Stochastic approximate inference

Kay H. Brodersen

Computational Neuroeconomics Group
Department of Economics
University of Zurich

Machine Learning and Pattern Recognition Group
Department of Computer Science
ETH Zurich

http://people.inf.ethz.ch/bkay/

When do we need approximate inference?

- How to evaluate the posterior distribution of the model parameters?
$$p(\theta|\mathcal{Y}) = \frac{p(\mathcal{Y}|\theta)p(\theta)}{p(\mathcal{Y})} = \boxed{\frac{1}{Z}p(\mathcal{Y}|\theta)p(\theta)}$$
sample from an arbitrary distribution

- How to compute the evidence term?
$$p(\mathcal{Y}) = \int p(\mathcal{Y}|\theta)p(\theta)\, d\theta = \boxed{\mathrm{E}_\theta[p(\mathcal{Y}|\theta)]}$$
compute an expectation

- How to compute the expectation of the posterior?
$$\boxed{E[\theta|\mathcal{Y}]} = \int \theta\, p(\theta|\mathcal{Y}) d\theta$$
compute an expectation

- How to make a point prediction?
$$\int y\, p(y|\mathcal{Y})\, dy = \boxed{\mathrm{E}[y|\mathcal{Y}]}$$
compute an expectation

Which type of approximate inference?

Deterministic approximations through structural assumptions

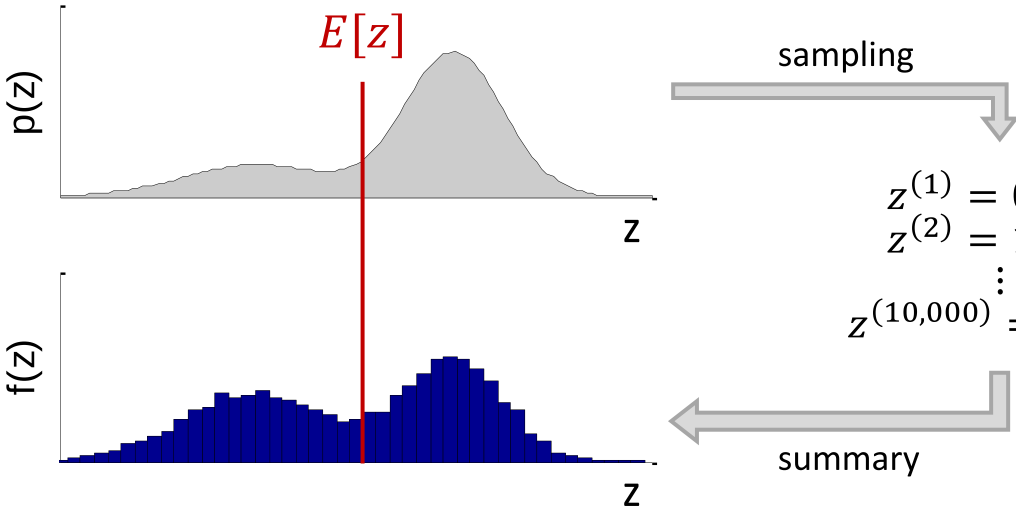
- ⊖ application often requires mathematical derivations (hard work)
- ⊖ systematic error
- ⊕ computationally efficient
- ⊕ efficient representation
- ⊕ learning rules may give additional insight

Stochastic approximations through sampling

- ⊖ computationally expensive
- ⊖ storage intensive
- ⊕ asymptotically exact
- ⊕ easily applicable general-purpose algorithms

Themes in stochastic approximate inference

Sampling from a desired target distribution
we need to find a way of drawing random numbers from some target distribution $p(z)$



$z^{(1)} = 0.201$
$z^{(2)} = 1.807$
$\vdots$
$z^{(10{,}000)} = 0.870$

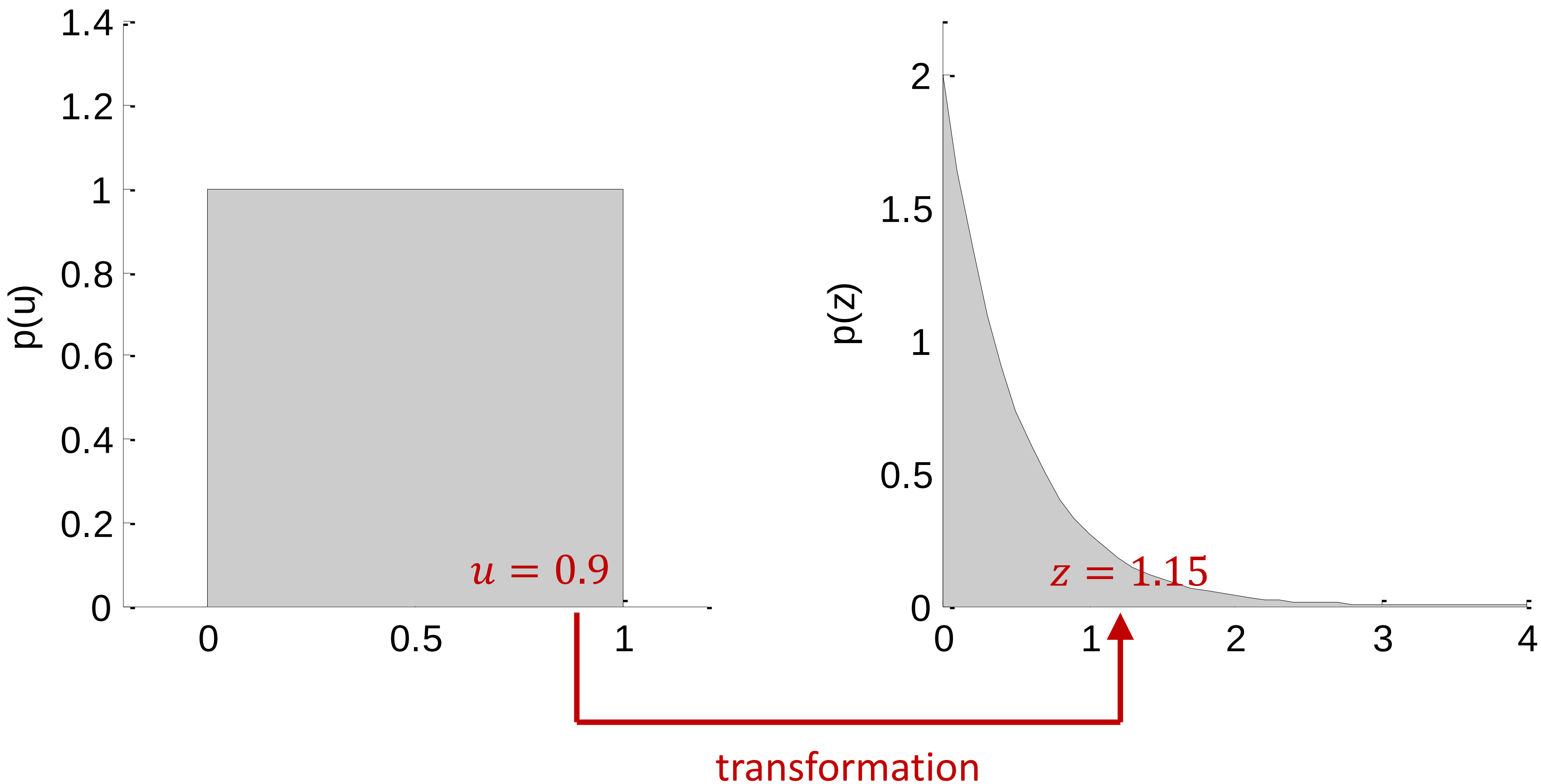
Computing an expectation w.r.t. that target distribution
we can approximate the expectation of z using the sample mean:
$E[z] \approx \frac{1}{T}\sum_{\tau=1}^{T} z^{(\tau)}$

1

Transformation method

Transformation method for sampling from $p(z)$

- Idea: we can obtain samples from some distribution $p(z)$ by first sampling from the uniform distribution and then *transforming* these samples.



Transformation method: algorithm

- Algorithm for sampling from $p(z)$
  - Draw a random number from the uniform distribution:
    $u^{(\tau)} \sim U(0,1)$
  - Transform u by applying the inverse cumulative density function (cdf) of the desired target distribution:
    $z^{(\tau)} = F^{-1}\left(u^{(\tau)}\right)$
  - Repeat both steps for $\tau = 1 \ldots T$.

Transformation method: example

- Example: sampling from the exponential distribution
  - The desired pdf is: $p(z|\lambda) = \lambda \exp(-\lambda z)$
  - The corresponding cdf is: $F(z) = 1 - \exp(-\lambda z)$
  - The inverse cdf is: $F^{-1}(u) = -\frac{1}{\lambda}\ln(1-u)$
  - Thus, $z^{(\tau)} = -\frac{1}{\lambda}\ln\left(1-u^{(\tau)}\right)$ is a sample from the exponential distribution.

- Implementation in MATLAB

```
for t=1:10000
    z(t) = -1/lambda*log(1-rand);
end
hist(z)
mean(z)
```

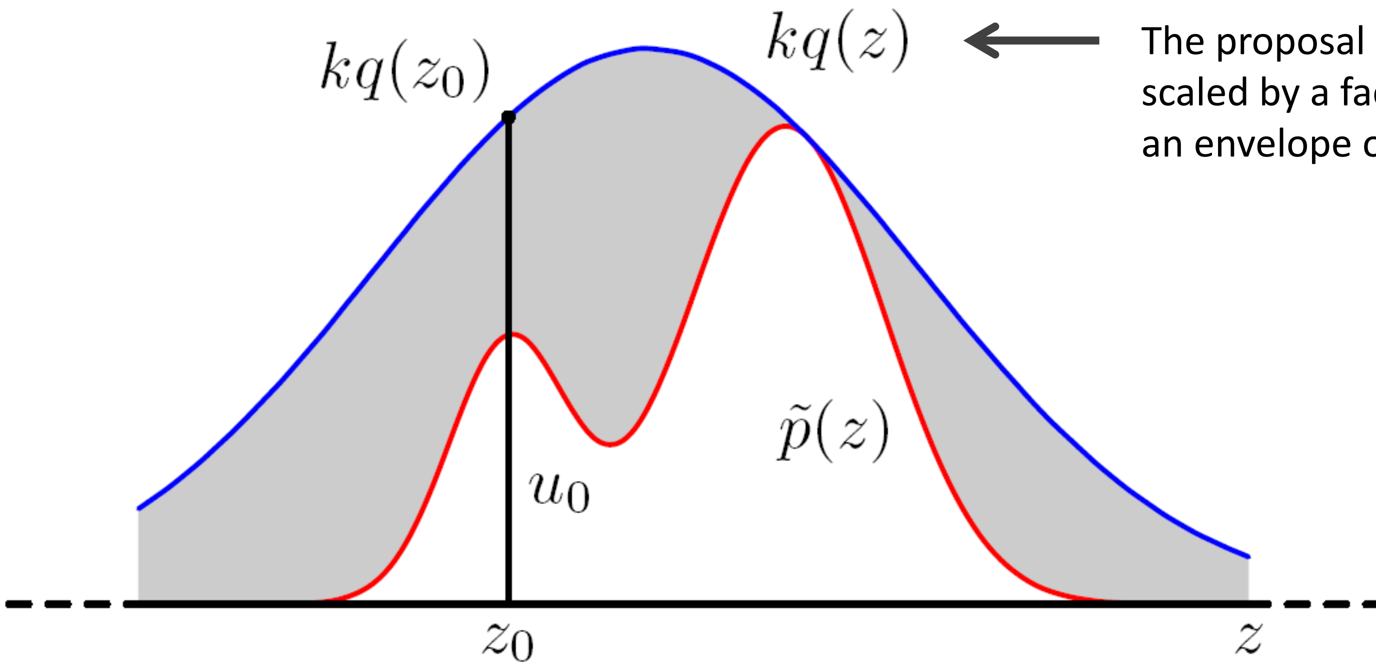
Transformation method: summary

- Discussion
  - ⊕ yields high-quality samples
  - ⊕ easy to implement
  - ⊕ computationally efficient
  - ⊖ obtaining the inverse cdf can be difficult

2 Rejection sampling and importance sampling

Rejection sampling

- Idea: when the transformation method cannot be applied, we can resort to a more general method called *rejection sampling*. Here, we draw random numbers from a simpler *proposal distribution* $q(z)$ and keep only some of these samples.



The proposal distribution $q(z)$, scaled by a factor k, represents an envelope of $p(z)$.

Rejection sampling: algorithm

- Algorithm for sampling from $p(z)$
  - Sample z_0 from $q(z)$
  - Sample u_0 from $U(0,1)$
  - If $u_0 \leq p(z_0)/kq(z_0)$, then accept the sample:
    $z^{(\tau)} = z_0$
  - Otherwise, discard z_0 and u_0.
  - Repeat until we have obtained T accepted samples.

Importance sampling

- Idea: if our goal is to compute the expectation $E[z]$, we can outperform rejection sampling by bypassing the generationg of random samples.

- Naïve approach
  - A naïve approach would be to approximate the expectation as follows. Rather than sampling from $p(z)$, we discretize z-space into a uniform grid and evaluate:

$$\mathbb{E}[z] \approx \sum_{l=1}^{L} p(z^{(l)})\, z^{(l)}$$

  - There are two problems with this approach:
    - The number of terms in the summation grows exponentially with the dimensionality of z.
    - Only a small proportion of the samples will make a significant contribution to the sum. Uniform sampling clearly is very inefficient.

Importance sampling

- Addressing the two problems of the naive approach above, given a proposal distribution $q(z)$, we can approximate the expectation as
$$\mathbb{E}[z] = \int z\, p(z) dz = \int z \frac{p(z)}{q(z)} q(z) dz \approx \frac{1}{L} \sum_{l=1}^{L} \frac{p\left(z^{(l)}\right)}{q\left(z^{(l)}\right)} z^{(l)}$$
where the samples $z^{(l)}$ are drawn from q.

- The quantities $r_l = \frac{p\left(z^{(l)}\right)}{q\left(z^{(l)}\right)}$ are known as *importance weights*, and they correct the bias introduced by sampling from the wrong distribution.

- Unlike in the case of rejection sampling, all of the generated samples are retained.

Markov Chain Monte Carlo

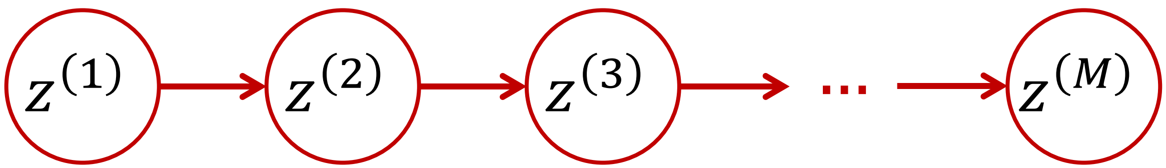
Markov Chain Monte Carlo (MCMC)

- Idea: we can sample from a large class of distributions and overcome the problems that previous methods face in high dimensions using a framework called *Markov Chain Monte Carlo*.

Background on Markov chains

- A first-order Markov chain is defined as a series of random variables $z^{(1)}, \dots, z^{(M)}$ such the following conditional-independence property holds:
  $p\left(z^{(m+1)} \middle| z^{(1)}, \dots, z^{(m)}\right) = p\left(z^{(m+1)} \middle| z^{(m)}\right)$

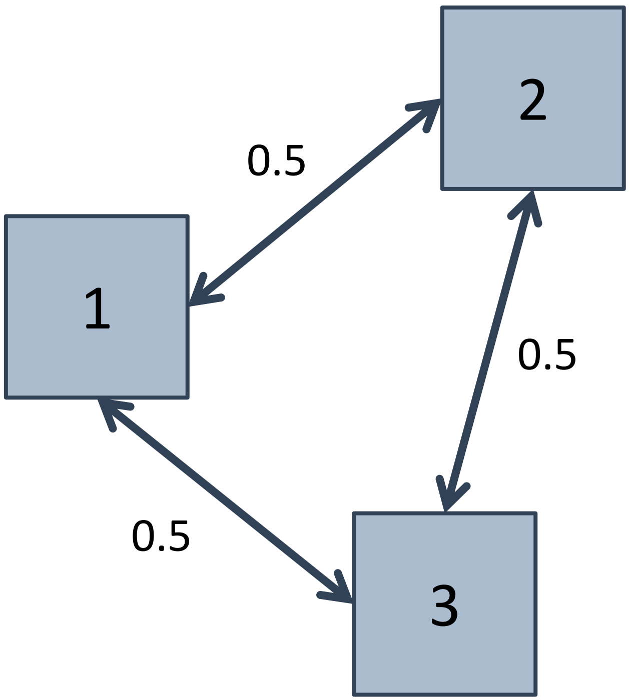
- Thus, the graphical model of a Markov chain is a chain:



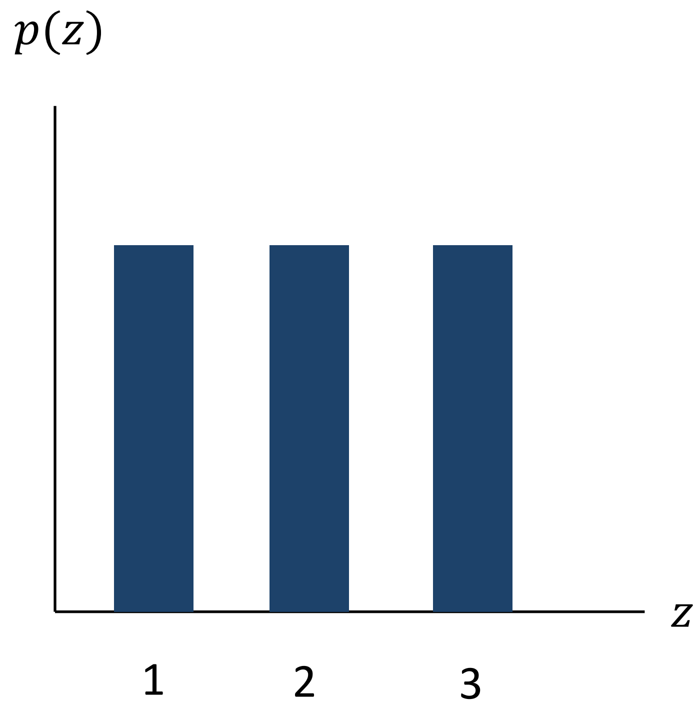
- A Markov chain is specified in terms of
  - the initial probability distribution $p\left(z^{(0)}\right)$
  - the transition probabilities $p\left(z^{(m+1)} \middle| z^{(m)}\right)$

Background on Markov chains

Markov chain: state diagram

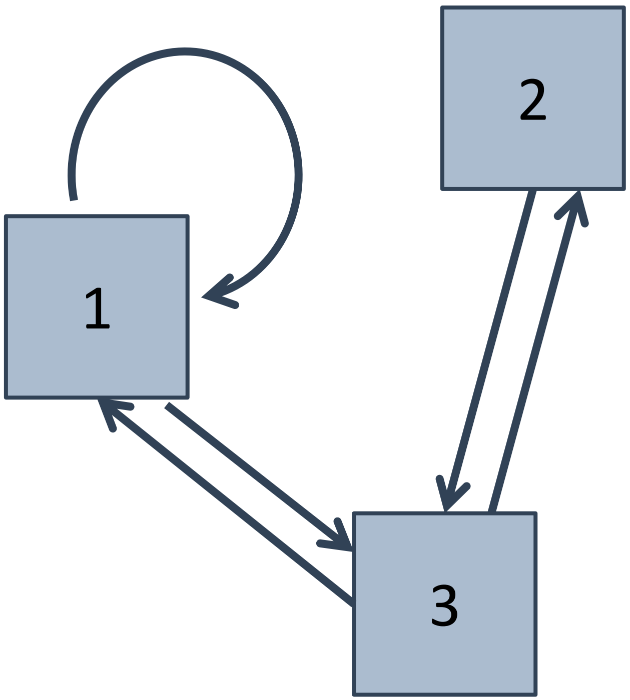


Equlibrium distribution

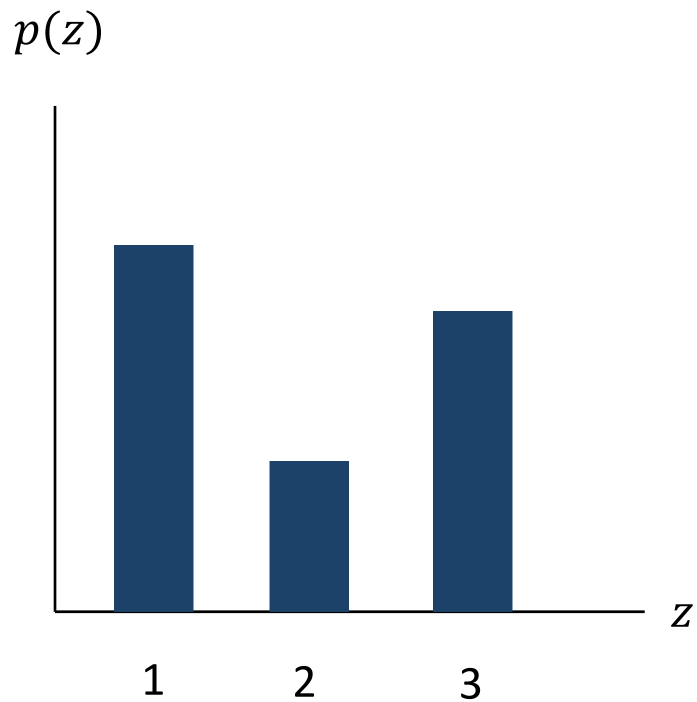


Background on Markov chains

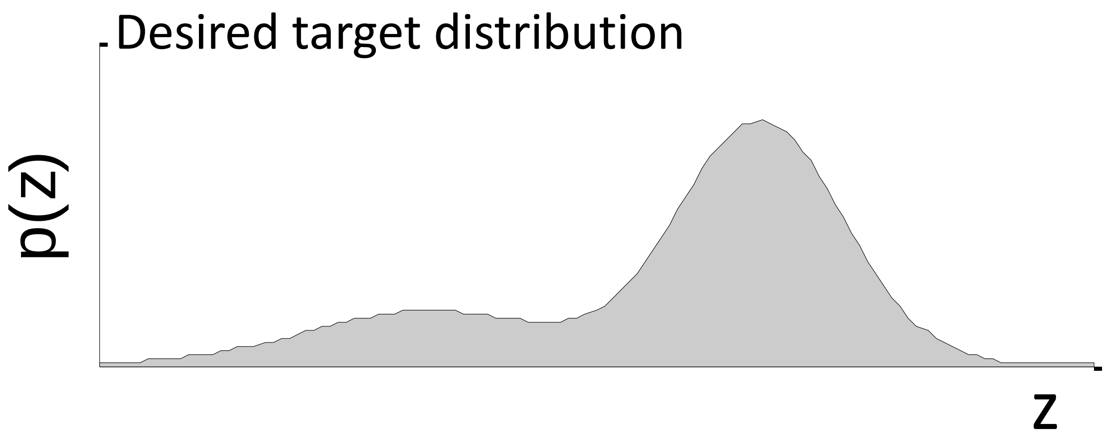
Markov chain: state diagram

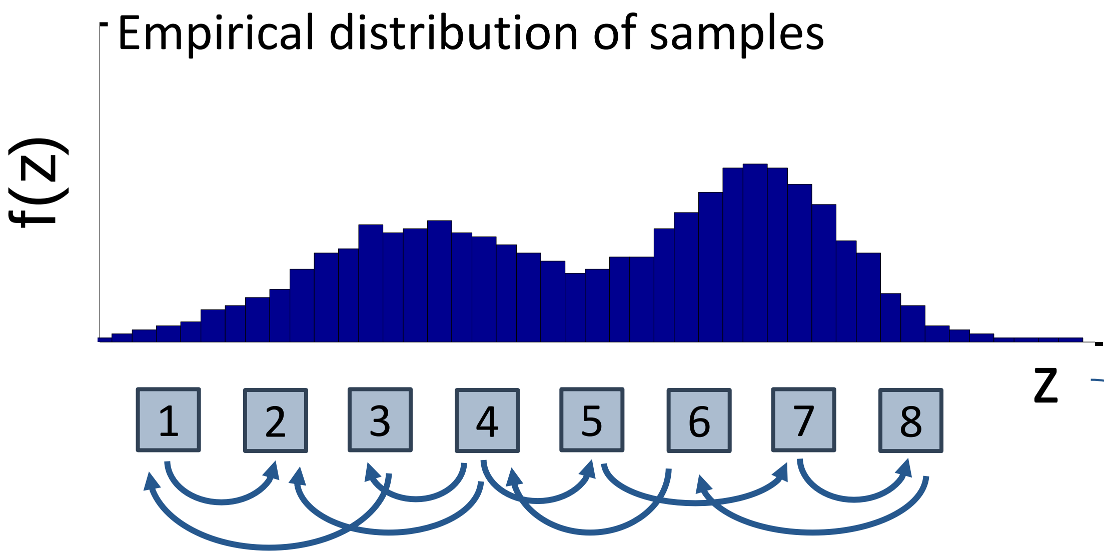


Equlibrium distribution



The idea behind MCMC





Markov chain whose equilibrium distribution is $p(z)$

Metropolis algorithm

- Algorithm for sampling from $p(z)$
  - Initialize by drawing $z^{(1)}$ somehow.
  - At cycle $\tau + 1$, draw a candidate sample z^* from $q\left(z \middle| z^{(\tau)}\right)$.
    Importantly, q needs to be symmetric, i.e., $q(z_1|z_2) = q(z_2|z_1)$.
  - Accept $z^{(\tau+1)} \leftarrow z^*$ with probability
    $A\left(z^*, z^{(\tau)}\right) = \min\left(1, \frac{p(z^*)}{p(z^{(\tau)})}\right) = \min\left(1, \frac{\tilde{p}(z^*)}{\tilde{p}(z^{(\tau)})}\right)$,
    and otherwise set $z^{(\tau+1)} \leftarrow z^{(\tau)}$.

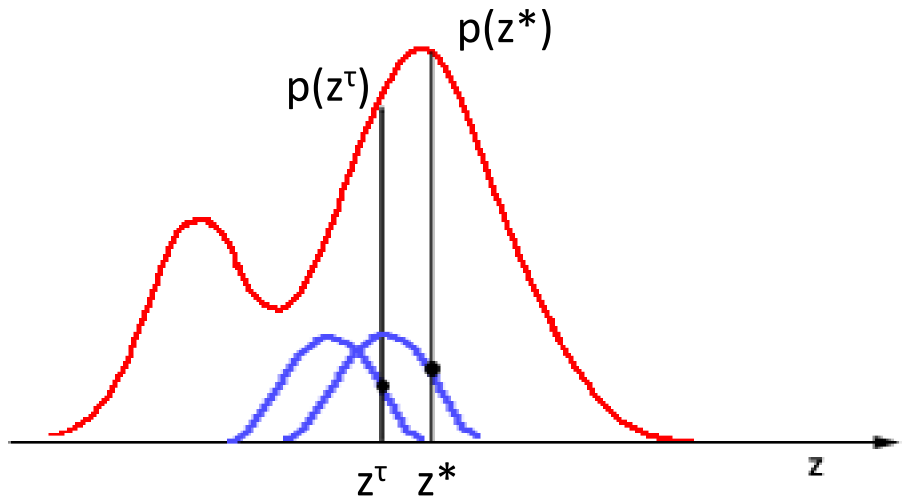
- Notes
  - In contrast to rejection sampling, each cycle leads to a new sample, even when the candidate z^* is discarded.
  - Note that the sequence $z^{(1)}, z^{(2)}, \ldots$ is not a set of independent samples from $p(z)$ because successive samples are highly correlated.

Metropolis-Hastings algorithm

- Algorithm for sampling from $p(z)$
  - Initialize by drawing $z^{(1)}$ somehow.
  - At cycle $\tau + 1$, draw a candidate sample z^* from $q(z|z^{(\tau)})$.
    In contrast to the Metropolis algorithm (see previous slide), q no longer needs to be symmetric.
  - Accept $z^{(\tau+1)} \leftarrow z^*$ with probability
    $A(z^*, z^{(\tau)}) = \min\left(1, \frac{\tilde{p}(z^*)q_k(z^{(\tau)}|z^*)}{\tilde{p}(z^{(\tau)})q_k(z^*|z^{(\tau)})}\right)$,
    and otherwise set $z^{(\tau+1)} \leftarrow z^{(\tau)}$.

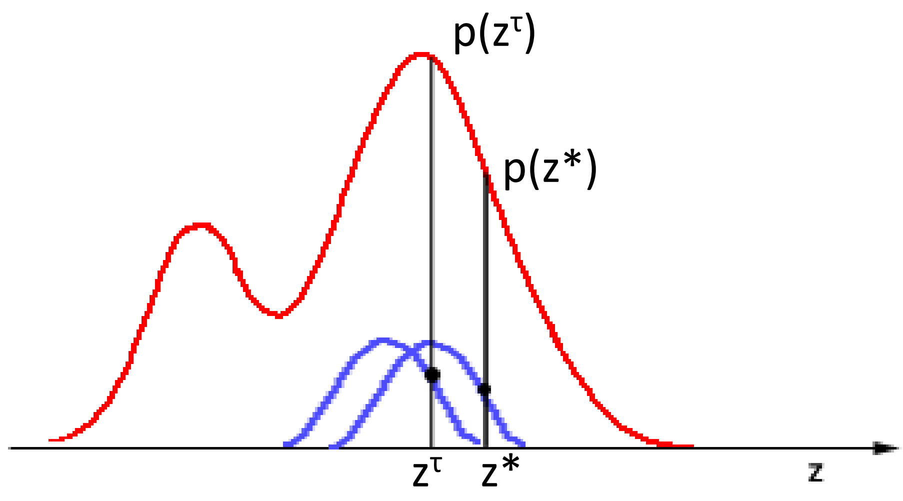
Metropolis: accept or reject?

Inrease in density:



$$\alpha = 1$$

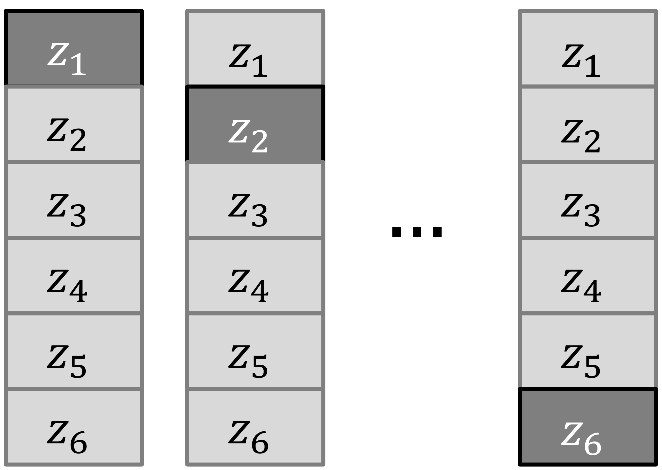
Decrease in density:



$$\alpha = \frac{\widetilde{p}(z^*)}{\widetilde{p}(z^\tau)}$$

Gibbs sampling

- Idea: as an alternative to the Metropolis-Hastings algorithm, *Gibbs sampling* is less broadly applicable but does away with acceptance tests and can therefore be more efficient.
- Suppose we wish to sample from a multivariate distribution $p(z) = p(z_1, \ldots, z_M)$, e.g., representing several variables in a model. For example, we might be interested in their joint posterior distribution.
- In Gibbs sampling, we update one component at a time.



Gibbs sampling

- Algorithm for sampling from $p(z)$
  - Initialize $\{z_i: i = 1, \dots, M\}$ somehow.
  - At cycle $\tau + 1$, sample $z_i^{(\tau)} \sim p\left(z_i \middle| z_{\setminus i}^{(\tau)}\right)$, i.e., replace the i^{th} variable by a new sample, drawn from a distribution that is conditioned of the current values of all other variables. The resulting new vector is our new sample.
  - In the next cycle, replace a different variable i. The simplest procedure is to go round $i = 1, \dots, M, 1, \dots, M, \dots$ Alternatively, i could be chosen randomly.

Summary

- Throughout Bayesian statistics, we encounter intractable problems. Most of these problems are: (i) evaluating a distribution; or (ii) computing the expectation of a distribution.
- Sampling methods provide a stochastic alternative to deterministic methods. They are usually computationally less efficient, but are asymptotically correct, broadly applicable, and easy to implement.
- We looked at three main approaches:
  - Transformation method: efficient sampling from simple distributions
  - Rejection sampling and importance sampling: sampling from arbitrary distributions; direct computation of an expected value
  - Monte Carlo Markov Chain (MCMC): efficient sampling from high-dimensional distributions through the Metropolis-Hastings algorithm or Gibbs sampling