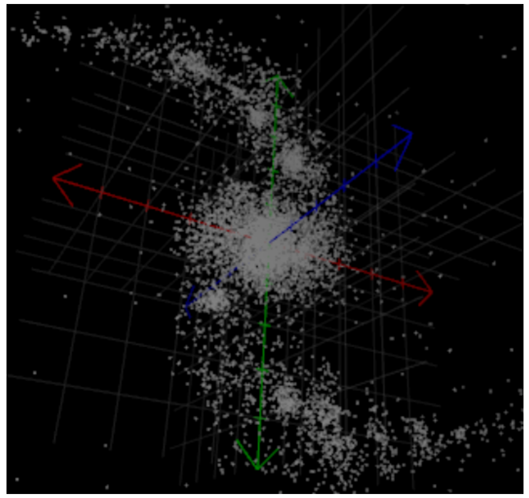
A Monad for Point Processes

Swaraj Dash & Sam Staton

University of Oxford

Proc. ACT 2020 tinyurl.com/act-pp-monad

Point processes: A point process (PP) is a random bag/multiset of points in some space X. In other words, a draw from a given region of a point process on X is a collection of potentially overlapping points where the number of points is a random variable.



Monads: A monadic interface for a type is a pair of functions, η and $\gg\!=$, that allow chaining together impure functions in a purely functional programming language.

Our work combines point process programming with monads. This enables us to construct interesting PPs by composing smaller PPs together in an algebraically sound manner.

We developed the point process monad in order to provide a functional programming equivalent for BLOG (`bayesianlogic.github.io`), a first-order probabilistic modelling language.

Giry monad: $G\mathbb{X} \equiv$ probability measures on $\mathbb{X}$

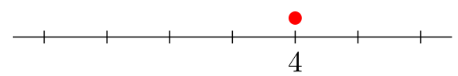
Bag monad: $B\mathbb{X} \equiv$ bags on $\mathbb{X}$

Point process monad: $\boxed{GB\mathbb{X} \equiv \text{ point processes on } \mathbb{X}}$

$$\eta : \mathbb{X} \to GB\mathbb{X}$$

Deterministically returns the singleton point with probability 1.

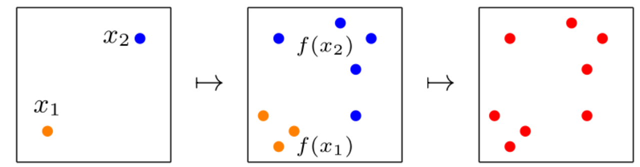
$$\eta(4) : GB\mathbb{N}$$



$$\gg\!= \,: GB\mathbb{X} \to (\mathbb{X} \to GB\mathbb{Y}) \to GB\mathbb{Y}$$

Simulating $p \gg\!= f$:

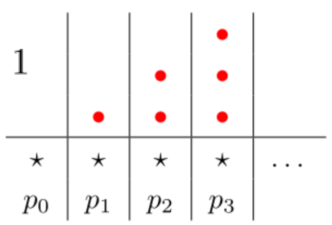
1. Draw a bag of points $[x_1, \ldots, x_n]$ from p
2. For each x_i, $f(x_i)$ is a point process
3. Draw a bag of points $[y_i^i, \ldots, y_{n_i}^i]$ from each $f(x_i)$
4. Return the union of the bags



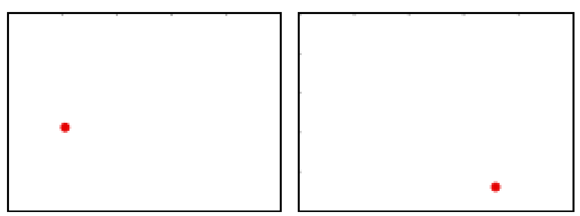
In Poisson PPs we observe a Poisson random number of points in a region proportional to its area. Given some region, a Poisson PP can be simulated by sampling a Poisson random number (`poisUnit`) and then uniformly distributing that many points inside the region (`unif`).

$\texttt{poisUnit} : GB\ \text{Unit}$

$$\sum_i p_i = 1$$

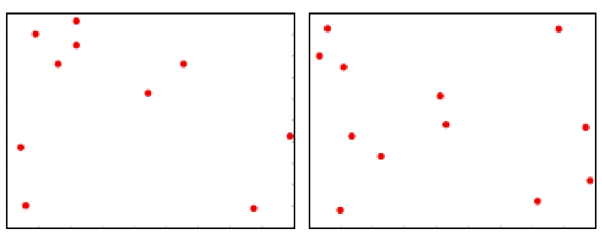


$\texttt{unif} : GB\ [0,1]^2$



$\texttt{poisSquare} : GB\ [0,1]^2$

$\texttt{poisSquare} = \texttt{poisUnit} \gg\!= \lambda\star \to \texttt{unif}$



Clustered point processes: we would like to have a Poisson number of clusters where each cluster gives rise to yet another Poisson number of trees normally distributed around the center. The clusters near the top should be more dense.

$\texttt{cluster} : GB\ [0,1]^2$

$\texttt{cluster} = \texttt{poisSquare} \quad \gg\!= \lambda(x,y) \to$

$\qquad \texttt{poisUnit'}(x,y) \gg\!= \lambda\star \quad \to \texttt{normal}\ (x,y)$