’Tis the Season to Be Jolly

Wonky Approach

Krzysztof Pytka

12/24/2018

Abstract

The purpose of this note is to propose a new method for sending holiday greetings. To this end, I employ simulation of random variables to create a *stochastic X-mas tree*. The proposed approach consists in two components, a theoretical part and a simulation part. The simulation part allows to introduce a new dimension of personalization of greetings, where the simulation seed is a function of the name of the greeting recipient. In this version of the greetings the recipient is *Duk*.

Theoretical framework

The X-mas tree is constituted by a trunk, needles, Christmas bulbs, and surrounding snow. Each component is defined by its data generating process:

1. a trunk follows:
$$(x^\tau, y^\tau): \quad x^\tau \sim \mathcal{U}[-.1, .1] \wedge y^\tau \sim \mathcal{U}[-.1, .1] \tag{1}$$
where $\mathcal{U}[a, b]$ is the uniform distribution with its support defined on $[a, b]$.

2. conifer needles are drawn from:
$$(x^\eta, y^\eta): \quad y^\eta \sim \mathcal{U}[0, 1] \wedge x^\eta \sim \begin{cases} \mathcal{U}\left[-\frac{1-y^\eta}{2}, \frac{1-y^\eta}{2}\right] & y^\eta > .66 \\ \mathcal{U}\left[-\frac{1-1.2y^\eta}{2}, \frac{1-1.2y^\eta}{2}\right] & y^\eta \in [.33, .66] \\ \mathcal{U}\left[-\frac{1-1.8y^\eta}{2}, \frac{1-1.8y^\eta}{2}\right] & y^\eta < .33 \end{cases} \tag{2}$$

3. it snows according to:
$$(x^s, y^s): \quad x^s \sim \mathcal{U}[-.5, 1] \wedge y^s \sim \mathcal{U}[-.1, 1.2] \tag{3}$$

4. christmas bulbs (x^β, y^β) hang on the tree branches, so they follow the same process (2) as needles do.

Having set out the building blocks of the tree I am in the position to define a stochastic X-mas tree.

Definition 1 (Stochastic X-mas Tree) *A stochastic X-mas Tree is a tuple* $\{(x^\tau, y^\tau), (x^\eta, y^\eta), (x^s, y^s), (x^\beta, y^\beta)\}$ *such that*

1. (x^τ, y^τ) *follows* (1)*;*
2. (x^η, y^η) *follows* (2)*;*
3. (x^s, y^s) *follows* (3)*;*
4. (x^β, y^β) *follows* (2)*.*

Simulation Results

In this section I simulate the X-mas tree, where the seed initializing the generator of (quasi-)random numbers is a function of the recipient, i.e. Duk converted to an integer number 292[1]. An example for readers who are unfamiliar with seeds, which shows how different names can generate different outcomse, is delegated to the appendix at the end of the note.

```
print(name)
```

```
## [1] "Duk"
```

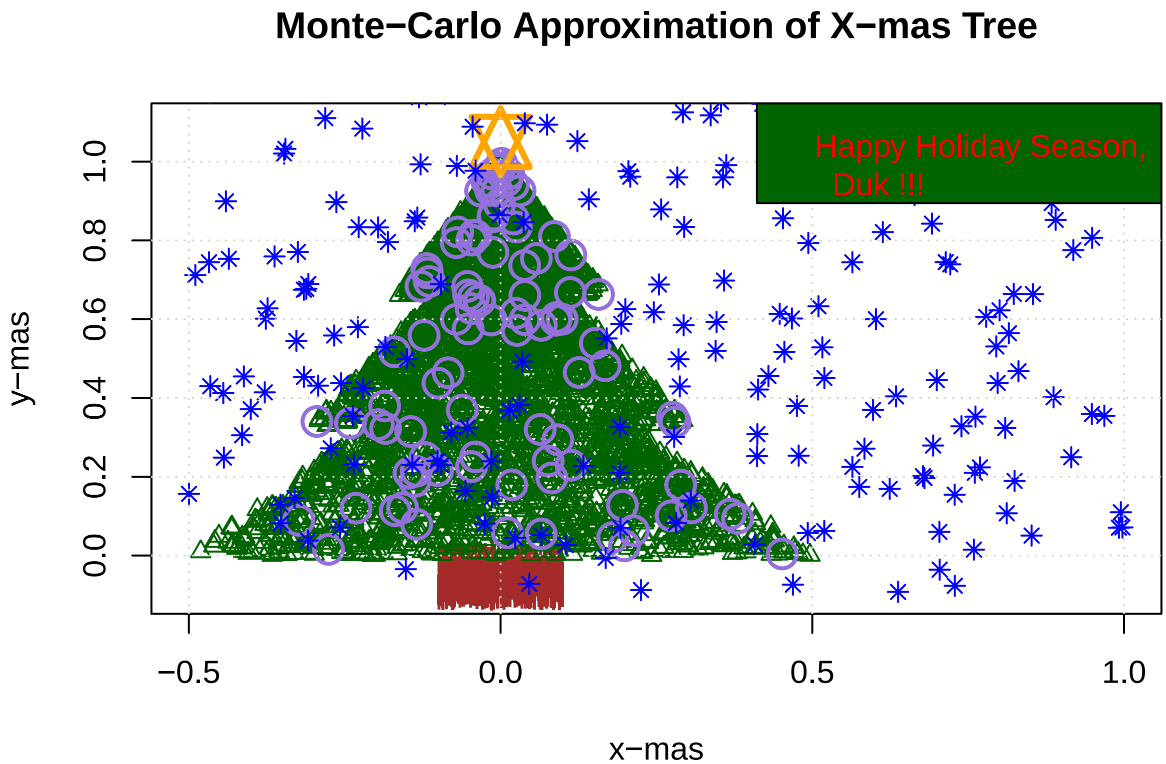
```
seed<-sum(utf8ToInt(name))
print(seed)
```

```
## [1] 292
```

```
set.seed(seed)
```

```
N<-4e3
y<-runif(N)

y_gen<-ifelse(y<.66, 1.2*y, y)
y_gen<-ifelse(y<.33, 1.8*y, y_gen)

x<- -(1-y_gen)/2+ runif(N)*(1-y_gen)

N_bombs<-1e2

y2<-sample(y, N_bombs)
y_gen2<-ifelse(y2<.66, 1.2*y2, y2)
y_gen2<-ifelse(y2<.33, 1.8*y2, y_gen2)
z<- -(1-y_gen2)/2+ runif(N_bombs)*(1-y_gen2)



plot(-.1+.2*runif(5e2), -.1+.1*runif(5e2), col='brown', pch="|",
     ylim=c(-.1, 1.1), xlim=c(-.5,1),
     main='Monte-Carlo Approximation of X-mas Tree',
     xlab='x-mas', ylab='y-mas')
points(y~x, col='darkgreen',  pch=2)
points(y2~z, col='mediumpurple', cex=2, lwd=2)

points(0, 1.05, cex=3, col='orange', pch=11, lwd=3)
points(-.5+1.5*runif(2e2), -.1+runif(2e2)*1.3, col='blue', pch=8)
grid()

legend("topright", bg='darkgreen',
       legend = paste('Happy Holiday Season,\n ',name,'!!!'),
       text.col='red')
```

[1] The drawback of my approach is that Duk will provide the same seed as kuD does.


Monte−Carlo Approximation of X−mas Tree
Happy Holiday Season,
Duk !!!
y−mas
1.0
0.8
0.6
0.4
0.2
0.0
−0.5
0.0
0.5
1.0
x−mas

Appendix

How different names generate different random numbers

```
name_example<-"John"
set.seed(sum(utf8ToInt(name_example)))
John<-cumsum(rnorm(1e3))
name_example<-"Jane"
set.seed(sum(utf8ToInt(name_example)))
Jane<-cumsum(rnorm(1e3))

plot(John, main=paste("Simulation for different names"),
     type='l', ylim=c(-30, 60), col='mediumpurple',
     lwd=2, ylab=paste('Value of the process') )
lines(Jane, col='orange', lwd=2)
grid()
legend('topleft', legend=c("John", "Jane"), col = c("mediumpurple", "orange"), lwd=2)
```