Inverse CDF Sampling

Troy Stribling

July 6, 2018

1 Introduction

Inverse CDF sampling is a method for obtaining samples from both discrete and continuous probability distributions that requires the CDF to be invertable. The method assumes values of the CDF are Uniform random variables on [0, 1]. CDF values are generated and used as input into the inverted CDF to obain samples with the distribution defined by the CDF.

2 Sampling Discrete Distributions

A discrete probability distribution consisting of a finite set of N probability values is defined by, $\{p_1, p_2, \dots, p_N\}$ with $p_i \geq 0, \forall i$ and $\sum_{i=1}^{N} p_i = 1$. The CDF specifies the probability that $i \leq n$ and is given by,

$$P(i \leq n) = P(n) = \sum_{i=1}^{n} p_i, \tag{1}$$

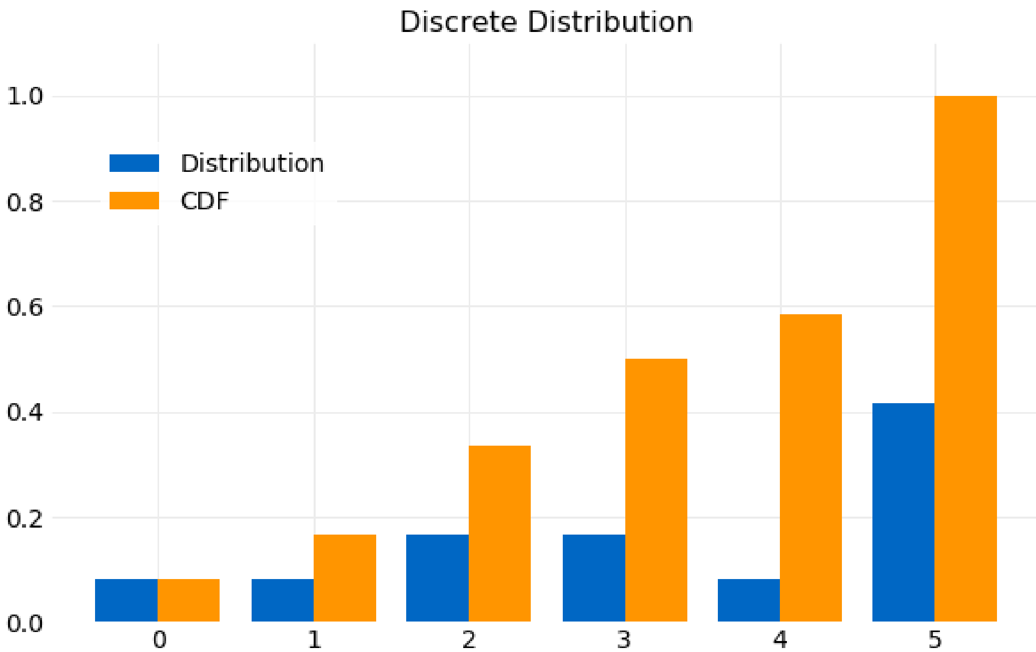
where $P(N) = 1$.

For a given generated CDF value, u, Equation (1) can always be inverted by evaluating it for each n and searching for the value of n that satisfies, $P(n) \geq u$. It can be seen that the generated samples will have distribution $\{p_n\}$ since the intervals $P(n) - P(n-1) = p_n$ are Uniformly sampled.

Consider the distribution,

$$\left\{\frac{1}{12}, \frac{1}{12}, \frac{1}{6}, \frac{1}{6}, \frac{1}{12}, \frac{5}{12}\right\} \tag{2}$$

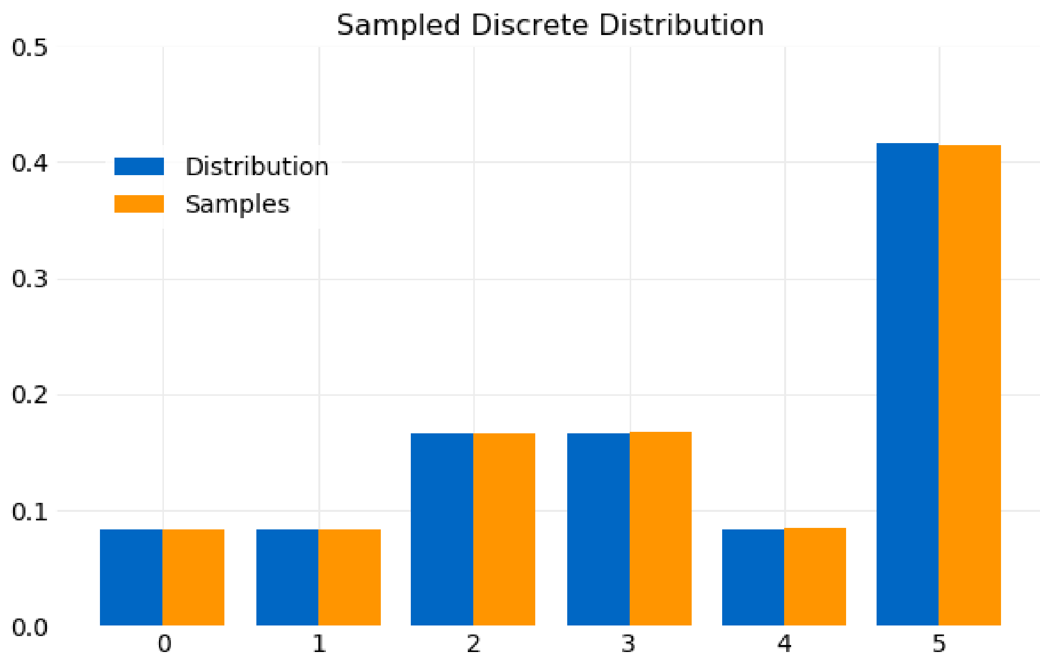
It is shown in the following plot with its CDF.



A sampler using the Inverse CDF method can be implemented in Python in a few lines of code. The program

```
import numpy

n = 10000
df = numpy.array([1/12, 1/12, 1/6, 1/6, 1/12, 5/12])
cdf = numpy.cumsum(df)

cdf_star = numpy.random.rand(n)
samples = [numpy.flatnonzero(cdf >= cdf_star)[0] for cdf_star in numpy.random.rand(nsamples)]
```

The figure below favorably compares generated samples and distribution (2),



It is also possible to directly sample $\{p_n\}$ using the `multinomial` sampler from `numpy`,

```
import numpy

n = 10000
df = numpy.array([1/12, 1/12, 1/6, 1/6, 1/12, 5/12])
samples = numpy.random.multinomial(n, df, size=1)/n
```

3 Sampling Continuous Distributions

A continuous probability distribution is defined by the PDF, $f_X(x)$, where $f_X(x) \geq 0, \forall x$ and $\int f_X(x)dx = 1$. The CDF is a monotonically increasing function that specifies the probability that $X \leq x$, namely,

$$P(X \leq x) = F_X(x) = \int^x f_X(w)dw. \tag{3}$$

3.1 Proof

To prove that Inverse CDF sampling works for continuos distributions it must be shown that,

$$P[F_X^{-1}(U) \leq x] = F_X(x), \tag{4}$$

where $F_X^{-1}(x)$ is the inverse of $F_X(x)$ and $U \sim \mathbf{Uniform}(0,1)$. A more general result needed to complete this proof is obtained using a change of variables on a CDF. If $Y = G(X)$ is a monotonically increasing invertable function of X then

$$P(X \leq x) = P(Y \leq y) = P[G(X) \leq G(x)]. \tag{5}$$

To prove this note that $G(x)$ is monotonically increasing so the ordering of values is preserved,

$$X \leq x \implies G(X) \leq G(x).$$

Consequently, the order of the integration limits is maintained by the transformation. Futher, since $G(x)$ is invertable, $x = G^{-1}(y)$ and $dx = \frac{dG^{-1}}{dy}dy$, so

$$\begin{aligned}
P(X \leq x) &= \int^x f_X(w)dw \\
&= \int^y f_X(G^{-1}(z))\frac{dG^{-1}}{dz}dz \\
&= \int^y f_Y(z)dz \\
&= P(Y \leq y) \\
&= P[G(X) \leq G(x)],
\end{aligned}$$

where,

$$f_Y(y) = f_X(G^{-1}(y))\frac{dG^{-1}}{dy}$$

The proof of Equation (4) follows from Equation (5), using $f_U(u) = 1$ since $U \sim \mathbf{Uniform}(0,1)$,

$$
\begin{aligned}
P[F_X^{-1}(U) \leq x] &= P[F_X(F_X^{-1}(U)) \leq F_X(x)] \\
&= P[U \leq F_X(x)] \\
&= \int_0^{F_X(x)} f_U(w)dw \\
&= \int_0^{F_X(x)} dw \\
&= F_X(x).
\end{aligned}
$$

3.2 Example

Consider the Weibull Distribution, with density

$$
f_X(x; k, \lambda) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{\left(\frac{-x}{\lambda}\right)^k} & x \geq 0 \\ 0 & xlt0, \end{cases} \tag{6}
$$

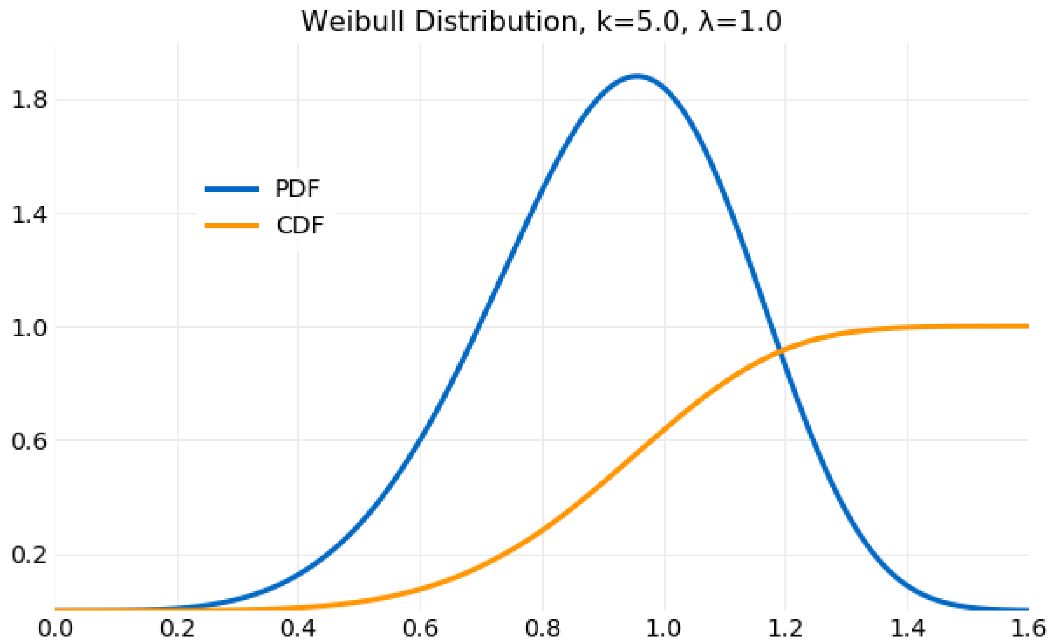
where k is the shape parameter and λ the scale parameter. The CDF is given by,

$$
F_X(x; k, \lambda) = \begin{cases} 1 - e^{\left(\frac{-x}{\lambda}\right)^k} & x \geq 0 \\ 0 & x < 0. \end{cases} \tag{7}
$$

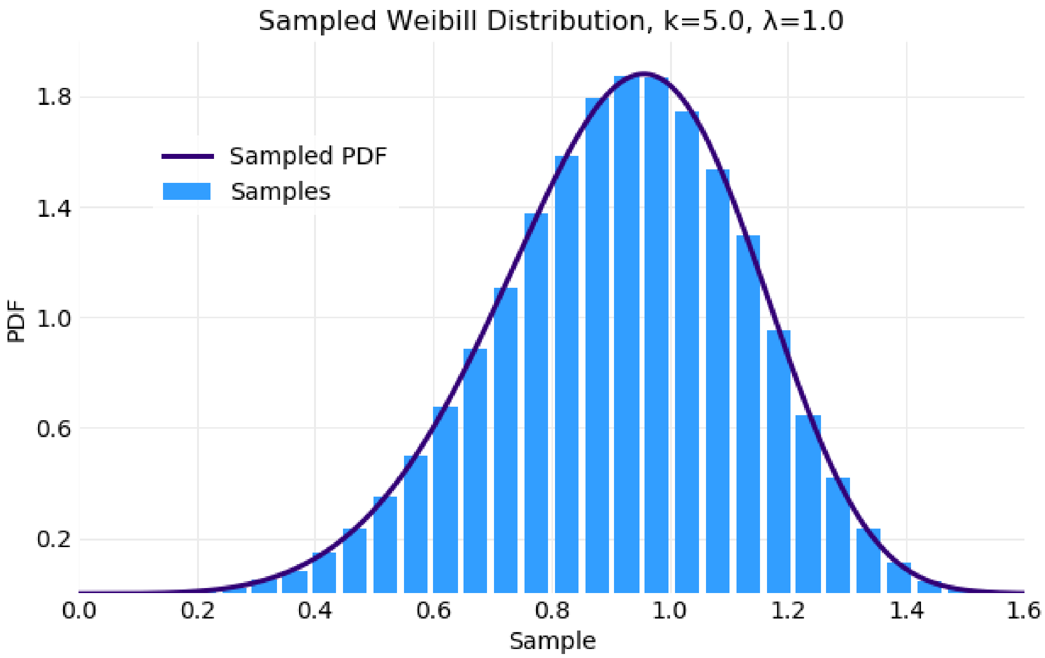
The CDF can be inverted to yield,

$$
F_X^{-1}(u; k, \lambda) = \begin{cases} \lambda \ln \left(\frac{1}{1-u}\right)^{\frac{1}{k}} & 0 \leq u \geq 1 \\ 0 & u < 0 \text{ or } u > 1. \end{cases} \tag{8}
$$

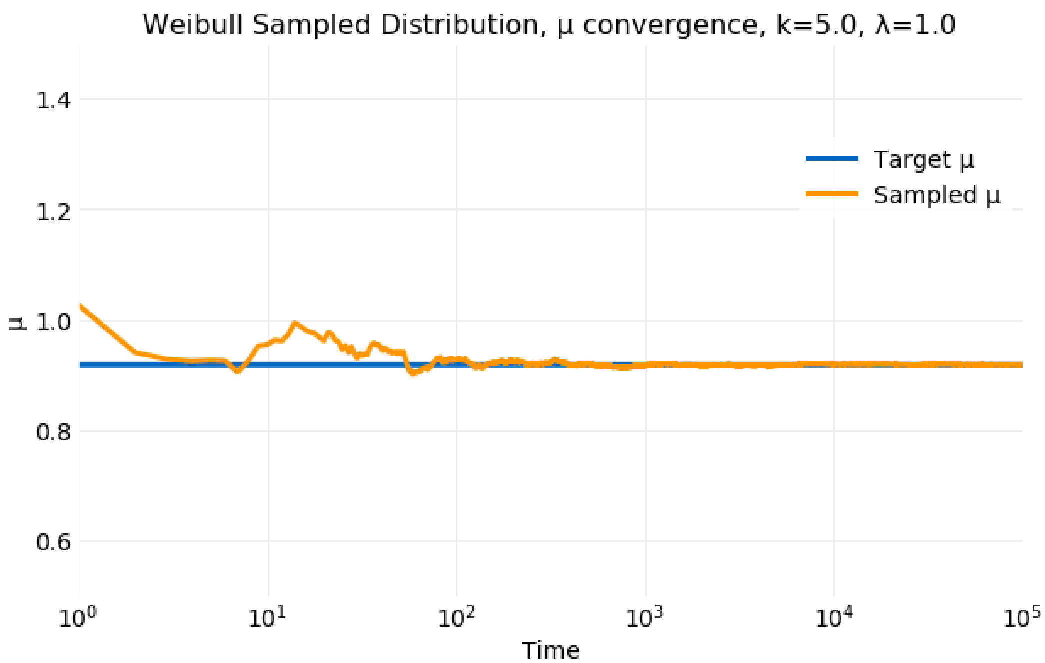
In the example described here it will be assumed that $k = 5.0$ and $\lambda = 1.0$. The following plot shows the PDF and CDF using these values.



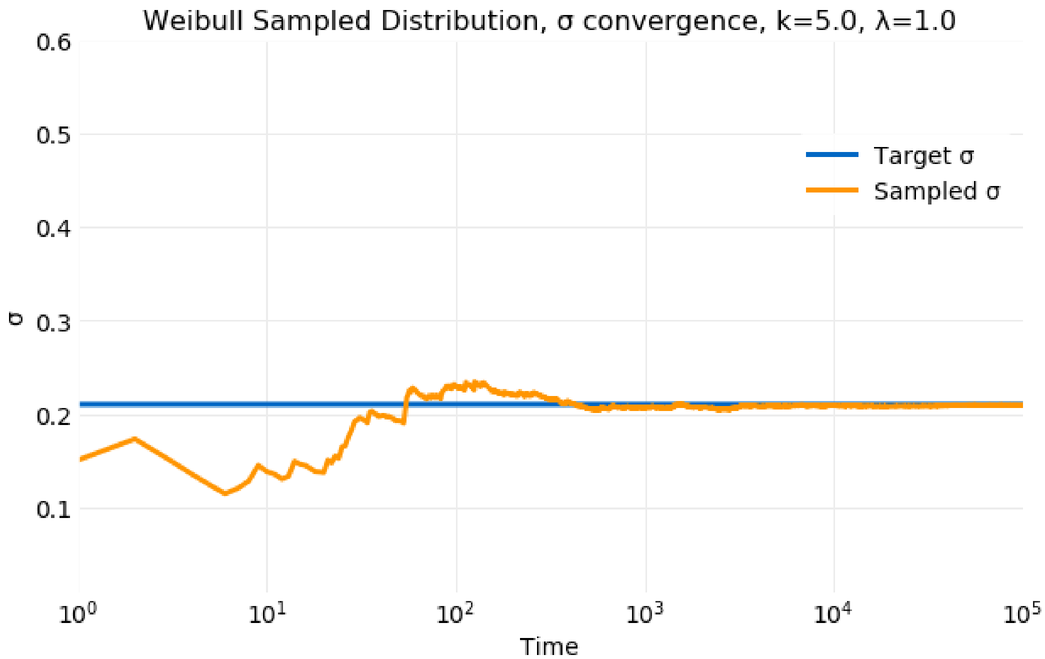

```
import numpy

k = 5.0
l = 1.0
nsamples = 100000

cdf_inv = lambda u: l * (numpy.log(1.0/(1.0 - u)))**(1.0/k)
samples = [cdf_inv(u) for u in numpy.random.rand(nsamples)]
```



Sampled Weibill Distribution, k=5.0, λ=1.0
Sampled PDF
Samples
PDF
Sample
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8



Weibull Sampled Distribution, μ convergence, k=5.0, λ=1.0
Target μ
Sampled μ
μ
Time
0.6
0.8
1.0
1.2
1.4
10^0
10^1
10^2
10^3
10^4
10^5



Weibull Sampled Distribution, σ convergence, k=5.0, λ=1.0
0.6
0.5
0.4
0.3
0.2
0.1
σ
Target σ
Sampled σ
10^0
10^1
10^2
10^3
10^4
10^5
Time