

LazyPPL

by Sam Staton
Hugo Paquet
and others



Exploring **laziness** and **types** in probabilistic programming

Laziness

- Unbounded point processes, infinite streams
dealt with entirely by Haskell
- Gaussian processes with infinite parameter spaces
explored lazily, entirely encapsulated

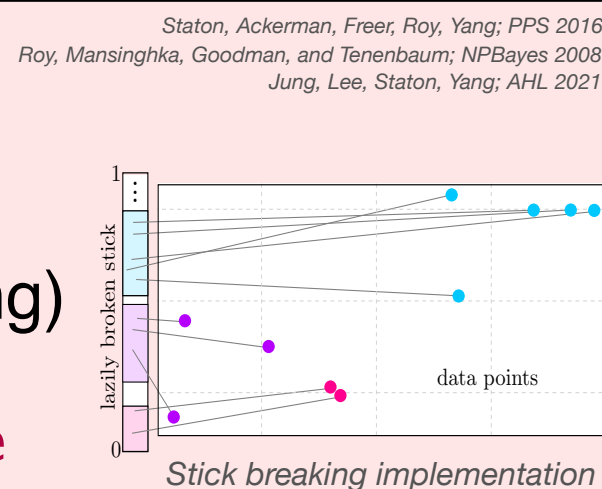
see also e.g. Koller, D. McAllester, and A. Pfeffer AAAI 1997
Kiselyov, Shan; DSL 2009
Bloem-Reddy, Mathieu, Foster, Rainforth, Teh,
Lomeli, Ge, Ghahramani; ApproxBayes 2017
Murray, Lundén, Kudlicka, Broman, Schön; AISTATS 2018

- `memoize :: (a -> Prob b) -> Prob (a -> b)`

Roy, Mansinghka, Goodman, and Tenenbaum; NPBates 2008
Wood, Archambeau, Gasthaus, James, Teh; ICML 2009

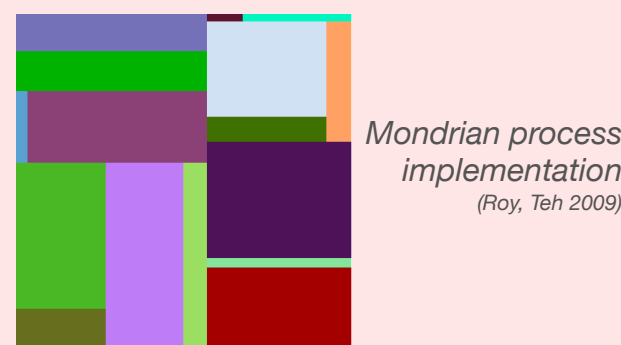
Abstract types

- Chinese restaurant process (clustering)
`newRest :: Prob Restaurant`
`newCust :: Restaurant -> Prob Table`



- Indian Buffet Process (feature extraction)
`newRest :: Prob Restaurant`
`newCust :: Restaurant -> Prob [Dish]`

- Graphs, matrices (relational model)
`newMatrix :: Prob Matrix`
`newRow :: Matrix -> Prob Row`
`newCol :: Matrix -> Prob Col`
`lookup :: Row -> Col -> Bool`

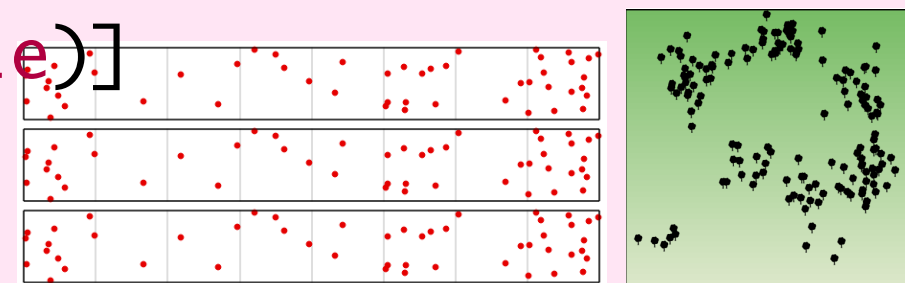
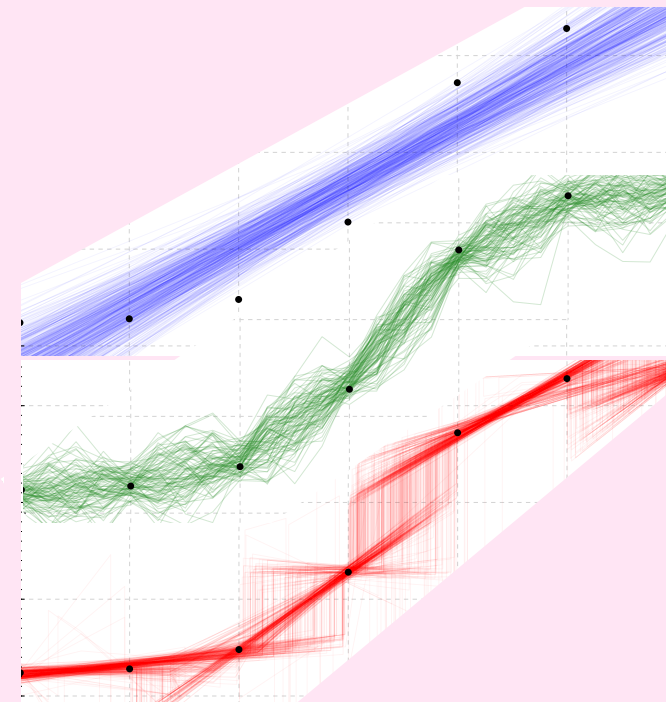


Types

- `Double -> (Prob Double)`
Parameterized distributions
- `Prob (Double -> Double)`
Random functions
- `Prob (Prob Bool)` , `Prob (Prob Double)`
Random distributions (beta, Dirichlet processes)
- `Prob [(Double, Double)]`
Point processes

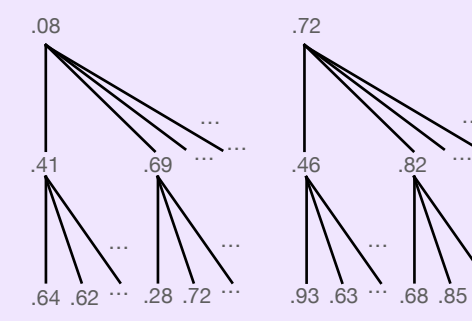
Heunen, Staton, Kammar, Wood; LICS 2017

Dash, Staton, ACT 2020 / MFPS 2021



Implementation

- Generative modelling is lazy (`Prob a`),
data / likelihood weighting is not lazy (`Meas a`)
- New lightweight MH!
 - Old idea (eager):
Wingate, Stuhlmüller, Goodman, AISTATS 2011
trace is a sequence of reals
LMH = randomly pick one dimension to change
 - New idea (lazy):
trace is an infinitely wide & deep forest
LMH = randomly mutate
each node independently



<https://bitbucket.org/samstaton/lazyppl>