# ENSAE - Computational Statistics

Professor Christian P. Robert

Zakarya Ali

December 14, 2017

**Problem 8.5.** Show that a possible slice sampler associated with the standard normal density, $f(x) \propto exp(-x^2/2)$, is associated with the two conditional distributions

$$\omega|x \sim U[0, exp(-x^2/2)]$$

$$X|\omega \sim U([-\sqrt{-2log(\omega)}, \sqrt{-2log(\omega)}])$$

Suggesting the following :

```
Algorithm 1 Slice Sampler
$\forall t$,
    • Draw $U^{(t+1)}|x(t) \sim U_{[0,exp(\frac{(-x^{(t)})^2}{2})]}$
    • Draw $X^{(t+1)}|\omega(t+1) \sim U_{\{x/\omega^{(t+1)} \leq exp(\frac{(-x^{(t)})^2}{2})\}}$
```

We notice that

$$\begin{aligned} 0 \leq \omega \leq exp(\frac{-x^2}{2}) &\Leftrightarrow log(\omega) \leq \frac{-x^2}{2} \\ &\Leftrightarrow \frac{x^2}{2} \leq -log(\omega) \\ &\Leftrightarrow -\sqrt{-2log(\omega)} \leq x \leq \sqrt{-2log(\omega)} \end{aligned}$$

If $x$ is uniformly distributed within these boundaries, then $\omega$ is uniformly distributed betwen 0 and $f(x)$ for a given $x$. Besides, the fundamental theorem of simulation shows that $x$ is distributed with regard to $f$.

Figure 1 show the generation of $10^5$ observations from the Slice Sampler. We notice they are well distributed. With Figure 2 we have a better vision of our sample density compared to the real normal distribution.

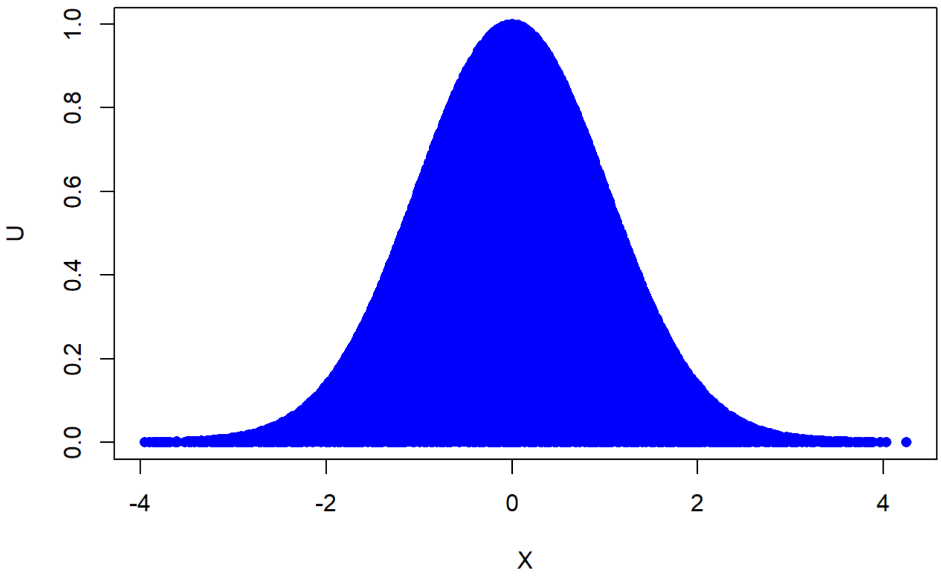

Figure 1: Slice sampler generation

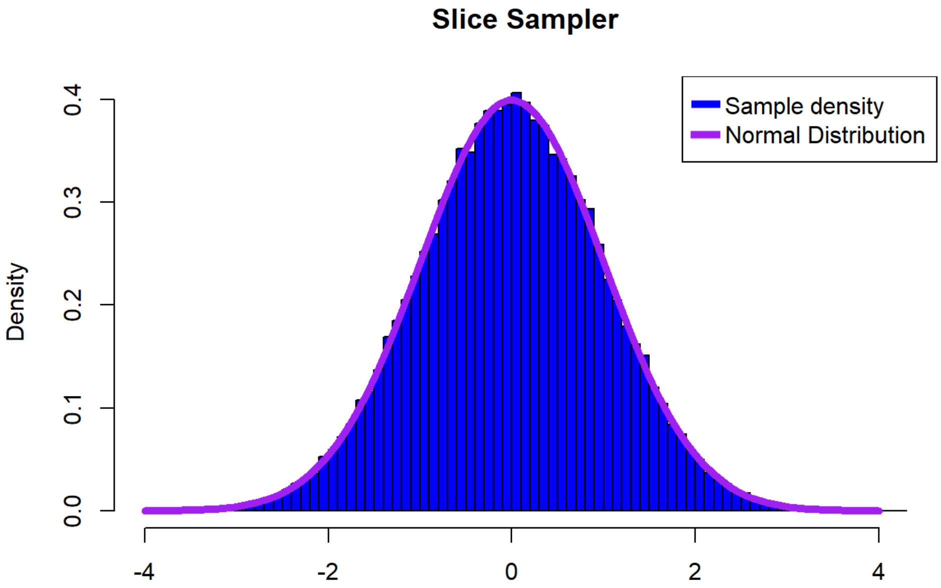

Figure 2: Slice sampler density compared to normal distribution

```
# Initialization
#set.seed(2)
iterations = 10**5

#function
slice_sampler_gamma ← function(n, x0){
  X = NULL
  U = NULL
  Xi = x0
  for(i in seq(n)){
    Ui = runif(1,0,exp((-Xi**2)/2))
    U = c(U, Ui)
    Xi = runif(1,-sqrt(-2*log(Ui)),sqrt(-2*log(Ui)))
    X = c(X,Xi)
  }
  m = matrix(c(X ,U), ncol =2)
   colnames (m) = c("X","U")
   return (m)
}

#generate slice sampler and normal distribution
sample ← slice_sampler(iterations ,0)
normal_distribution ← dnorm(seq(-4, 4, length=100), 0, 1)

#plot
plot(sample, main="Slice Sampler", pch=16, col="blue")
```

Listing 1: Code for Slice Sampler generation and plotting

**Compare the performances of this slice sampler with those of an iid sampler from N(0, 1) by computing the empirical cdf at 0, .67, .84, 1.28, 1.64, 1.96, 2.33, 2.58, 3.09, and 3.72 for two samples of same size produced under both approaches. (Those figures correspond to the .5, .75, .8, .9, .95, .99, .995, .999 and .9999 quantiles, respectively.)**

To assess the slice sampler precision we compare the cdf obtained by it with R command **rnorm** for the same amount of iterations. Table 1 shows what we get with **rnorm** while Table 2 present the Sampler results.

For $10^5$ iterations we get a mean squared error of $8.448221 * 10^4$ for the rnom and $8.0543433 * 10^4$ for the slice sampler. Hence, our method perform slightly better for these inputs.

```
# get cdfs
probs = c(0.5,0.75,0.8,0.9,0.95,0.99,0.995,0.999,0.9999)
rnorm_cdfs = quantile(ecdf(rnorm(iterations)),probs=probs)
rnorm_cdfs_unnamed = unname(rnorm_cdfs)
rnorm_cdfs

slice_cdfs = quantile(ecdf(sample[,1]),probs=probs)
slice_cdfs_unnamed = unname(slice_cdfs)
```

| | rnorm | sampler | N(0,1) |
|---|---|---|---|
| 50% | 0.002887848 | 0.002832747 | 0 |
| 75% | 0.674783753 | 0.682401822 | 0.67 |
| 80% | 0.844931961 | 0.846329639 | 0.84 |
| 90% | 1.279642718 | 1.288363751 | 1.28 |
| 95% | 1.644700687 | 1.644684744 | 1.64 |
| 99% | 2.346769318 | 2.331568072 | 1.96 |
| 99.5% | 2.585425696 | 2.577721797 | 2.58 |
| 99.9% | 3.037116534 | 3.070540260 | 3.09 |
| 99.99% | 3.653529575 | 3.801048043 | 3.72 |

Table 1: Performance of Slice Sampler and rnorm N(0,1)

```
slice_cdfs

real_cdfs = c(0, 0.67,.84, 1.28, 1.64,2.33, 2.58, 3.09, 3.72)
#rnorm
mse_rnorm = mean((rnorm_cdfs_unnamed - real_cdfs)^2)
#slice sampler
mse_slice = mean((slice_cdfs_unnamed - real_cdfs)^2)
```

Listing 2: Getting rnorm and Slice sampler CDFs and MSE