Markov Chain Monte Carlo Methods

Michael Kenning

Two purposes:

- To find the optimal solution to some objective function
- To compute the posterior probability of some statistical learning problem

Principal Idea

- Construct samples as we go along such that the samples are from the most probable parts of the state space.
- The state space might be very large, but we’re only interested in the ultimate solution, not the steps in between.
- The **steps** are called *Markov chains*.
- The **sampling** is termed *Monte Carlo sampling.*

Sampling

- Pseudo-random numbers:
  - Using the modulus. Feed one number in, get another out, etc.
  - Generates numbers in a cycle; ideally the *period* (the number of numbers in a cycle) is as large as possible.
  - Used in random number generators.
  - Not genuinely random.
  - *Entropy* can be used to test the randomness of a sequence.
  - **But** just because a sequence has succeeded at a cursory test, doesn't mean the generator is truly random.
  - *Vice versa*, just because it fails, doesn't mean it is not truly random.
  - Hence the headache.

Sampling

- Gaussian random numbers:
  - Producing numbers from some distribution.
  - Requires two *uniformly distributed variables* to generate Gaussian-distributed values.

Monte Carlo

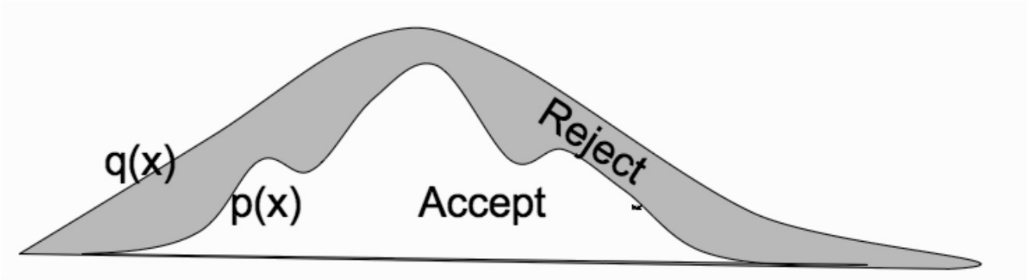
- Named after the principality of Monaco known for gambling.
- Statement of technique:
  - Suppose we sample *n* samples x from an unknown, high-dimensional distribution *p(*x).
  - The samples are independent and identically distributed (well-behaved).
  - As *n* increases, the *sample distribution* converges to the true distribution.

Monte Carlo

- As we draw more samples from the true distribution, the samples will be more likely to come from the high-probability parts of the true distribution.

The Proposal Distribution

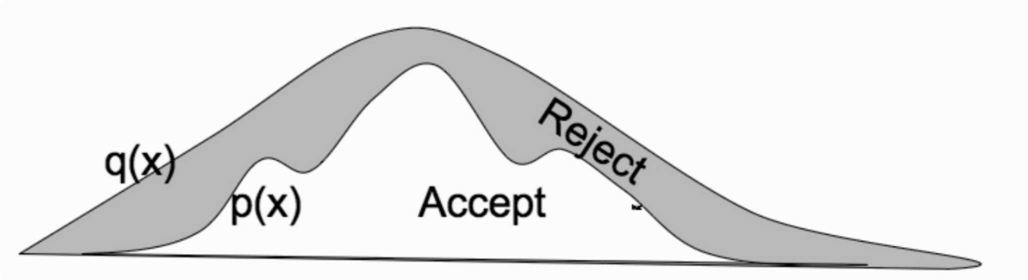
- We do not know what the true distribution $p(x)$ is.
- In some cases it is not easy to sample from, so we sample instead from a simpler distribution $q(x)$.
- To evaluate a sample x from $q(x)$, we use the distribution $\tilde{p}(x)$ *related* to $p(x)$ to evaluate the samples from $q(x)$.
- We want to satisfy $\tilde{p}(x) \leq mq(x)$, where m is some arbitrary scalar.
  - In other words, $q(x)$ must be *at least* m-times greater than $\tilde{p}(x)$ everywhere on the function; x cannot be more likely to appear in $q(x)$ than $\tilde{p}(x)$.



The Proposal Distribution: Rejection Sampling

The Rejection Sampling Algorithm

- Sample $\mathbf{x}^*$ from $q(\mathbf{x})$ (e.g., using the Box–Muller scheme if $q(\mathbf{x})$ is Gaussian)
- Sample u from uniform$(0, \mathbf{x}^*)$
- If $u < p(\mathbf{x}^*)/Mq(\mathbf{x}^*)$:
  - add $\mathbf{x}^*$ to the set of samples
- Else:
  - reject $\mathbf{x}$ and pick another sample



The Proposal Distribution: Rejection Sampling

- $mq(x)$ forms an *envelope* on $p(x)$ (by the related function $\tilde{p}(x)$).