2-inverse_transform_sampling

Heitor

2023-04-13

Inverse Transform Sampling Method

The ideia here is that we can sample random variables from a distribution that we don't know the density function from another (often easier) distribution that we do know its density function.

To do so, we claim that the inverse of that distribution that we know the p.m.f/p.d.f equals the p.d.f of the distribution we want to sample from.

Given that we need to have the inverse of that function, for some distributions that's harder to do, such as the Normal Distribution.

More intuition

Suppose you have a game with three possible champions:

1. A mage

2. A warrior
3. A rogue

And suppose that, in that game, you want 30% being mages, 50% being warriors and 20% being rogues. You could simply, given a number of characters allocate each class accordingly to each distribution. But, that'd be very predictable.

Remember that we want to generate random samples, and it wouldn't be random. So, to make it random, we make use of the Standard Uniform Distribution. A Standard Uniform Distribution is a random generator where each value between [0,1] has an equal probability of being choosen.

So, the algorithm goes as follows:

1. Pick a random number from a Uniform Dsitribution
2. If that number is less or equal than 0.3, generate a mage. If it is less or equal than 80% generate a warrior, else, generate a Rogue.

That's it. We're using the CDF of a Uniform Distribution onto the density of the distribution we want to generate samples from, to get the desired values.

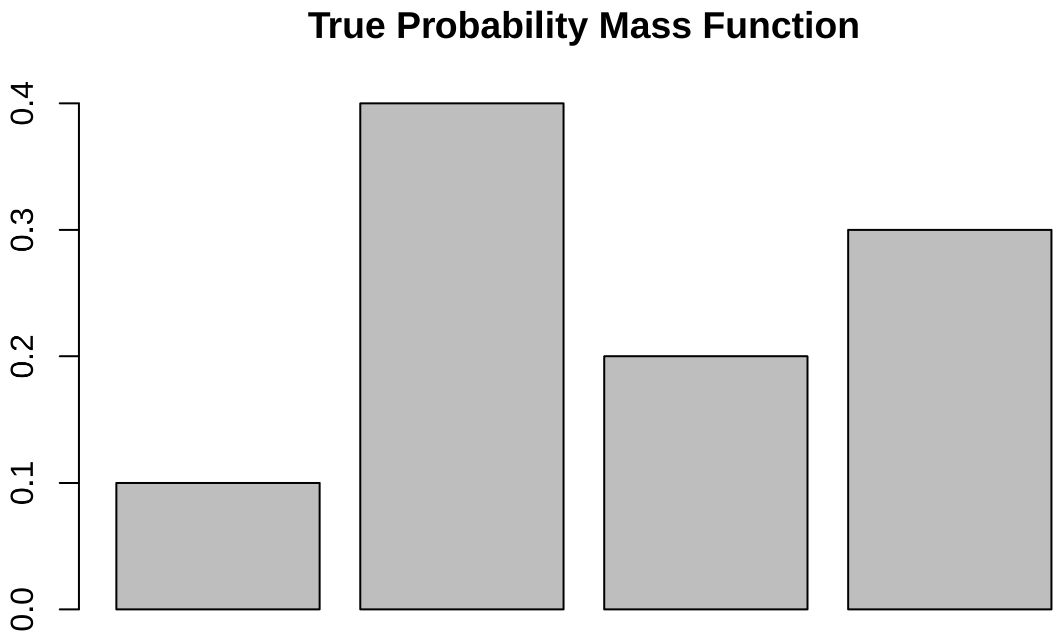
Discrete Distributions

```r
#let's consider a multinomial distribution

discrete.inv.transform.sample <- function(p.vec){
  U <- runif(1)
  # print(U)
  if(U <= p.vec[1]){
    return (1)
  }
  for(state in 2:length(p.vec)){
    if(sum(p.vec[1:(state-1)]) < U && U <= sum(p.vec[1:state])){
      return(state)
    }
  }
}
```

Simulating for a discrete distribution

```r
num.samples <- 1000
p.vec <- c(0.1, 0.4, 0.2, 0.3)
samples <- numeric(num.samples)
for(i in seq_len(num.samples)){
  samples[i] <- discrete.inv.transform.sample(p.vec)
}
barplot(p.vec, main = "True Probability Mass Function")
```


True Probability Mass Function