11. Sampling Methods

Matti Pöllä

April 2, 2007

Sampling

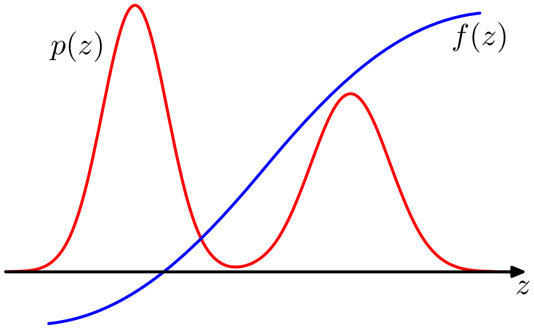
In many practical problems sampling is used to

- Obtain a collection of samples $\{\mathbf{z}^{(l)}\}$ from a distribution $p(\mathbf{z})$
- Approximate the expected value of a function $E[f] = \int f(\mathbf{z}p(\mathbf{z}))dz$

Sampling allows the expectation to be approximated as

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

Samping



What's so hard about sampling?

Why not just simply examine all possible values of $p(\mathrm{z})$ and generate samples accordingly.

Consider a problem where a 1000-dimensional variable can have 50 values in all dimensions. Integrating over the whole distribution would require evaluating $p(\mathrm{z})$ in 50^{1000} points. By comparison, the total amount of electrons in the universe is about 2^{266}...

Problems

- How to normalize the distribution $p(z) = \frac{1}{Z_p}\tilde{p}(z)$?
- How to generate independent samples?

The solution

Most sampling methods use the basic idea of

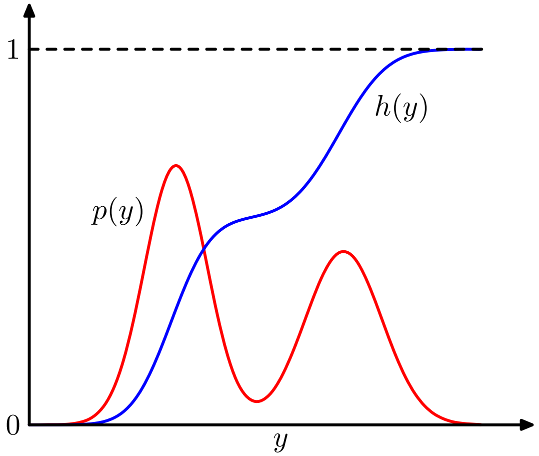
1. first selecting some other distribution $q(\mathbf{z})$ that *can* be sampled directly and then
2. compensate for the adverse effects of using a wrong distribution.

From uniform to nonuniform random numbers

- Generating uniformly distributed (pseudo)random numbers is easy
- Suppose that z is uniformly distributed in $(0, 1)$ and a function $f(\cdot)$ is used to make a transformation $y = f(z)$ and $p(y) = p(z)|\frac{dz}{dy}|$

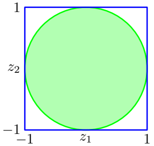
$$z = h(y) \equiv \int_{-\infty}^{y} p(\hat{y}) d\hat{y}$$

From uniform to nonuniform random numbers



Box-Muller

- The Box-Muller method can be used to generate Gaussian random numbers from uniformly distributed random numbers
- First, random number pairs $z_1, z_2 \in (-1, 1)$ are generated from



  a uniform distribution inside a unit circle
- Then, we evaluate

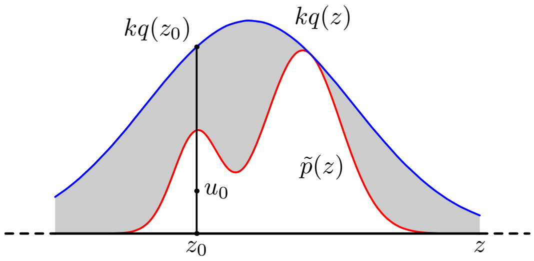
$$y_i = z_i \left(\frac{-2 \ln z_i}{r^2} \right)^{1/2}$$

  where $i = \{1, 2\}$ and $r^2 = z_1^2 + z_2^2$
- Now y_1 and y_2 are independent and have a Gaussian distribution with zero mean and unit variance

Rejection sampling

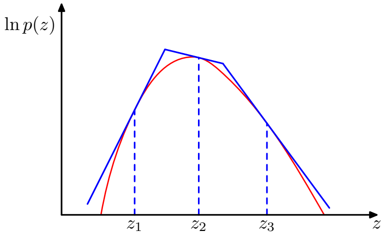
- Suppose we wish to generate samples from $p(\mathbf{z})$ for which we can easily evaluate $p(\mathbf{z}) = \frac{1}{Z}\tilde{p}(\mathbf{z})$
- We can draw samples from another (simpler) proposal distribution $q(\mathbf{z})$ multiplied by a constant k so that $kq(z) \geq \tilde{p}(z)\, \forall z$
- Now we can generate a random number z_0 from $q(z)$ and accept it with a probability $\tilde{p}(z)/kq(z)$
- The selection of $q(z)$ and k effects the rejection rate of samples

Rejection sampling



Adaptive rejection sampling

- In practice it is often difficult to determine a suitable analytic form for $q(z)$
- The envelope distribution can be adjusted during the sampling process by defining $q(z)$ in smaller pieces



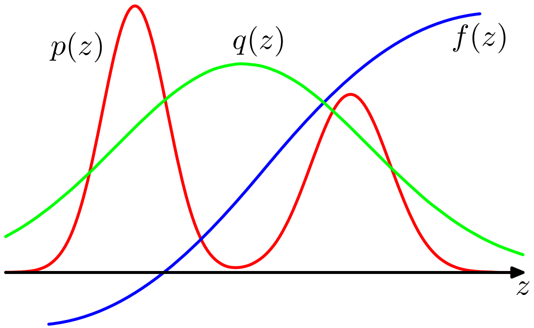
Importance sampling

- If we are only interested in evaluating the expectations of a function, we can use importance sampling to draw samples from a proposal distribution and compensate the error by using importance weights

$$E[f] = \int f(\mathbf{z})p(\mathbf{z})d\mathbf{z} = \int f(\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z} \simeq \frac{1}{L}\sum_{l=1}^{L} \underbrace{\frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z})^{(l)}}}_{\text{weight}} f(\mathbf{z}^{(l)})$$

- Similarly to rejection sampling, the performance is heavily dependent on the selection of $q(z)$

Importance sampling



Sampling-importance-resampling

- It is often difficult to define the parameter k which was needed in rejection sampling
- We can avoid using the parameter by the following proceedure
  1. draw L samples from $q(\mathbf{z})$
  2. compute importance weights $(w_1, \ldots, w_L)$ for the samples
  3. re-select L samples from $\{\mathbf{z}\}$ according to probabilities given by the weights

Sampling and the EM algorithm

- Sampling can also be useful in finding maximum likelihood solutions, for example, if the E-step cannot be performed analytically
- Consider a model with observed variables $\mathbf{X}$, hidden variables $\mathbf{Z}$ and parameters θ
- The maximized (M-step) complete-data log likelihood can be approximated by sampling

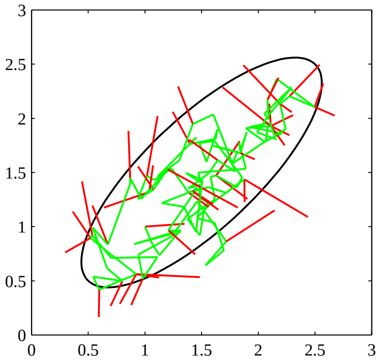
$$Q(\theta, \theta^{\text{old}}) = \int p(\mathbf{Z}|\mathbf{X}, \theta^{\text{old}}) \ln p(\mathbf{Z}, \mathbf{X}|\theta) d\mathbf{Z} \simeq \frac{1}{L} \sum_{l=1}^{L} \ln p(\mathbf{Z}^{(l)}, \mathbf{X}|\theta)$$

Markov Chain Monte Carlo

- The previously discussed sampling methods suffer from limitations in high-dimensional spaces due to high rejection rates
- MCMC methods produce samples by maintaining a record of a current state $\mathbf{z}^{(\tau)}$ and a proposal distribution $q(\mathbf{z}|\mathbf{z})^{(\tau)}$ to produce a sequence of samples $\mathbf{z}^1, \mathbf{z}^2, \mathbf{z}^3$ forming a Markov chain
- At each step a candidate sample $\mathbf{z}^*$ is generated and then accepted or rejected according to a criterion
- In the basic Metropolis algorithm, a candidate sample is accepted with a probability

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

Metropolis algorithm



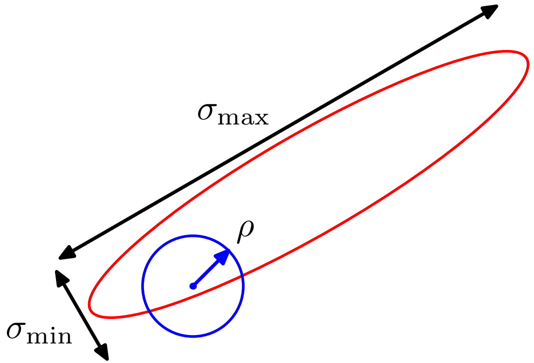
Metropolis-Hastings algorithm

- In the Metropolis-Hastings algorithm the proposal distribution is no longer symmetric and the acceptance of the proposal state is defined by

$$A_k(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*) q_k(\mathbf{z}^{(\tau)}|\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)}) q_k(\mathbf{z}^*|\mathbf{z}^{(\tau)})}\right)$$

  where k labels the members of the set of possible transitions being considered
- The scale of the proposal distribution has a strong influence on the performance of the algorithm
  - large step size $\longrightarrow$ independent samples
  - small step size $\longrightarrow$ small rejection rate

Metropolist-Hastings algorithm



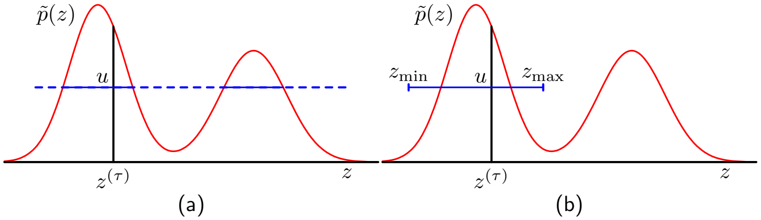
Gibbs sampling

- Gibbs sampling is a modification of the Metropolis-Hastings algorithm which involves iteratively replacing the value of one variable by a value drawn from the distribution of the value conditioned with all the other variables

$$z_i \leftarrow p(z_i|\mathbf{z}_{\setminus i})$$

- The practical applicability of Gibbs sampling is determined by the difficulty of sampling from the conditional distributions $p(z_i|\mathbf{z}_{\setminus i})$

Splice sampling



- The sensitivity to the step size in Metropolis algorithms can be overcome by using an adaptive step size using slice sampling
- Slice sampling involves augmenting z with an additional variable u and then drawing samples from the joint (z, u) space

Dynamical systems

The Hybrid Monte Carlo algorithm is rooted in the simulation of dynamical physical systems (Hamiltonian dynamics). Details omitted here, see section 11.5.1 of the book..

Basic idea: augment the variable position $\mathbf{z}$ in the state space with a momentum variable $\mathbf{r}$ and define the total energy of the system as the sum of the potential and kinetic energy

$$H(\mathbf{z}, \mathbf{r}) = E(\mathbf{z}) + K(\mathbf{r})$$

Hybrid Monte Carlo

- Standard Metropolis algorithms suffer from random walk -type behavior where the exploration of the state stapce is slow (proportional to $\sqrt{\text{number of steps}}$)
- Using a larger step size would only lead to a high rejection rate
- Now, new proposed states are accepted according to transisions in the Hamiltonian dynamics and accepting the state with the probability

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

Estimating the Partition Function

- Most of the considered sampling methods have only used the functional form of the distribution $\tilde{p}(\mathbf{z})$ without knowlegeabout the normalization constant Z (=partition function)
- If we write
$$p_E(\mathbf{z}) = \frac{1}{Z_E}\exp(-E(\mathbf{z}))$$
is not needed to draw samples from $p(\mathbf{z})$
- Still, knowledge about Z_E would be useful for Bayesian model comparison (since it represents the model evidence)
- To compare models, only the *ratio* of partition functions is needed and can be computed as
$$\frac{Z_E}{Z_G} = \frac{\sum_{\mathbf{z}} \exp(-E(\mathbf{z}))}{\sum_{\mathbf{z}} \exp(-G(\mathbf{z}))} \simeq \sum_{l} \exp(-E(\mathbf{z}^{(l)}) + G(\mathbf{z}^{(l)}))$$

Summary

- Sampling methods can be used to evaluate expectations in situations where deterministic inference methods are not applicable
- The accuracy of Monte Carlo estimates depends only on the variance (not dimensionality) of the sampled space – often a small number of samples suffices
- Basic sampling methods based on proposal distributions $q(\mathbf{z})$ do not scale well to higher-dimensional spaces
- Markov Chain Monte Carlo methods are useful in higher dimensions but the random walk behavior can produce dependent samples