COMPSCI 330 Sept. 26th 2017

Lecture # 9: Rejection Sampling and Monte-Carlo Method

Lecturer: Rong Ge *Scribe: Xingyu Chen*

1 Rejection Sampling

In mathematics, rejection sampling is a basic technique used to generate observations from a distribution. Using this technique, given one distribution, we can draw samples seemingly from another distribution by some rejection/acceptance to the sample.

1.1 Problem Setting

Problem: Given a random variable X, $Pr[X = i] = p_i$. (This is our base distribution)
Want to generate a random variable Y, such that $Pr[Y = i] = q_i$. (This is the distribution we want to obtain)

1.2 Technique We Use

Procedure: Sample X, then with probability $\frac{q_i}{c \cdot p_i}$ keep this sample, otherwise throw the sample away.

We claim that with this operation, given the sample is kept, the probability of drawing the ith sample is exactly q_i. The proof is as follows:

Claim 1. $Pr[x = i | x \text{ is kept}] = q_i = Pr[Y = i]$

Proof.

$$Pr[x = i|\text{x is kept}] = \frac{Pr[x = i, \text{x is kept}]}{Pr[\text{x is kept}]}$$

$$Pr[x = i, \text{x is kept}] = p_i \cdot \frac{q_i}{c \cdot p_i} = \frac{q_i}{c}$$

$$Pr[\text{x is kept}] = \sum_t Pr[x = t, \text{x is kept}] = \sum_t \frac{q_t}{c} = \frac{1}{c}$$

Combining the above equations, we have

$$Pr[x = i|\text{x is kept}] = \frac{Pr[x = i|\text{x is kept}]}{Pr[\text{x is kept}]} = \frac{\frac{q_i}{c}}{\frac{1}{c}} = q_i$$

□

1.3 Example: Coin Toss

Suppose we only have a biased coin (the coin lands on heads with probability $p > \frac{1}{2}$). How do we simulate a fair coin? By using the rejection sampling technique we can just

- Sample the biased coin. $Pr[H] = p_1, Pr[T] = p_2$.

- Then we set $q_i = \frac{1}{2}$, $c = \frac{1}{2 \cdot p_1 \cdot p_2}$.
- With probability $\frac{\frac{1}{2}}{\frac{1}{2 \cdot p_1 \cdot p_2} \cdot p_i}$, we keep the sample. Otherwise, we throw it away.

Using this technique, we sample from an unbiased coin a biased coin. In fact, the previous abstract technique is equivalent to the following:

- Sample the biased coin twice.
- HT means a head. TH means a tail. HH and TT are thrown away.

The reason for the equivalency is because $\frac{\frac{1}{2}}{\frac{1}{2 \cdot p_1 \cdot p_2} \cdot p_i}$ is equal to $p_{\{1,2\} \setminus i}$. So it's essentially the same as the possibility of getting tail when you get head at the first round or getting head when you get tail at the first round.

After the proof of our scheme, a natural problem comes to our mind.
Problem: How many coin tosses do we need to do until we get the result?

Claim 2. *We need to toss the coin with expectation of $\frac{1}{p(1-p)}$ times to get the result.*

Proof. Let X be the number of tries that we need before we succeed.

$$
\begin{aligned}
Pr[X = 1] &= 2p(1-p) = q \\
Pr[X = 2] &= (1-q) \cdot q \\
&\cdots \\
Pr[X = i] &= (1-q)^{i-1} \cdot q
\end{aligned}
$$

After these equations, we can calculate the expected times to get the first success.

$$
\begin{aligned}
\mathbf{E}[X] &= \sum_{i=1}^{\infty} Pr[X = i] \cdot i \\
&= \sum_{i=1}^{\infty} Pr[X \geq i] \\
&= \sum_{i=1}^{\infty} (1-q)^{i-1} \\
&= \frac{1}{q} = \frac{1}{p \cdot (1-p)}
\end{aligned}
$$

In conclusion, we know the expected value of coin tosses to get the result is $\frac{1}{p \cdot (1-p)}$ times. □

2 Monte Carlo Algorithm

Monte Carlo methods (or Monte Carlo experiments) are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. Their essential idea is using randomness to solve problems that might be deterministic in principle.

2.1 Problem Setting

Problem: Compute the area of a circle in (-1,-1) to (1,1) square.

2.2 Algorithm

Pseudocode:

```
Count = 0
FOR i = 1 to n
      Generate x, y from [-1,1]
      IF x^2+y^2 <= 1 THEN
            Count = Count + 1
RETURN 4.0*Count/n
```

Simple Analysis:

Let $x_i = \begin{cases} 0 & \text{i-th point is not in circle} \\ 1 & \text{i-th point is in circle} \end{cases}$

Then we can calculate the expected value of X_i

$$\mathbf{E}[X_i] = Pr[X_i = 1] = \frac{\text{area of circle}}{\text{area of square}} = P$$

Let $X = \sum_{i=1}^{n} X_i$, then the output is $area \cdot Pr[\text{Point in Circle but not Square}] = 4 \cdot \frac{X}{n}$.

2.3 Analysis

A natural question arrives here. How does our algorithm do? Or in other words, how much points do I need to approximate the correct answer?

Fact 3.

$$\begin{aligned} Var[X_i] &= \mathbf{E}[(X_i - [\mathbf{X_i}])^2] \\ &= \mathbf{E}[(X_i - P)^2] \\ &= (1-P)(0-P)^2 + P(1-P)^2 \\ &= P(1-P)[P + (1-P)] \\ &= P(1-P) \end{aligned}$$

Then the variance for X is

$$\begin{aligned} Var[X] &= \sum_{i=1}^{n} Var[X_i] \\ &= n \cdot P \cdot (1-P) \end{aligned}$$

Since $Var[c \cdot X] = c^2 Var[X]$ *and* $Output = \frac{4X}{n}$,

$$Var[Output] = (\frac{4}{n})^2 \cdot Var[X] = \frac{16 \cdot P \cdot (1-P)}{n}$$

With the previous facts, we want to prove our algorithm is close enough to the correct answer with high possibility. The claim is with possibility $\frac{3}{4}$, output is within $\frac{4}{\sqrt{n}}$ of the correct answer.

Claim 4. $Pr[|Output - Area\ of\ Circle| > \frac{4}{\sqrt{n}}] \leq \frac{1}{4}$

Proof.

$$\begin{aligned} Var[\text{Output}] &= \frac{16P(1-P)}{n} \leq \frac{4}{n} \\ \sqrt{Var[\text{Output}]} &\leq \frac{2}{\sqrt{n}} \\ 2\sqrt{Var[\text{Output}]} &\leq \frac{4}{\sqrt{n}} \end{aligned}$$

According to Chebshev's inequality, $Pr[|X - E(X)| \geq \lambda\sigma] \leq \frac{1}{\lambda^2}$, where $\lambda = 2$ and $\sigma = \sqrt{\text{Variance}}$

$$Pr[|\text{Output} - \text{Area of Circle}| > \frac{4}{\sqrt{n}}] \leq Pr[|\text{Output} - \text{Area of Circle}| > 2\sqrt{Var[\text{Output}]}] \leq \frac{1}{4}$$

□