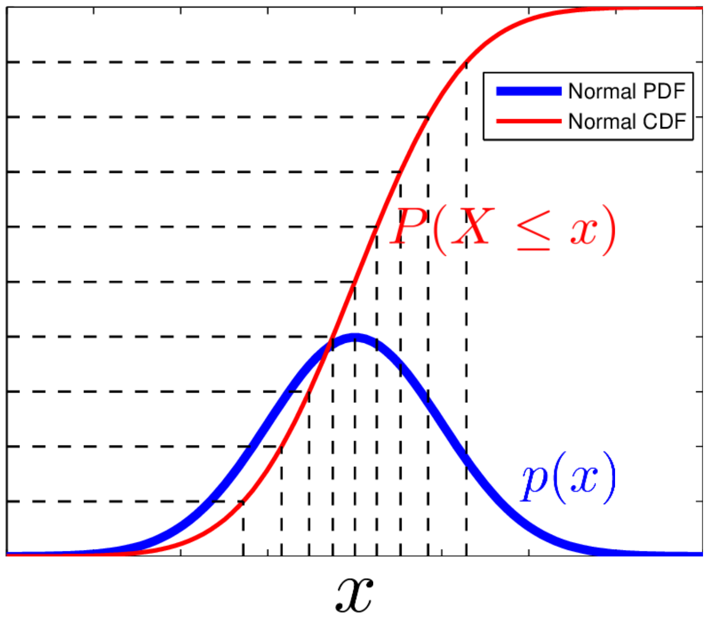
$$P(X \leq x) = \int_{-\infty}^{x} p(X = t)\, dt$$

Observation 1 Equally spaced intervals of CDF correspond to regions of equal event probability

Observation 2 The same events have unequal regions under PDF

Question Given samples $\{x_i\}_{i=1}^N \sim p(x)$ what is the probability distribution of the CDF values,

$$\{P(X \leq x_i)\}_{i=1}^N \sim ???$$



Sampling Continuous RVs

Answer The CDF of iid samples has a **standard uniform** distribution!

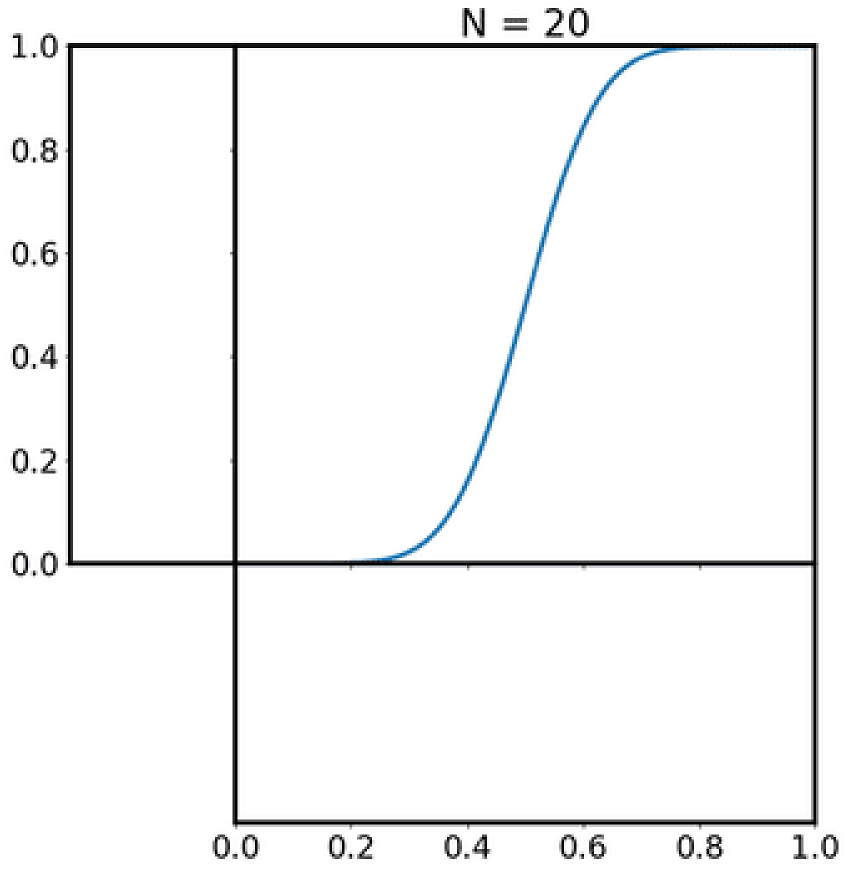
$$\{P(X \leq x_i)\}_{i=1}^N \sim \text{Uniform}(0,1)$$

This property is called the **probability integral transform**

Question How can we use this fact to sample *any* RV?

Answer Apply this relationship in reverse:

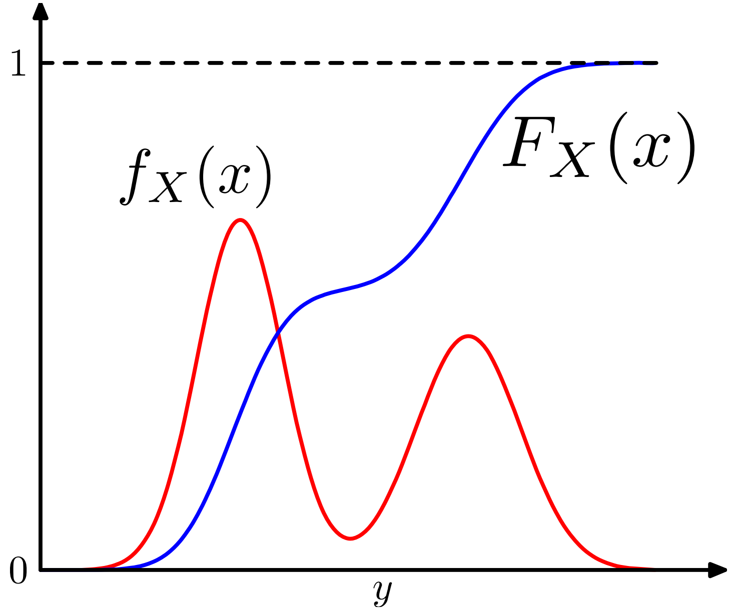
1. Sample iid standard uniform RVs
2. Compute *inverse* CDF
3. Result are samples from the target



Inverse Transform Sampling

- Input: Independent standard uniform variables $U_1, U_2, U_3, \ldots$
- We can use these to exactly sample from any continuous distribution using the cumulative distribution function:

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_X(z)\, dz$$



- Assuming continuous CDF is invertible:

$$h(u) = F_X^{-1}(u)$$

$$X_i = h(U_i)$$

Requires us to have access to inverse CDF

$$P(X_i \leq x) = P(h(U_i) \leq x) = P(U_i \leq F_X(x)) = F_X(x)$$

This function transforms uniform variables to our target distribution!

Inverse Transform Sampling

- Very nice trick that applies to *all* continuous RVs (in theory)
- Yay, we know how to sample any RV right? Wrong…
- Don't always have the *inverse* CDF (or cannot calculated it)
- Doesn't extend easily to multivariate RVs (that's why I only showed 1-dimensional)

Rejection Sampling

Assume

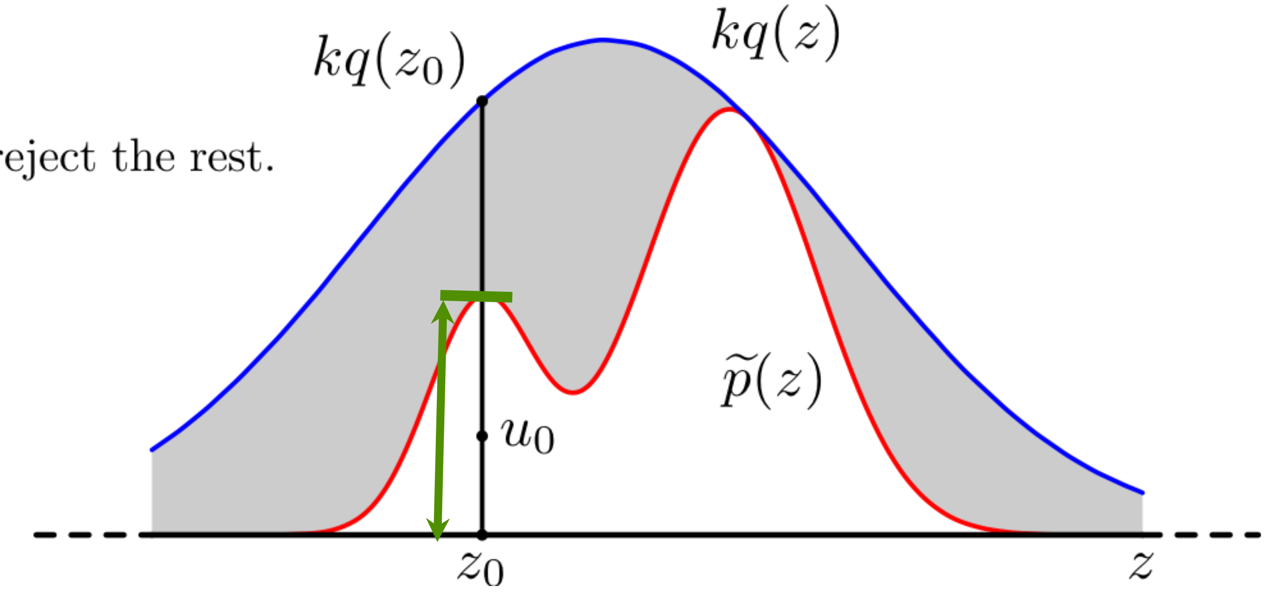
- Access to easy-to-sample distribution $q(z)$
- Constant *k* such that $\widetilde{p}(z) \leq k \cdot q(z)$

Proposal Distribution
Where we can use one of methods on previous slides to sample efficiently

Algorithm

1) Sample $q(z)$

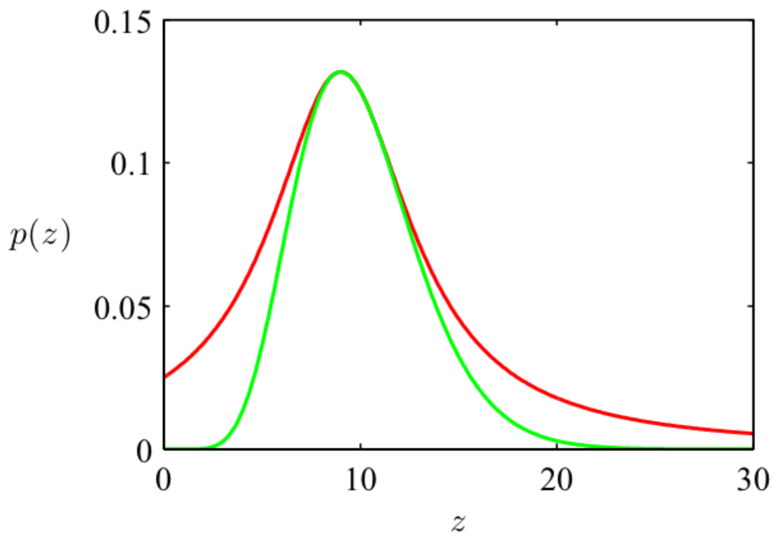
2) Keep samples in proportion to $\dfrac{\tilde{p}(z)}{k \bullet q(z)}$ and reject the rest.



Example Uses Gaussian proposal q to draw samples from multimodal distribution p

Rejection Sampling

- Rejection sampling is hopeless in high dimensions, but is useful for sampling low dimensional “building block” functions.
- For example, the Box-Muller method for generating samples from a Gaussian uses rejection sampling.



A second example where a gamma distribution is approximated by a Cauchy proposal distribution.

Inference (and related) Tasks

- Simulation: $x \sim p(x) = \frac{1}{Z} f(x)$
- Compute expectations: $\mathbb{E}[\phi(x)] = \int p(x)\phi(x)\, dx$
- Optimization: $x^* = \arg\max_x f(x)$
- Compute normalizer / marginal likelihood: $Z = \int f(x)\, dx$

Monte Carlo Integration

One reason to sample a distribution is to approximate expected values under that distribution…

Expected value of function $f(x)$ w.r.t. distribution $p(x)$ given by,

$$\mathbb{E}_p[f(x)] = \int p(x) f(x) \, dx \equiv \mu$$

- Doesn't always have a closed-form for arbitrary functions
- Suppose we have iid samples: $\{x_i\}_{i=1}^N \sim p(x)$
- *Monte Carlo* estimate of expected value,

$$\hat{\mu}_N = \frac{1}{N} \sum_{i=1}^N f(x_i) \approx \mathbb{E}_p[f(x)]$$

Monte Carlo Integration

$$\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \approx \mathbb{E}_p[f(x)]$$

- Expectation estimated from *empirical distribution* of N samples:

$$\hat{p}_N(x) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_i}(x) \qquad \{x_i\}_{i=1}^{N} \sim p(x)$$

- The *Dirac delta* is *loosely* defined as a piecewise function:

$$\delta_{x_i}(x) = \begin{cases} +\infty & x = x_i \\ 0 & x \neq x_i \end{cases}$$

Caveat This is technically incorrect. Dirac is only well-defined within integrals, $\int \delta_{\bar{x}}(x) f(x)\, dx = f(\bar{x})$ but it gets the intuition across.

- For any *N* this estimator, a random variable, is *unbiased*:

$$\mathbb{E}[\hat{\mu}_N] = \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \mathbb{E}_p[f(x)]$$

Monte Carlo Asymptotics

$$\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \approx \mathbb{E}_p[f(x)]$$

- Estimator variance reduces at rate 1/N:

$$\mathrm{Var}[\hat{\mu}_N] = \frac{1}{N}\mathrm{Var}[f] = \frac{1}{N}\mathbf{E}\left[(f(x)-\mu)^2\right]$$

Independent of dimensionality of random variable X

- If the true variance is **finite** have *central limit theorem*:

$$\sqrt{N}(\hat{\mu}_N - \mu) \underset{N\to\infty}{\Longrightarrow} \mathcal{N}(0, \mathrm{Var}[f])$$

- Even if true variance is **infinite** have *laws of large numbers*:

Weak Law

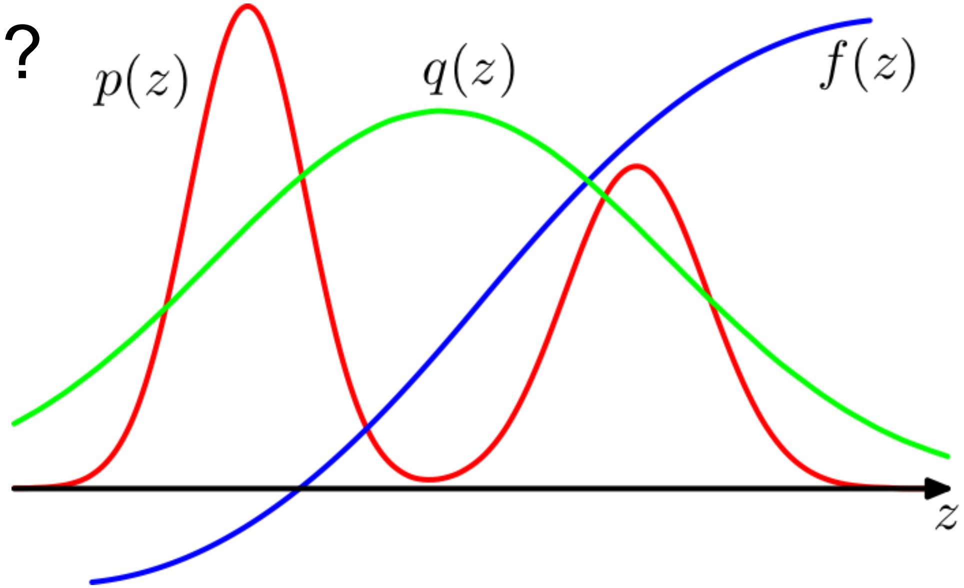
$$\lim_{N\to\infty} \Pr\left(|\hat{\mu}_N - \mu| < \epsilon\right) = 1, \qquad \text{for any} \epsilon > 0$$

Strong Law

$$\Pr\left(\lim_{N\to\infty} \hat{\mu}_N = \mu\right) = 1$$

Importance Sampling

Can we estimate $\mathbb{E}_p[f]$ without sampling p(z)?



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

q(z) is an easy-to-sample proposal distribution

$$= \int f(z)\frac{p(z)}{q(z)}q(z)\,dz$$

Monte Carlo estimate over samples $\{z_i\}_{i=1}^{N} \sim q$ from proposal q(z):

$$\mathbb{E}_p[f] \approx \frac{1}{N}\sum_{i=1}^{N}\frac{p(z_i)}{q(z_i)}f(z_i)$$

Key: We can sample from an "easy" distribution q(z) instead!

Importance Sampling

IS weights are the ratio of target / proposal distributions:

$$\mathbb{E}_p[f] \approx \frac{1}{N} \sum_{i=1}^{N} w_i f(z_i) \qquad \text{where} \qquad w_i = \frac{p(z_i)}{q(z_i)}$$

But we often do not know the normalizer of the target distribution,

$$p(z) = \frac{1}{Z_p} \widetilde{p}(z) \qquad \text{where} \qquad Z_p = \int \widetilde{p}(z)\, dz$$

Can only evaluate unnormalized target

Can we evaluate IS estimate in terms of unnormalized weights?

$$\widetilde{w}_i = \frac{\widetilde{p}(z_i)}{q(z_i)}$$

Yes! Let's see how…

Importance Sampling (Normalized)

Recall, the importance sampling estimate is given by,

$$\mathbb{E}_p[f] \approx \frac{1}{N} \sum_{i=1}^{N} \frac{p(z_i)}{q(z_i)} f(z_i)$$

With normalized target and proposal distributions, respectively:

$$p(z) = \frac{1}{Z_p} \widetilde{p}(z) \qquad\qquad q(z) = \frac{1}{Z_q} \widetilde{q}(z)$$

Substitute and pull out ratio of normalizers,

$$\mathbb{E}_p[f] \approx \left(\frac{Z_q}{Z_p} \right) \frac{1}{N} \sum_{i=1}^{N} \frac{\widetilde{p}(z_i)}{\widetilde{q}(z_i)} f(z_i)$$

Need to compute this…

Easy to compute

Importance Sampling (Normalized)

Idea Compute importance sampling estimate of target normalizer:

$$Z_p = \int \widetilde{p}(z)\, dz = \int \frac{\widetilde{p}(z)}{q(z)} q(z)\, dz \approx \frac{1}{N} \sum_{i=1}^{N} \frac{\widetilde{p}(z_i)}{q(z_i)}$$

Typically we have normalized proposal $q(z)$ so $Z_q=1$ and,

$$\frac{Z_p}{Z_q} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{\widetilde{p}(z_i)}{q(z_i)} = \frac{1}{N} \sum_{i=1}^{N} \widetilde{w}_i$$

Where $\widetilde{w}_i$ are our *unnormalized importance weights*,

$$\widetilde{w}_i = \frac{\widetilde{p}(z_i)}{q(z_i)}$$

We can compute this!

Importance Sampling (normalized)

Given samples $\{z_i\}_{i=1}^N \sim q$ we can write the IS estimate as,

$$\mathbb{E}_p[f] \approx \left(\frac{Z_q}{Z_p}\right) \frac{1}{N} \sum_{i=1}^N \widetilde{w}_i f(z_i)$$

The ratio of normalizers is approximated by normalized weights,

$$\frac{Z_p}{Z_q} \approx \frac{1}{N} \sum_{i=1} \widetilde{w}_i$$

Substituting the normalized weights yields,

$$\mathbb{E}_p[f] \approx \frac{\sum_{i=1}^N \widetilde{w}_i f(z_i)}{\sum_{j=1}^N \widetilde{w}_j} \qquad \text{where} \qquad \widetilde{w}_j = \frac{\widetilde{p}(z_j)}{\widetilde{q}(z_j)}$$

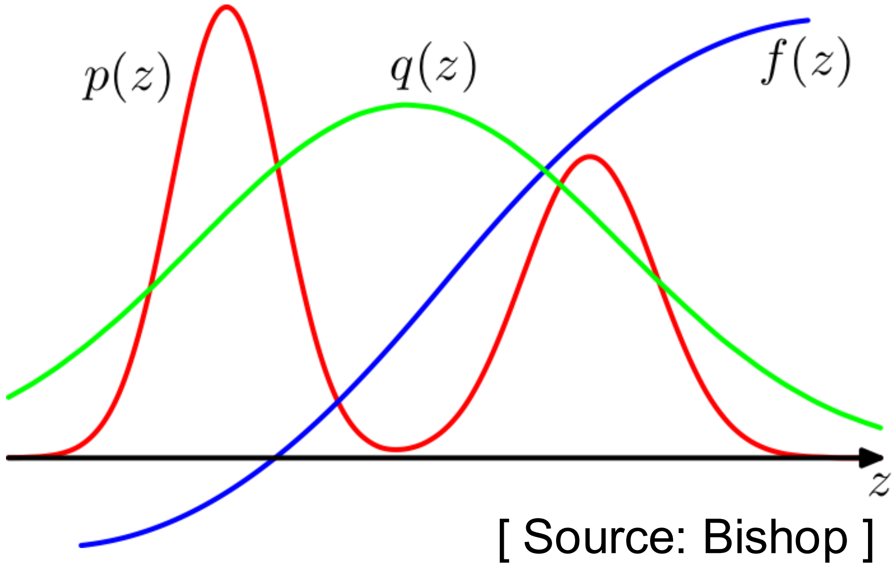
Importance Sampling On-A-Slide

1. Simulate from an "easy" distribution

$$\{z_i\}_{i=1}^N \sim q(z)$$

2. Compute importance weights & normalize

$$\widetilde{w}_i = \frac{\widetilde{p}(z_i)}{q(z_i)} \qquad w_i = \frac{\widetilde{w}_i}{\sum_{i=1}^N \widetilde{w}_i}$$



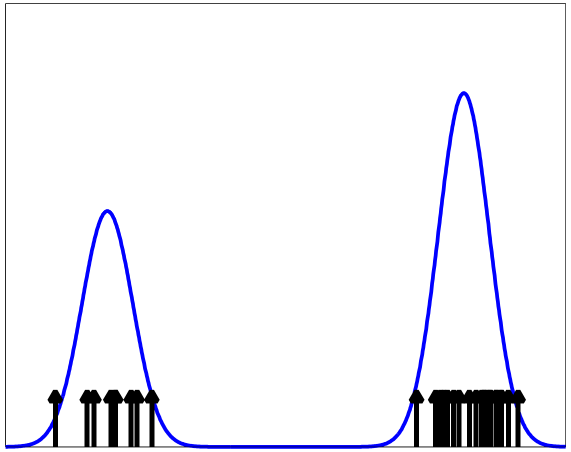
[Source: Bishop]

3. Compute importance-weighted expectation

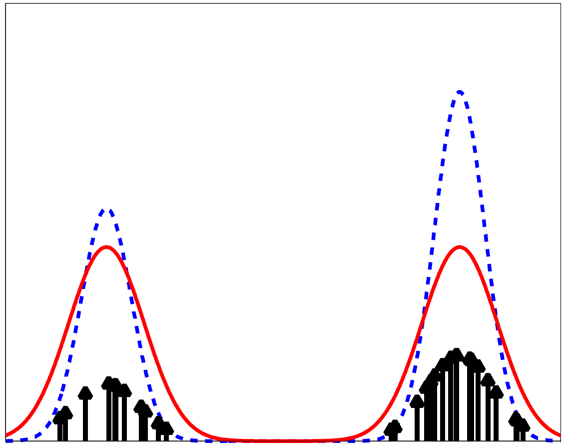
$$\mathbf{E}_p[f(z)] \approx \sum_{i=1}^N w_i f(z_i) \equiv \hat{f}$$

Note There is no 1/N term since it is part of the normalized IS weights

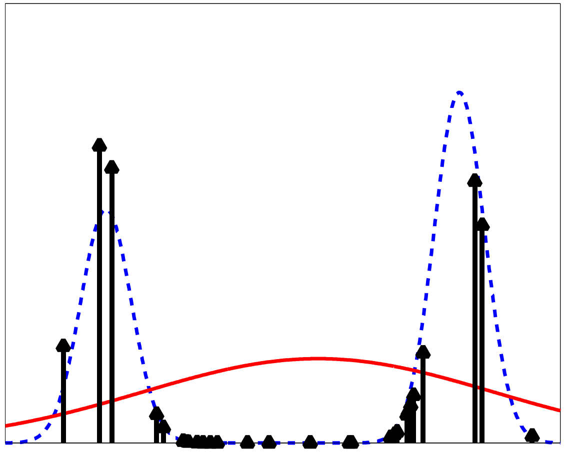
Selecting Proposal Distributions



Target Distribution



Good Proposal

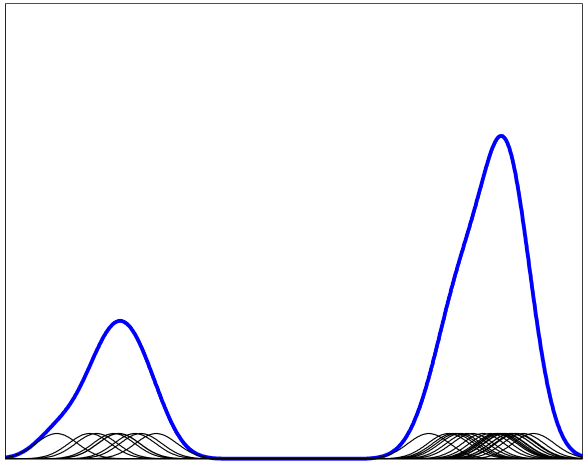


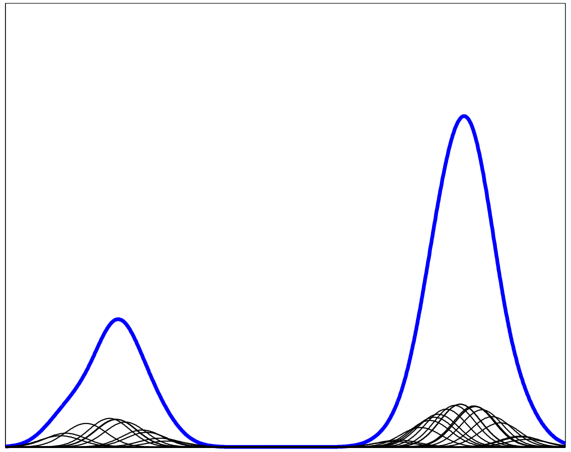
Poor Proposal

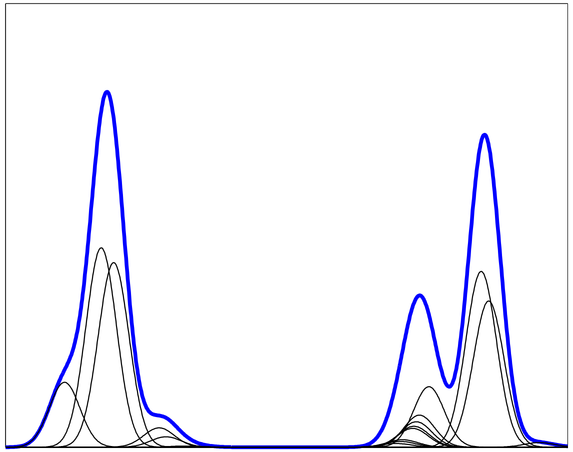
Kernel or Parzen window estimators interpolate to predict density:

$$\hat{p}(x) = \sum_{\ell=1}^{L} w^{(\ell)} \mathcal{N}(x; x^{(\ell)}, \Lambda)$$

$$w^{(\ell)} \propto \frac{p(x^{(\ell)})}{q(x^{(\ell)})}$$







Importance Sampling

Q: What is a good proposal distribution?

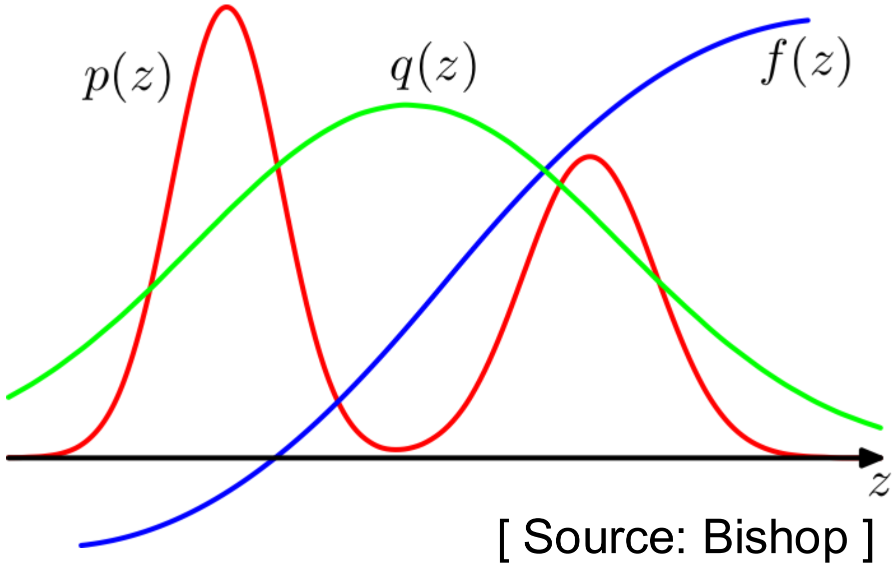
A: Minimize estimator variance

$$q^* = \arg\min_q \mathrm{Var}_q(\hat{f})$$

Minimum variance obtained when,

$$q^* \propto |f(z)|p(z)$$

E.g. can do better than q=p



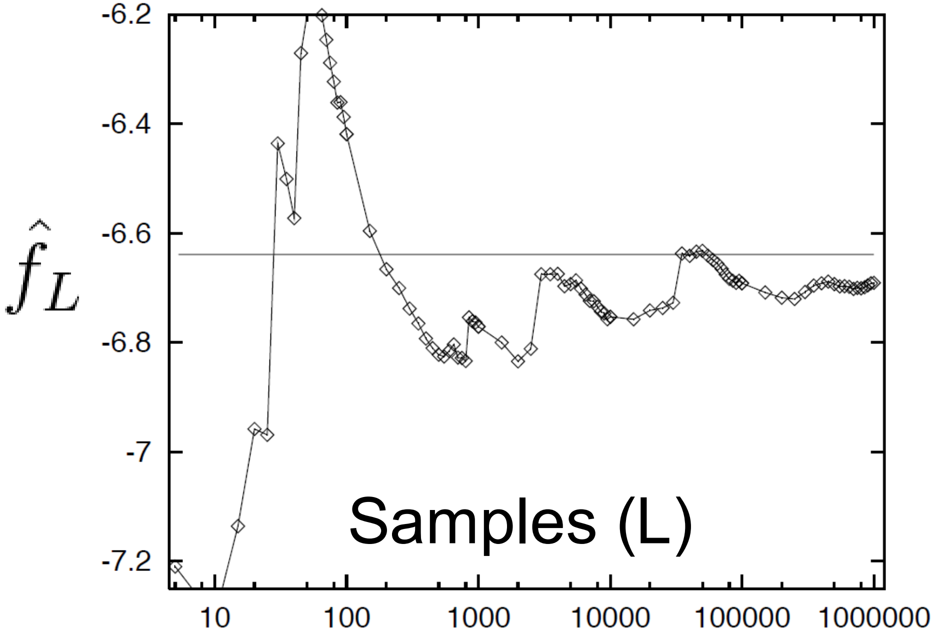
[Source: Bishop]

Estimator variance scales catastrophically with dimension:

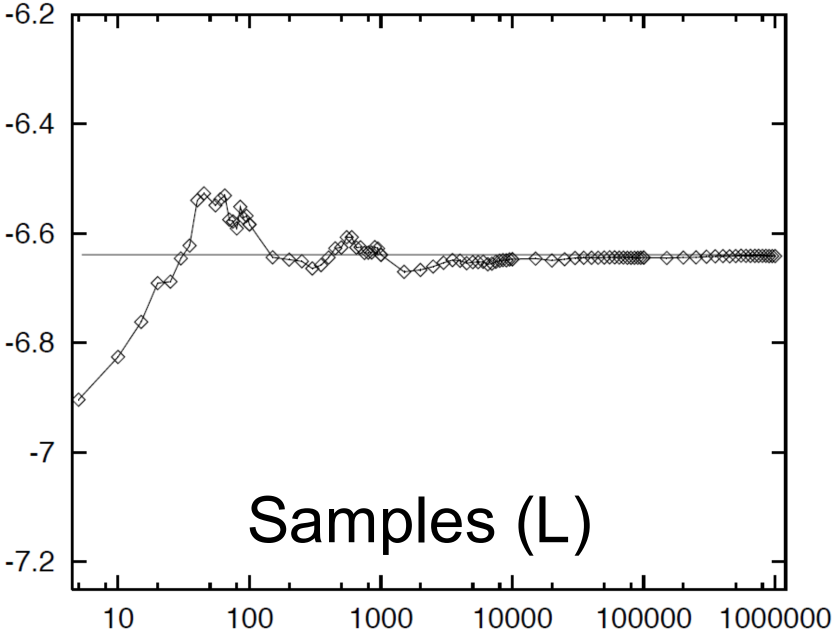
e.g. for N-dim. X and Gaussian q(x): $\mathrm{Var}_{q^*}(\hat{f}) = \exp(\sqrt{2N})$

Selecting Proposal Distributions

- For a toy one-dimensional, heavy-tailed target distribution:



Gaussian Proposal



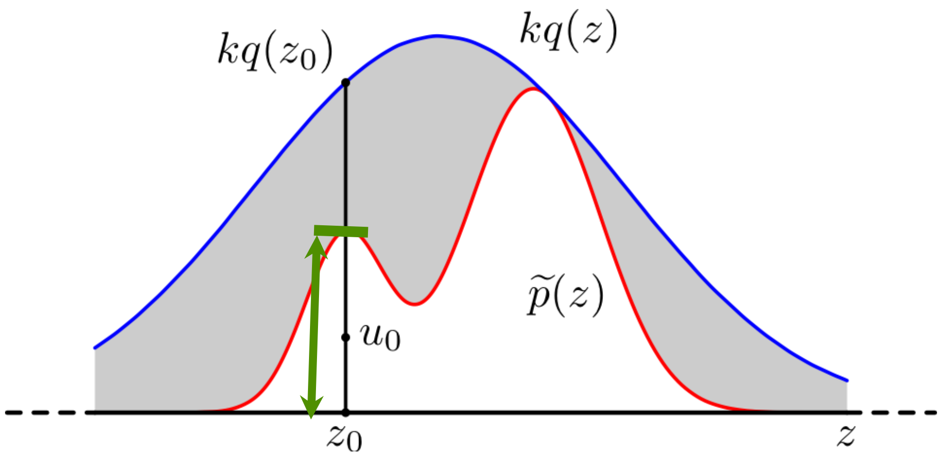
Cauchy (Student's-t) Proposal

Empirical variance of weights may not predict estimator variance!

- Always (asymptotically) unbiased, but variance of estimator can be enormous unless weight function bounded above:

$$\mathbb{E}_q[\hat{f}_L] = \mathbb{E}_p[f] \qquad \mathrm{Var}_q[\hat{f}_L] = \frac{1}{L}\mathrm{Var}_q[f(x)w(x)] \qquad w(x) = \frac{p(x)}{q(x)}$$

Monte Carlo Methods Summary



Rejection sampling

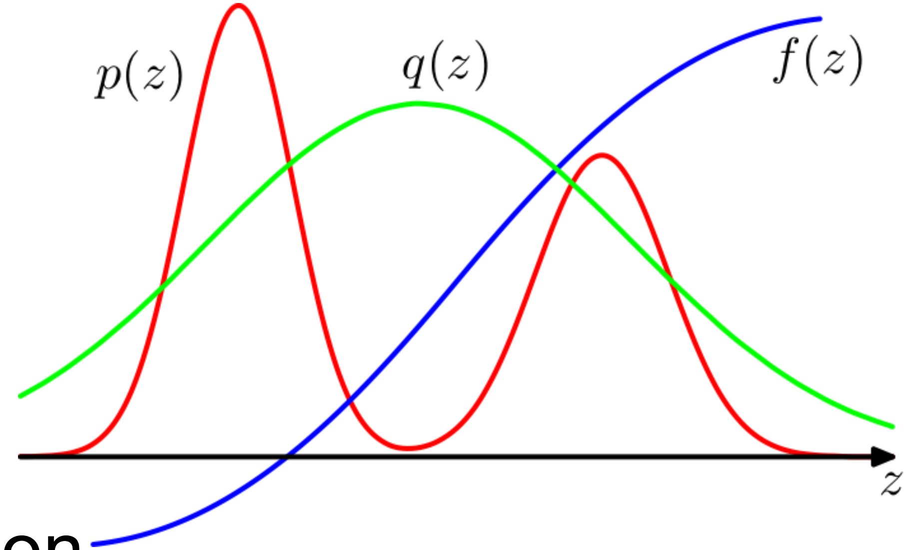
- Choose q such that: $\widetilde{p}(z) \leq k \cdot q(z)$
- Sample q(z) and keep with probability: $\frac{\widetilde{p}(z)}{k \cdot q(z)}$

Pro: Efficient, easy to implement

Con: Acceptance rate evaporates as dimension increases

Importance Sampling

$$\mathbf{E}_p[f(z)] \approx \sum_{l=1}^{L} \frac{\widetilde{r}^{(l)}}{\sum_{i=1}^{L} \widetilde{r}^{(i)}} f(z^{(l)}) \qquad \widetilde{r}^{(l)} = \frac{\widetilde{p}(z^{(l)})}{q(z^{(l)})}$$



Pro: Efficient, easy to implement

Con: Variance grows exponentially in dimension

Outline

- Monte Carlo Estimation

- Markov Chain Monte Carlo

See separate MCMC slides…

Monte Carlo Methods Summary

- Simulation: $x \sim p(x) = \frac{1}{Z} f(x)$ **Rejection sampling, MCMC**

- Compute expectations: $\mathbb{E}[\phi(x)] = \int p(x)\phi(x)\, dx$

  Importance sampling or any simulation method

- Optimization: $x^* = \arg\max_x f(x)$ **Simulated annealing**

- Compute normalizer / marginal likelihood: $Z = \int f(x)\, dx$

  Reverse importance sampling (Did not cover)

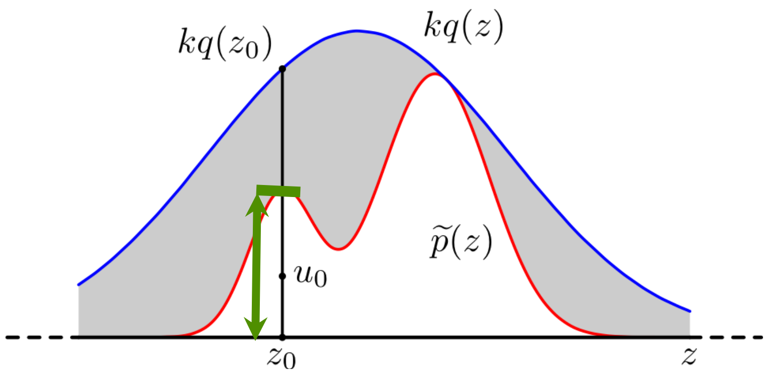
Monte Carlo Methods Summary

- In complex models we often have no other choice than to simulate realizations

- Rejection sampler choose proposal/constant s.t. $\widetilde{p}(z) \leq kq(z)$

1) Sample $q(z)$

2) Keep samples in proportion to $\dfrac{\tilde{p}(z)}{k \bullet q(z)}$ and reject the rest.



- Monte carlo estimate via independent samples $\{z^{(i)}\}_{i=1}^{L} \sim p$,

$$\mathbf{E}_p[f] \approx \frac{1}{L} \sum_{i=1}^{L} f(z^{(i)})$$

- Unbiased
- Consistent
- Law of large numbers
- Central limit theorem (if f is finite variance)

Monte Carlo Methods Summary

- Importance sampling estimate over samples $\{z^{(i)}\}_{i=1}^{L} \sim q$,

$$\mathbf{E}_p[f] \approx \sum_{i=1}^{L} w^{(i)} f(z^{(i)}) \qquad w^{(i)} \propto \frac{\widetilde{p}(z^{(i)})}{q(z^{(i)})}$$

Proposal

Importance Weights

- Avoids simulation of *p(z)* but variance scales exponentially with dim.
- Sequential importance sampling extends IS for sequence models, with proposal given by dynamics,

$$q(z) = q(z_0) \prod_{t=1}^{T} p(z_t \mid z_{t-1}) \qquad w_t(z^{(i)}) \propto w_{t-1}(z^{(i-1)}) p(y_t \mid z_t^{(i)})$$

"Bootstrap" Particle Filter

Recursively update weights

- **Resampling** step necessary to avoid weight degeneracy

Monte Carlo Methods Summary

- Lots of other methods to explore…
  - Hamiltonian Monte Carlo
  - Slice Sampling
  - Reversible Jump MCMC (and other *transdimensional samplers*)
  - Parallel Tempering

- Some good resources if you are interested…

Neal, R. “Probabilistic Inference Using Markov Chain Monte Carlo Methods”, U. Toronto, 1993

MacKay, D. J. “Introduction to Monte Carlo Methods”, Cambridge U., 1998

Andrieu, C., et al., “Introduction to MCMC for Machine Learning”, 2001