Pattern Recognition and Machine Learning
Chapter 11: Sampling Methods

Elise Arnaud Jakob Verbeek

May 22, 2008

Outline of the chapter

Goal of sampling methods

- fundamental problem: find the expectation of some function $f(\mathbf{z})$ with respect to a probability distribution $p(\mathbf{z})$

$$\mathbb{E}[f] = \int f(\mathbf{z})\, p(\mathbf{z})\, d\mathbf{z} \tag{11.1}$$

- idea: If we obtain a set of samples $\mathbf{z}^{(l)}, l = 1 \ldots L$ drawn independantly from $p(\mathbf{z})$, the expectation may be approximated by :

$$\hat{f} = \frac{1}{L} \sum_{l=1}^{L} f(\mathbf{z}^{(l)}) \tag{11.2}$$

- new problem: How can we obtain independant samples from a distribution $p(\mathbf{z})$ we do not know how to sample from ?

Outline

11.1.1 Standard distributions: the transformation method

- We aim at sampling from $p(.)$.

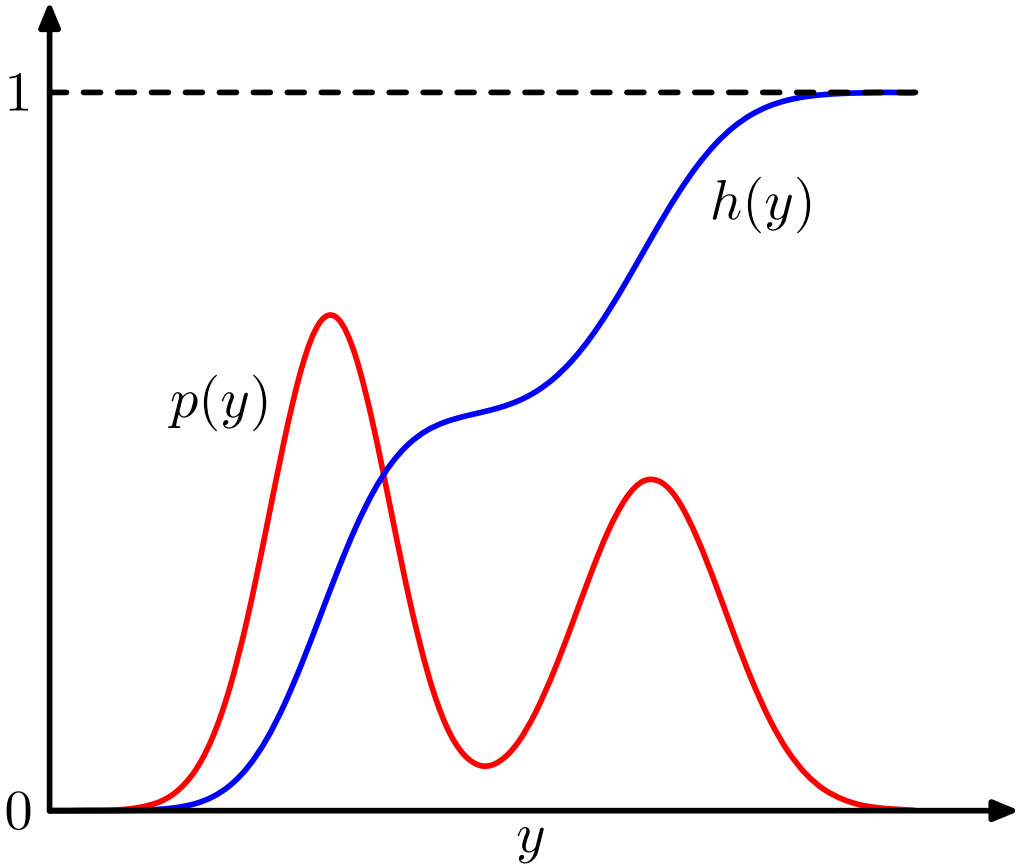
- Suppose that we have available samples z uniformly distributed over the interval $(0, 1)$
- Transforming z into y using $y = h^{-1}(z)$ where h is defined as the cumulative distribution fuction of p:

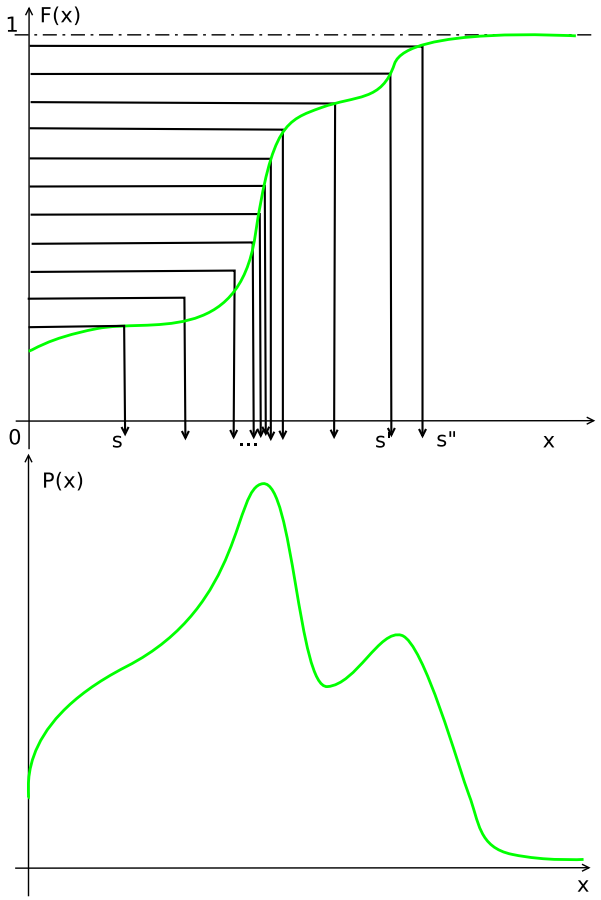
$$h(y) = \int_{-\infty}^{y} p(x)\, dx \qquad \text{(11.6 modified)}$$

- Then y are independant samples from $p(.)$.

- ex: exponential distribution, Cauchy distribution, Gaussian distribution

11.1.1 Standard distributions: the transformation method





11.1.2 Rejection sampling

- Assumption 1: Sampling directly from $p(\mathbf{z})$ is difficult but we are able te evaluate $p(\mathbf{z})$ for any value of $\mathbf{z}$ up to a unknown normalizing constant Z_p

$$p(z) = \frac{1}{Z_p}\,\tilde{p}(z) \tag{11.13}$$

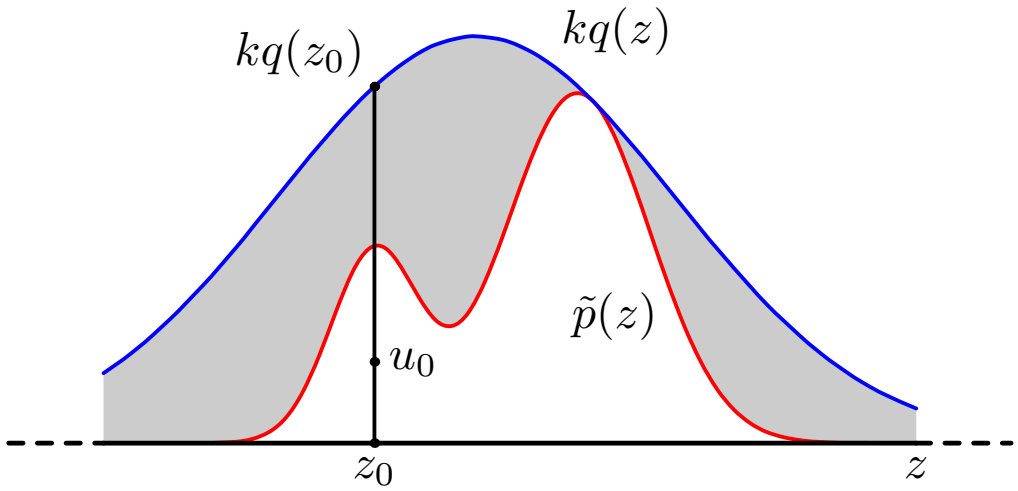
- Assumption 2: We know how to sample from a proposal distribution $q(z)$, and there exists a constant k such that

$$k\; q(z) \geq \tilde{p}(z)$$

- Then we know how to obtain independant samples from $p(.)$

11.1.2 Rejection sampling

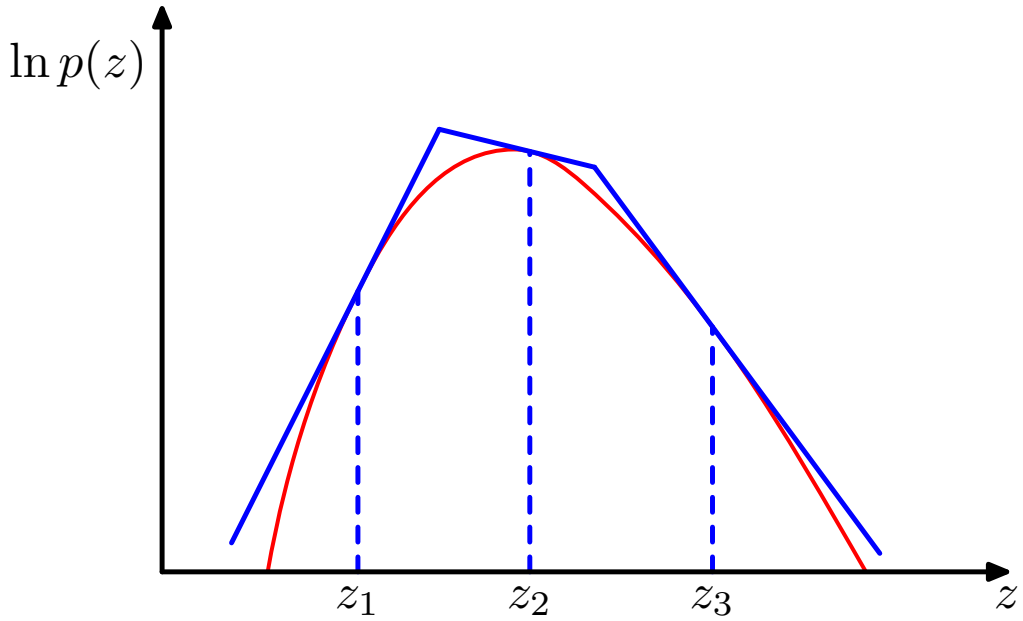
1. generate a number z_0 from $q(.)$
2. generate a number u_0 from the uniform distribution over $[0, kq(z_0)]$
3. if $u_0 > \tilde{p}(z_0)$ then the sample is rejected, otherwise z_o is kept
4. The set of kept z are distributed according to $p(.)$



- efficiency of the method depend on the ratio between the grey area and the white area $\longrightarrow$ the proposal distribution $q(.)$ as to be as close as possible from $p(.)$

11.1.3 Adaptive Rejection sampling

- the proposal distribution $q(.)$ may be constructed on the fly
- eg, if p is log-concave, use of the derivative of $\ln p$ at some given grid points



- in any cases, rejection sampling methods are inefficient if sampling in high dimension (exponential decrease of acceptance rate with dimensionality)

11.1.4 Importance sampling

- provides a framework for approximating expectations directly
- technique again based on the use of a proposal distibution $q(.)$

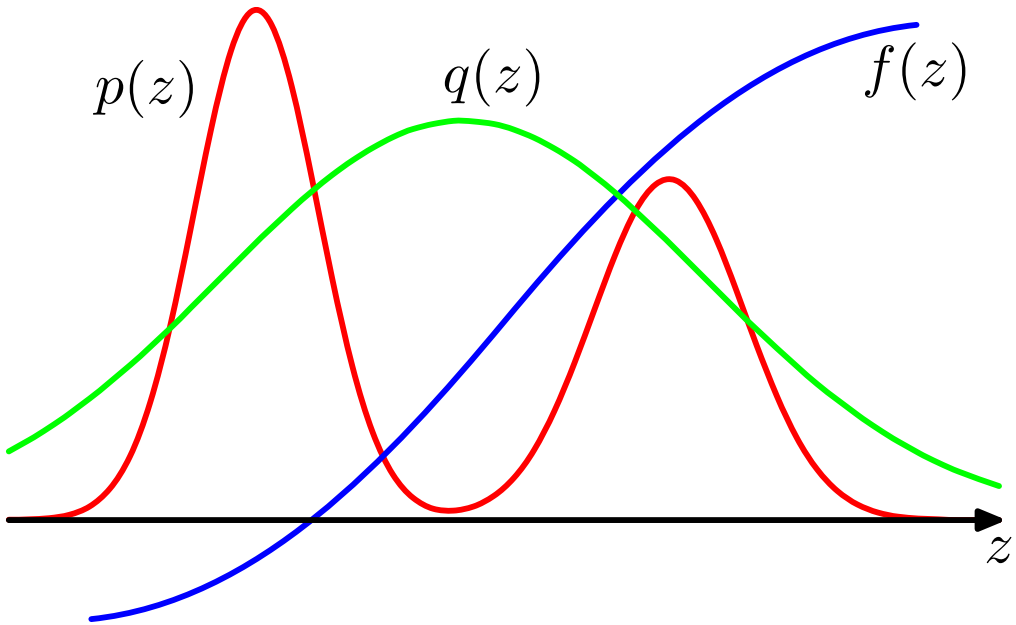
The trick:

$$\mathbb{E}[f] = \int f(\mathbf{z})p(\mathbf{z})d\mathbf{z} = \int f(\mathbf{z}) \frac{p(\mathbf{z})}{q(\mathbf{z})} q(\mathbf{z})d\mathbf{z} \approx \frac{1}{L} \sum_{l=1}^{L} \frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})} f(\mathbf{z}^{(l)}) \tag{11.19}$$

$$\mathbb{E}[f] \approx \sum_{l=1}^{L} w_l f(\mathbf{z}^{(l)}) \tag{11.22}$$

- The importance weights correct the bias introduced by sampling from a wrong distribution. w_l are the normalized imp. weights.
- all the generated samples are retained

11.1.4 Importance sampling



Again, the success of the method depend on how well the proprosal $q(.)$ fit the desired distribution $p(.)$. In particular, $p(z) > 0 \Rightarrow q(z) > 0$

11.1.5 Sampling-importance-resampling

- importance sampling is an alternative to rejection sampling
- technique based on the use of a proposal distibution $q(.)$
  the assumption on the existence of a constant k is relaxed

1. Draw L samples $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots \mathbf{z}^{(L)}$ from $q(\mathbf{z})$
2. Calculate the importance weight $\frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})} \quad \forall l = 1 \ldots L$
3. Normalize the weights to obtain $w_1 \ldots w_L$
4. Draw a second set of L samples from the discrete distribution $(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots \mathbf{z}^{(L)})$ with probabilities $(w_1 \ldots w_L)$

- The resulting L samples are distibuted according to $p(\mathbf{z})$ if $L \rightarrow \infty$

11.1.6 Sampling and the EM algorithm

Monte Carlo EM algorithm

- Use some Monte Carlo methods to approximate the expectation of the E step
- The expected complete-data log likelihood, given by (**Z** hidden ; **X** observed ; θ parameters):

$$Q(\theta, \theta^{old}) = \int p(\mathbf{Z}|\mathbf{X}, \theta^{old}) \ln p(\mathbf{Z}, \mathbf{X}|\theta) \, d\mathbf{Z} \tag{11.28}$$

may be approximate by (where $\mathbf{Z}^{(l)}$ are drawn from $p(\mathbf{Z}, \mathbf{X}|\theta^{old})$)

$$Q(\theta, \theta^{old}) \approx \frac{1}{L} \sum_{l=1}^{L} \ln p(\mathbf{Z}^{(l)}, \mathbf{X}|\theta), \tag{11.29}$$

stochastic EM algorithm

- Considering a finite mixture model, only one sample **Z** may be drawn at each E step (meaning a hard assignement of each data point to one of the components in the mixture)

Outline

Markov Chain Monte Carlo (MCMC)

- MCMC : general strategy which allows sampling from a large class of distributions
- MCMC scales well with the dimensionality of the sample space $\neq$ importance sampling / rejection sampling
- use the mechanism of Markov chains

- goal: to generate a set of samples from $p(\mathbf{z})$
- assumption: we know how to evaluate $\tilde{p}(\mathbf{z})$

$$\tilde{p}(\mathbf{z}) = Z_p \; p(\mathbf{z})$$

MCMC: the idea

- goal: to generate a set of samples from $p(\mathbf{z})$
- idea: to generate samples from a Markov Chain whose invariant distribution is $p(\mathbf{z})$

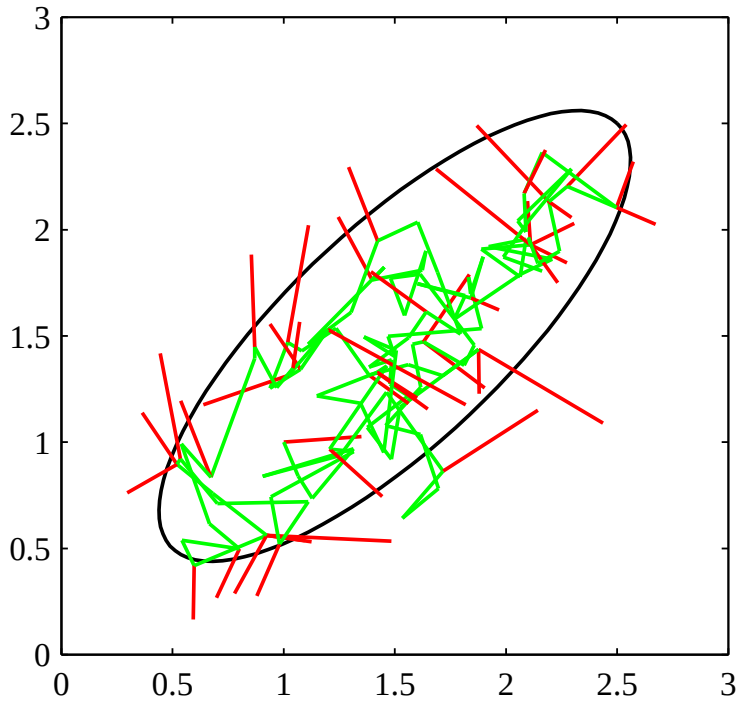
1. knowing the current sample is $\mathbf{z}^{(\tau)}$, generate a candidate sample $\mathbf{z}^*$ from a proposal distribution $q(\mathbf{z}|\mathbf{z}^{(\tau)})$ we know how to sample from.
2. accept the sample according to an appropriate criterion.
3. if the candidate sample is accepted then $\mathbf{z}^{(\tau+1)} = \mathbf{z}^*$ otherwise $\mathbf{z}^{(\tau+1)} = \mathbf{z}^{(\tau)}$.

- the proposal distribution depends on the current state
- samples $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots$ form a Markov chain

Metropolis algorithm

- the proposal distibution is symetric $q(\mathbf{z}_A|\mathbf{z}_B) = q(\mathbf{z}_B|\mathbf{z}_A)$
- the candidate sample is accepted with probability

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})}\right) \tag{11.33}$$



Use of Metropolis algorithm to sample from a Gaussian distribution. The proposal distribution is an isotropic Gaussian whose $\sigma = 0.2$. Accepted steps in green, rejected steps in red. 150 candidate samples, 43 rejected.

11.2.1 Why is it working ? Markov chains

- idea: to generate samples from a Markov Chain whose invariant distribution is $p(\mathbf{z})$
- why is it working ? under what circumstances will a Markov chain converge to the desired distribution ?

  - first order Markov chain: series of random variables $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(M)}$ such that

$$p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}) = p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)}) \quad \forall m \qquad (11.37)$$

  - Markov chain specified by $p(\mathbf{z}^{(0)})$ and the *transition probabilities*

$$T_m(\mathbf{z}^{(m)}, \mathbf{z}^{(m+1)}) \equiv p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)})$$

11.2.1 Why is it working ? Markov chains

- a Markov chain is called *homogeneous* is the transition probabilities are the same for all m
- a distribution $p^*(\mathbf{z})$ is said to be *invariant* w.r.t a Markov chain if each transition leaves the distribution invariant

$$p^*(\mathbf{z}) = \sum_{\mathbf{z}'} T(\mathbf{z}', \mathbf{z})\ p^*(\mathbf{z}') \tag{11.39}$$

- a sufficient condition for ensuring $p^*(\mathbf{z})$ to be invariant is to choose the transitions to satisfy the property of *detailed balance* defined by

$$p^*(\mathbf{z})\ T(\mathbf{z}, \mathbf{z}') = T(\mathbf{z}', \mathbf{z})\ p^*(\mathbf{z}') \tag{11.40}$$

- a Markov chain that respect the detailed balance is said to be *reversible*
- a Markov chain is said *ergodic* if it converges to the invariant distribution irrespective of the choice of the initial distribution

11.2.1 Why is it working ? Markov chains

- goal: to generate a set of samples from $p(\mathbf{z})$
- idea: to generate samples from a Markov Chain whose invariant distribution is $p(\mathbf{z})$
- how: choose the transition probability $T(\mathbf{z}, \mathbf{z}^*)$ to satisfy the property of *detailed balance* for $p(\mathbf{z})$

- remark : $T(\mathbf{z}, \mathbf{z}^*)$ can be a mixture distribution

11.2.2 The Metropolis-Hasting algorithm

- generalization of the Metropolis algorithm.
- the proposal distribution q is no longer symetric

- knowing the current sample is $\mathbf{z}^{(\tau)}$, generate a candidate sample $\mathbf{z}^*$ from a proposal distribution $q(\mathbf{z}|\mathbf{z}^{(\tau)})$,
- accept it with probability:

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)\ q(\mathbf{z}^{(\tau)}|\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})\ q(\mathbf{z}^*|\mathbf{z}^{(\tau)})}\right) \qquad (11.44)$$

11.2.2 The Metropolis-Hasting algorithm

- Transition probability of this chain: $T(\mathbf{z}, \mathbf{z}') = q(\mathbf{z}'|\mathbf{z})\ A(\mathbf{z}', \mathbf{z})$
- to proove that $p(\mathbf{z})$ is the invariant distribution of the chain, It is sufficient to proove the property of detailled balance

$$p(\mathbf{z})\ T(\mathbf{z}, \mathbf{z}') = T(\mathbf{z}', \mathbf{z})\ p(\mathbf{z}')$$

11.2.2 The Metropolis-Hasting algorithm

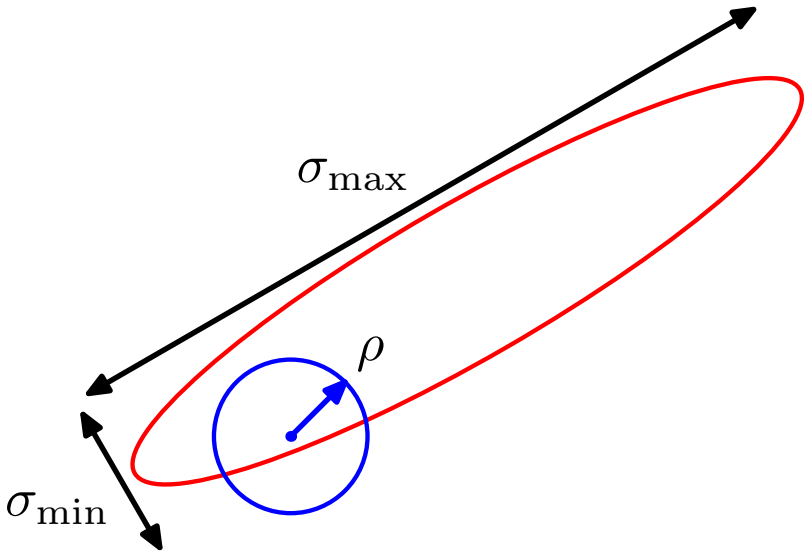
- Transition probability of this chain: $T(\mathbf{z}, \mathbf{z}') = q(\mathbf{z}'|\mathbf{z})\ A(\mathbf{z}', \mathbf{z})$
- to proove that $p(\mathbf{z})$ is the invariant distribution of the chain, It is sufficient to proove the property of detailled balance

$$p(\mathbf{z})\ T(\mathbf{z}, \mathbf{z}') = T(\mathbf{z}', \mathbf{z})\ p(\mathbf{z}')$$

$$\begin{aligned}
& p(\mathbf{z})\ q(\mathbf{z}'|\mathbf{z})\ A(\mathbf{z}', \mathbf{z}) \\
&= \min\left\{p(\mathbf{z})\ q(\mathbf{z}'|\mathbf{z})\ ;\ p(\mathbf{z}')\ q(\mathbf{z}|\mathbf{z}')\right\} \\
&= \min\left\{p(\mathbf{z}')\ q(\mathbf{z}|\mathbf{z}')\ ;\ p(\mathbf{z})\ q(\mathbf{z}'|\mathbf{z})\right\} \\
&= p(\mathbf{z}')\ q(\mathbf{z}|\mathbf{z}')\ \min\left\{1; \frac{p(\mathbf{z})\ q(\mathbf{z}'|\mathbf{z})}{p(\mathbf{z}')\ q(\mathbf{z}|\mathbf{z}')}\right\} \\
&= p(\mathbf{z}')\ q(\mathbf{z}|\mathbf{z}')\ A(\mathbf{z}, \mathbf{z}') \qquad \text{(11.45 modified)}
\end{aligned}$$

11.2.2 The Metropolis-Hasting algorithm

- common choice for q: Gaussian centered on the current state
  - small variance $\longrightarrow$ high rate of acceptation but slow exploration of the state space + non independant samples
  - large variance $\longrightarrow$ high rate of rejection



Use of an isotropic Gaussian proposal (blue circle), to sample from a Gaussian distribution (red). The scale ρ of the proposal should be on the order of σ_{min}, but the algorithm may have low convergence (low to explore the state space in the other direction)

Outline

Gibbs sampling

- Special case of the Metropolis-Hasting algorithm.
- objectif law : $p(\mathbf{z}) = p(z_1, \dots z_M)$
- $p(z_i|\mathbf{z}_{\setminus i})$ known

- Each step of the Gibbs sampling procedure involve replacing the value of one of the variables z_i by a value drawn from the distribution of that variable conditioned on the values of the remaining variables $p(z_i|\mathbf{z}_{\setminus i})$

- procedure repeated either by cycling through the variables in some order, or by choosing the variable to be updated at each step from some distribution

Gibbs sampling - example

- The objectif law is $p(z_1^{(i)}, z_2^{(i)}, z_3^{(i)})$
- at step i, we have selected values $z_1^{(i)}, z_2^{(i)}, z_3^{(i)}$.
- then we obtain $z_1^{(i+1)}, z_2^{(i+1)}, z_3^{(i+1)}$ with:

$$z_1^{(i+1)} \sim p(z_1 | z_2^{(i)}, z_3^{(i)}) \tag{11.46}$$

$$z_2^{(i+1)} \sim p(z_2 | z_1^{(i+1)}, z_3^{(i)}) \tag{11.47}$$

$$z_3^{(i+1)} \sim p(z_3 | z_1^{(i+1)}, z_2^{(i+1)}) \tag{11.48}$$

- If, instead of drawing a sample from the conditional distribution, we replace the variable by the maximum of the conditional distribution, we obtain the iterated conditional modes (ICM) algorithm.

Gibbs sampling

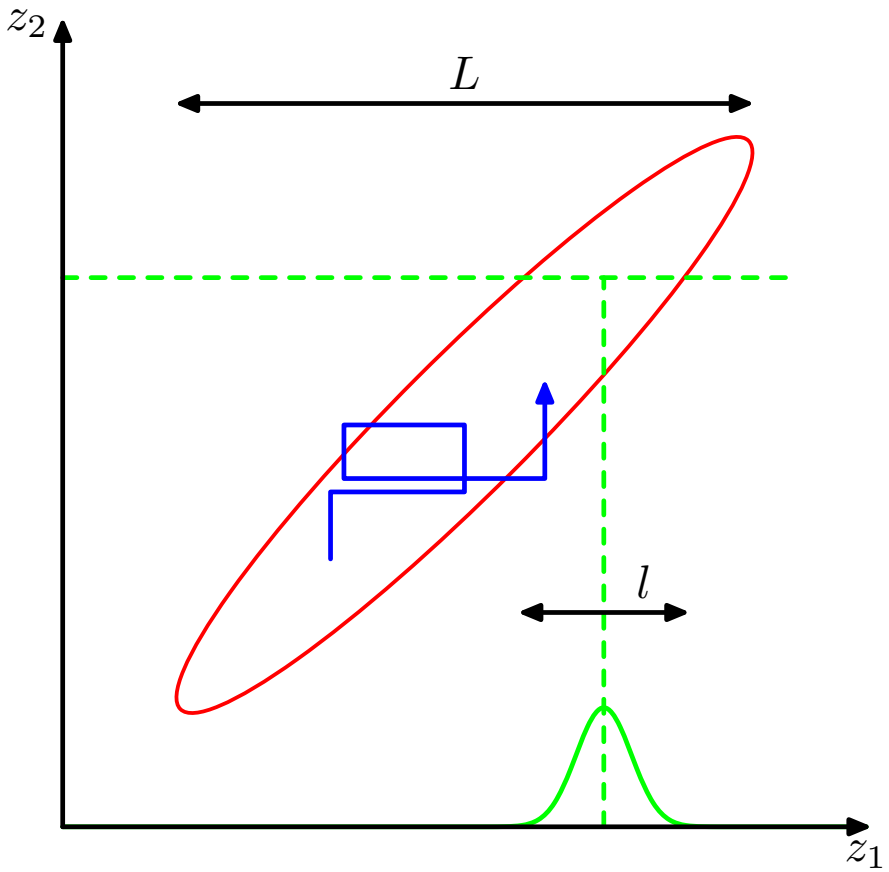
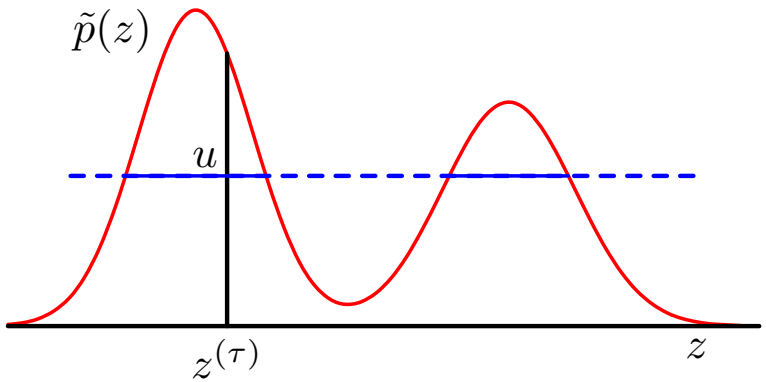


Illustration of Gibbs sampling, by alternate updates of two variables (blue steps) whose distribution is a correlated Gaussian (red). The conditional distributions are Gaussian (green curve).

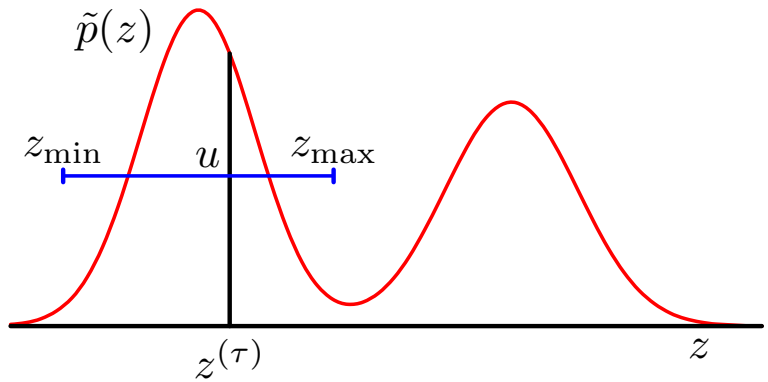
Outline

11.4: Slice Sampling (1/2)

- **Problem** of Metropolis algorithm (proposal $q(z|z') = q(z'|z)$)
  - step size too small, slow convergence (random walk behavior)
  - step size too large, high estimator variance (high rejection rate)
- **Idea:** adapt step size automatically to suitable value
- **Technique:** introduce variable u and sample (u, z) jointly.
  - Ignoring u leads to the desired samples of $p(z)$



(a)



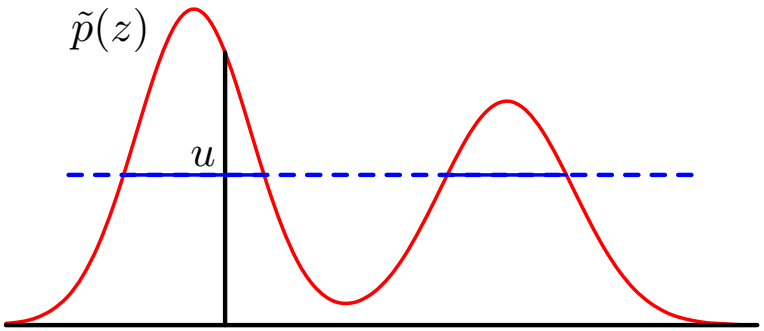
(b)

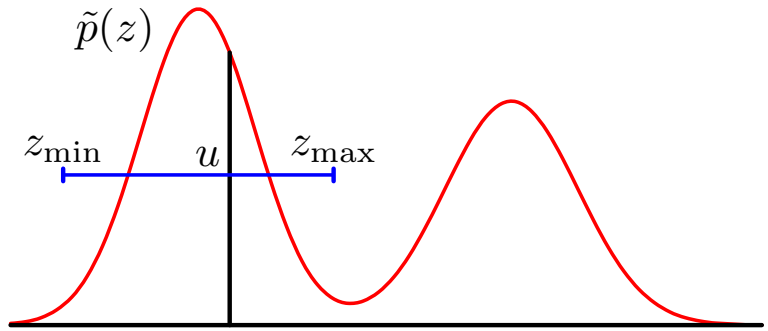
11.4: Slice Sampling (2/2)

- Sample z and u uniformly from area under the distribution
  - fix z, sample u uniform from $[0, \tilde{p}(z)]$
  - fix u, sample z uniform from **slice** : $\{z : \tilde{p}(z) = u\}$

- How to sample z from the slice?! [Neal, 2003]
  - Start with region of width w containing $z^{(\tau)}$
  - If end point in slice, then extend region by w in that direction
  - Sample z' uniform from region
  - If z' in slice, then accept as $z^{(\tau+1)}$,
  - If not: make z' new end point of the region, and resample z'

- Multivariate distributions: slice sampling within Gibbs sampler





Outline

11.5 Hybrid Monte Carlo

- Problem of Metropolis algorithm is the step size trade-off.
- Hybrid Monte Carlo suitable in continuous state spaces
  - able to make large jumps in state space
  - low rejection rate
  - based on dynamical systems
  - need to evaluate gradient of log-prob. w.r.t. state $\mathbf{z}$
  - based on *Hamiltonian Dynamics*
- Goal is to sample from

$$p(\mathbf{z}) = \frac{1}{Z_p} \exp(-E(\mathbf{z})) \tag{11.54}$$

  where $E(\mathbf{z})$ is interpreted as potential energy of system in $\mathbf{z}$

- Hamiltonian dynamical system over energy landscape $E(\mathbf{z})$

11.5 Hybrid Monte Carlo: Hamiltonian Dynamics

- Evolution of state variable $\mathbf{z} = \{z_i\}$ under continuous time τ
- Momentum variables correspond to rate of change of state

$$r_i = \frac{\mathrm{d}z_i}{\mathrm{d}\tau} \tag{11.53}$$

- Joint $(\mathbf{z}, \mathbf{r})$ space is called *phase space*
- Rate of change of momentum (acceleration, applied force):

$$\frac{\mathrm{d}r_i}{\mathrm{d}\tau} = -\frac{\mathrm{d}E(\mathbf{z})}{\mathrm{d}z_i} \tag{11.55}$$

- Hamiltonian is constant under dynamical system evolution

$$H(\mathbf{z}, \mathbf{r}) = E(\mathbf{z}) + K(\mathbf{r}), \qquad \text{with} \quad K(\mathbf{r}) = \frac{1}{2}\sum_i r_i^2$$

11.5 Hybrid Monte Carlo: Distribution over Phase Space

- Let Hamiltonian define a distribution over phase space:

$$p_H(\mathbf{z}, \mathbf{r}) = \frac{1}{Z_H} \exp(-H(\mathbf{z}, \mathbf{r})) = \frac{1}{Z_H} \exp(-E(\mathbf{z}) - K(\mathbf{r})).$$

  - Sate $\mathbf{z}$ and momentum $\mathbf{r}$ are independently distributed.

- Hamiltonian dynamics leave this distribution invariant!
  - If $(\mathbf{z}, \mathbf{r}) \sim p_H$, and evolve in time τ to $(\mathbf{z}^\star, \mathbf{r}^\star)$, then also $(\mathbf{z}^\star, \mathbf{r}^\star) \sim p_H$
  - volume and H are constant under Hamiltonian dynamics

- Hamiltonian evolution is not an *ergodic* sampler of p_H
  - Resampling of $\mathbf{r}$ using $p_H(\mathbf{r}|\mathbf{z}) = p_H(\mathbf{r})$
  - Gibbs sampling step

11.5 Hybrid Monte Carlo

- Combination of Metropolis algorithm & Hamitonian Dynamics
- Markov chain that alternates
  - stochastic update of the momentum variables $\mathbf{r}$
  - Hamiltonian dynamical updates with acceptance probability

$$\min(1, \exp\{H(\mathbf{z}, \mathbf{r}) - H(\mathbf{z}^\star, \mathbf{r}^\star)\}) \tag{11.67}$$

  to compensate for numerical errors: if $H(\mathbf{z}, \mathbf{r}) \neq H(\mathbf{z}^\star, \mathbf{r}^\star)$

- Direction of time is chosen randomly to have detailed balance
- Hybrid Monte Carlo generates independent samples faster.
  - the use of gradient information causes this difference

Outline

11.6 Estimating the Partition Function

- Most sampling algorithms require distribution up to the constant **partition function** Z_E:

$$p_E(\mathbf{z}) = \frac{1}{Z_E}\exp(-E(\mathbf{z})), \tag{11.71}$$
$$Z_E = \sum_{\mathbf{z}} \exp(-E(\mathbf{z})).$$

- Partition function is useful for model comparison
  - it expresses the probability of observed data in

$$p(hidden|observed) = \frac{1}{p(observed)}p(hidden, observed)$$

- For model comparison we need the ratio of partition functions.
- Often intractable to compute due to sum over *many* terms.

11.6 Estimating the Partition Function: Strategy 1

- Use importance sampling from proposal p_G with energy $G(z)$

$$\begin{aligned}
\frac{Z_E}{Z_G} &= \frac{\sum_{\mathbf{z}} \exp(-E(\mathbf{z}))}{\sum_{\mathbf{z}} \exp(-G(\mathbf{z}))} \\
&= \frac{\sum_{\mathbf{z}} \exp(-E(\mathbf{z}) + G(\mathbf{z})) \exp(-G(\mathbf{z}))}{\sum_{\mathbf{z}} \exp(-G(\mathbf{z}))} \\
&= \mathbb{E}_{p_G}[\exp(-E(\mathbf{z}) + G(\mathbf{z}))] \\
&\simeq \frac{1}{L} \sum_{l=1}^{L} \exp(-E(\mathbf{z}^{(l)}) + G(\mathbf{z}^{(l)})) \qquad (11.72)
\end{aligned}$$

- $\mathbf{z}^{(l)}$ are sampled from p_G
- If Z_G is easy to compute we can estimate Z_E
- For this approach to work p_E needs to match p_G well...

11.6 Estimating the Partition Function: Strategy 1 cntd.

- Problem: how to come up with a p_G that matches p_E?
- Idea: we can use samples $\mathbf{z}^{(l)}$ from p_E from a Markov chain:

$$p_G(\mathbf{z}) = \frac{1}{L} \sum_{l=1}^{L} T(\mathbf{z}^{(l)}, \mathbf{z}), \tag{11.73}$$

  where T gives the transition probabilities of the chain.

- We now define $G(\mathbf{z}) = -\log p_G(\mathbf{z})$, and use this in (11.72)

11.6 Estimating the Partition Function: Chaining

- Partition function ratio estim. requires matching distributions
- Problematic when estimating absolute value of part. function
  - Partition function Z_G needs to be evaluated exactly
  - Only simple, poor matching, distributions allow this...
- Use set of distributions between simple p_1 and complex p_M

$$\frac{Z_M}{Z_1} = \frac{Z_2}{Z_1}\frac{Z_3}{Z_2}\cdots\frac{Z_M}{Z_{M-1}} \tag{11.74}$$

- The intermediate distributions interpolate from E_1 to E_M

$$E_\alpha(\mathbf{z}) = (1-\alpha)E_1(\mathbf{z}) + \alpha E_M(\mathbf{z}) \tag{11.75}$$

- Now each term can be reasonably approximated