MCMC - SAMPLING

Machine Learning

IIIy CSE IDP and IDDMP

Refer : Stephen Marsland

Motivation

Revolution in Statistical Computing & Physics

1. MCMC methods have transformed these fields over the past 20 years.
2. Core algorithm dates to 1953, but only became practical when computers could run real-world examples in hours instead of weeks.
3. Now cited as one of the most influential algorithms ever created.

Two Fundamental Problems Addressed by MCMC

4. Finding the optimum solution to an objective function.
5. Estimating the posterior distribution in a statistical learning problem.

Challenges

6. Very large state spaces.
7. Only interested in the best answers; intermediate steps are not important.

Sampling

Monte Carlo Sampling Idea

As we explore the state space, we build up samples that concentrate in the most probable regions.

To understand this, we'll review:

- Monte Carlo sampling
- Markov chains

Sampling in Previous Algorithms

Used for weight initialization and other steps.

Examples:

- Uniform sampling on [0, 1): np.random.rand()
- Gaussian sampling: np.random.normal()

Random Number Generation

True randomness is not possible on deterministic computers.

Pseudo-random numbers are generated via algorithms.

Simplest example: the linear congruential generator:

$$x_{n+1} = (ax_n + c) \mod m$$

Next Steps

Explore how Markov chains use these samples to approximate complex distributions.

Pseudo-Random Number Generators

Linear Congruential Generator (LCG)

Recurrence uses only integers: seed x_0 and parameters **a,c,m.**

Outputs are in {0,…,m−1} once a value repeats, the sequence cycles.

Period = length before repeat; ideal when period = **m**.

Common “good” parameters:

$m=2^{32}$; a = 1,664,525 ; c = 1,013,904,223

Mersenne Twister

- Industry standard algorithm based on Mersenne primes.
- Default RNG in NumPy.

Randomness Caveats

Pseudo-random $\Rightarrow$ not genuinely random.

No proof of randomness; only tests for non-random patterns (e.g., entropy checks, compression ratio, odd/even balance).

Passing tests ≠ guaranteed randomness; truly random sequences can still “fail” occasionally.

Gaussian Random Numbers

1. Goal

a. Convert uniform random samples into independent standard Gaussian samples.

2. Underlying Concept

b. View a pair of independent Gaussian variables as points in the plane, characterized by a random distance from the origin and a random angle.

c. The angle is uniformly distributed around the circle.

d. The distance follows a specific distribution that concentrates more samples near the origin.

3. Descriptive Algorithm Steps

e. **Draw two uniform samples** from your RNG.

f. **Map one sample to an angle** around the circle (evenly around 360°).

g. **Map the other sample to a radius** by transforming it so that distances closer to the origin are more likely.

h. **Convert polar to Cartesian** by interpreting the angle and radius as coordinates in the plane.

i. **Result:** Two independent values that follow a standard normal distribution.

4. Why It Works

j. Uniform angles ensure all directions are equally likely.

k. The special radius transformation clusters points according to the Gaussian bell shape.

l. In the next slide: detailed Box–Muller algorithm.

Box-Muller Scheme

The Box–Muller Scheme

- Pick two uniformly distributed random numbers $0 \leq U_1, U_2 \leq 1$
- Set $\theta = 2\pi U_1$ and $r = \sqrt{(} - 2\ln(U_2))$
- Then $x = r\cos(\theta)$ and $y = r\sin(\theta)$ are independent Gaussian-distributed variables with zero mean and unit variance

Rejection Method for Gaussian Sampling

Marsaglia “Polar” Approach

1. Draw two uniforms and rescale to $[-1,1]$; interpret as a point in the plane.
2. Compute $w^2=U_1^2+U_2^2$.
3. If the point lies outside the unit circle ($w^2>1$), reject and redraw.
4. Otherwise, apply a simple transform to (U_1,U_2) to obtain two independent Gaussian samples.

Trade-Offs

5. **Box–Muller** uses trigonometric functions (sin, cos).
6. **Polar method** incurs unpredictable rejection overhead.
7. Performance depends on language and hardware.

Beyond Gaussians

8. Other distributions may lack a “closed-form” sampler.
9. Rejection sampling offers a general framework: propose candidates, reject those outside the target density.
10. Adds some computational cost when many proposals are discarded, but often simpler than custom transforms.

(Figure 15.1 illustrating Box–Muller samples can be placed on the next slide.)

Gaussian Random Numbers

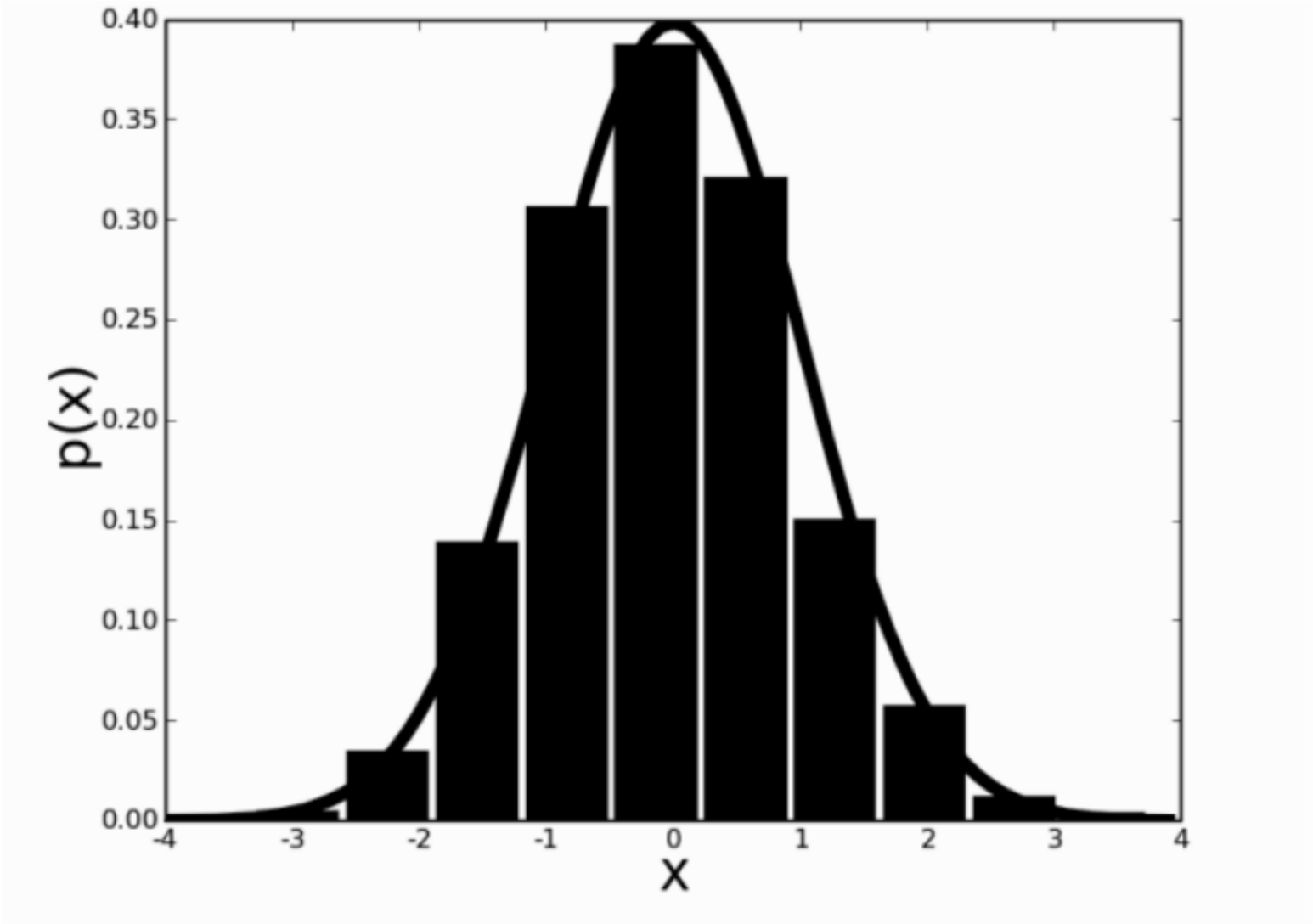

FIGURE 15.1 Histogram of 1,000 Gaussian samples created by the Box–Muller scheme. The line gives the Gaussian distribution with zero mean and unit variance.

Summary & Conclusion

MCMC Motivation

- Powerful for optimizing complex objectives and sampling posteriors in large state spaces.
- Became practical with modern computing power.

Sampling Foundations

- Pseudo-random numbers (LCG, Mersenne Twister) underpin Monte Carlo methods.
- True randomness impossible; rely on quality of generators and randomness tests.

Transforming Uniform to Gaussian

- **Box–Muller**: polar coordinate interpretation using angle and radius transforms.
- **Polar (Rejection) Method**: discard points outside unit circle, then scale.
- Each has trade-offs: trig vs. rejection overhead.

General Rejection Sampling

- Core idea: propose candidates, reject those that don't meet density criteria.
- Forms the basis for many MCMC algorithms

Key Takeaway

- Rejection and transformation techniques allow sampling from virtually any distribution without custom code.