An introduction to ProbProg

Ohad Kammar
University of Edinburgh
and
Denotational Limited

Huawei Tech-Talk

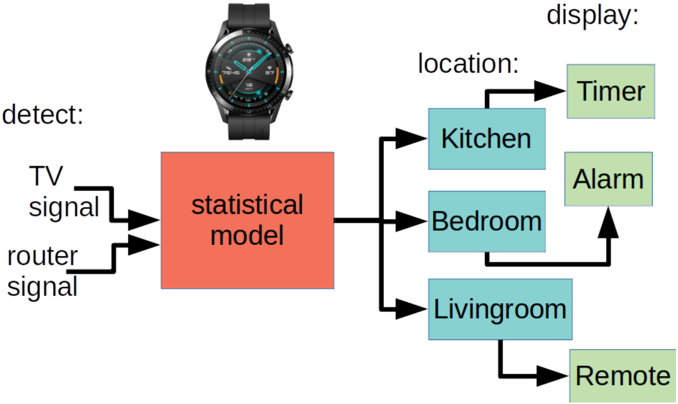
HUAWEI

14 January 2021

THE UNIVERSITY of EDINBURGH informatics lfcs Laboratory for Foundations of Computer Science BayesCentre

supported by: THE ROYAL SOCIETY The Alan Turing Institute Facebook Research

Spatial-context inference

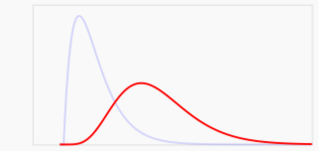


Spatial-context inference

```haskell
data Location = Kitchen|Livingroom|Bedroom|Study|Bath
data Device = TV | Fridge | Router | Printer | Tablet
type SensorData = (Device, Double)

dataset :: [SensorData]
dataset = [(Fridge, 3.25)
          ,(TV    , 2.25)
          ,(Router, 3.25)
          ,(Tablet, 1.75)]
```

Spatial-context inference

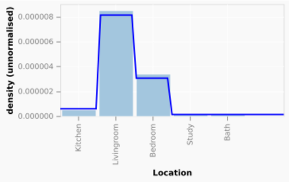
```
data Location = Kitchen|Livingroom|Bedroom|Study|Bath
data Device = TV | Fridge | Router | Printer | Tablet
type SensorData = (Device, Double)

dataset :: [SensorData]
-- condition: d ~ Γ(x, 1.0)
proximity :: Double -> Double -> Dist ()
proximity d x = condition(d, gammaDistr x 1.0)
-- condition: devices from close in dataset are close
closeTo :: [SensorData] ->  [Device] -> Dist ()
closeTo dataset close = do
  forM dataset $ \(dev,dist) ->
    if dev `elem` close
    then proximity 2.0 dist
    else proximity 6.0 dist
  pure ()
```

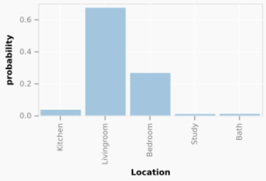


density of $\Gamma(2.0, 1.0)$ and $\Gamma(6.0, 1.0)$

Spatial-context inference

```
dataset :: [SensorData]
-- condition: d ~ Γ(x, 1.0)
proximity :: Double -> Double -> Dist ()
-- condition: devices from close in dataset are close
closeTo :: [SensorData] ->  [Device] -> Dist ()
spatialModel :: [SensorData] -> Dist Location
spatialModel dataset = do
  loc <- uniformD locations
  dataset `closeTo` case loc of
    Kitchen    -> [Fridge]
    Livingroom -> [TV, Router]
    Bedroom    -> [Tablet]
    Study      -> [Printer]
    Bath       -> []
  return loc
```





Features

- Meaningful with small data
  Data can be expensive
- Model is descriptive
  Accessible to domain experts
- Deals with missing data
  `Printer` was missing
- Quantifies uncertainty
  Uncertainty is mathematically meaningful
- Application-embedded
  Stay within ecosystem, personal data stays on device

What is ProbProg?

- Machine interpretable statistical modelling
- ProbProg $=$ PL $+$ sampling $+$ conditioning $+$ inference
- common language for:
  - domain experts
  - algorithmic statisticians
  - performance engineers
  - computer

Talk Structure

- ProbProg "Hello, World!"
- What are we computing?
- Inference: compilation schemes for ProbProg

Apology: many ProbPLs out there, we'll mention only a few

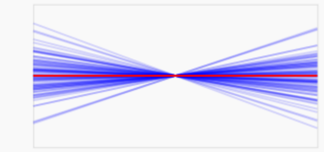
Request: only a few slides, so so please interact!

“Hello, World!”

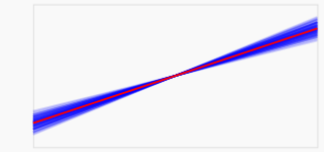
Homogeneous linear regression

```
model :: Dist (Double -> Double)
model = do
  a <- normal 0.0 1.0
  let f x = a * x
  condition (f 2.0, normal 6.0 0.25)
  return f
```

Prior:



Posterior:



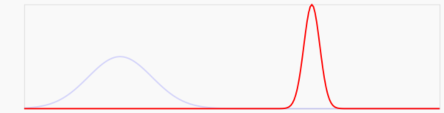
What are we computing? (for statisticians)

$$\text{prior} \in \text{Dist}\ (X \times \Theta) \xrightarrow{\text{Hypothesis}} H \in \text{Dist}\ \Theta \ni \theta$$

$$\text{prior} \in \text{Dist}\ ((\mathbb{R} \to \mathbb{R}) \times \mathbb{R}) \xrightarrow{\text{Hypothesis}} H \in \text{Dist}\ \mathbb{R} \ni \theta$$

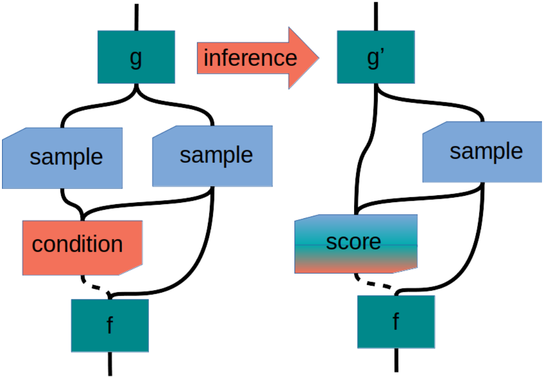
Bayes's Law: posterior $\propto \frac{\mathrm{d}\theta}{\mathrm{d}H} \cdot$ prior

hypothesis and observation density wrt Lebesgue



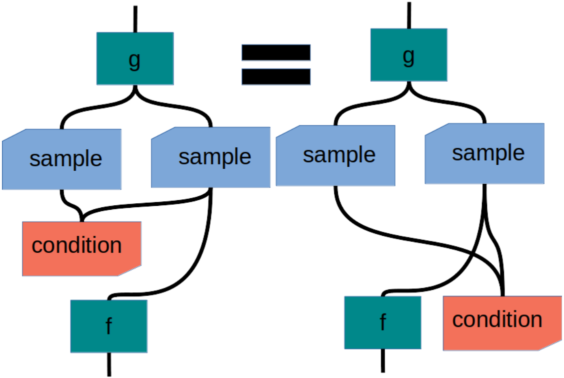
density of observation wrt hypothesis

What are we computing? (for compiler engineers)



Invariant: $\frac{\mathrm{d}\theta}{\mathrm{d}H} \cdot$ prior

What are we computing? (for compiler engineers)



Invariant: $\frac{\mathrm{d}\theta}{\mathrm{d}H} \cdot$ prior

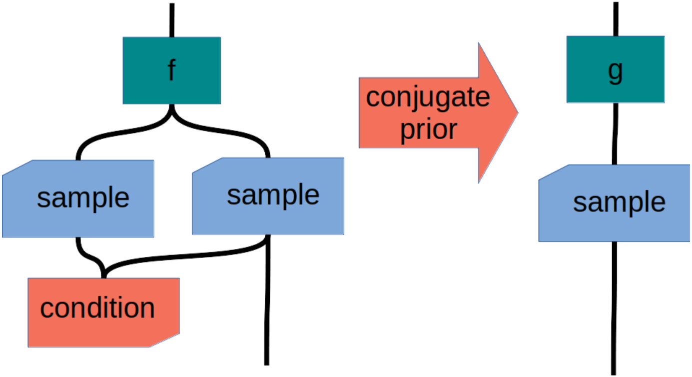
Why is it difficult?

$$\frac{\mathrm{d}\theta}{\mathrm{d}H} \cdot \mathsf{prior} = \lambda f.\int_X \mathsf{prior}(\mathrm{d}x\,\mathrm{d}t)\frac{\mathrm{d}\theta}{\mathrm{d}H}(t) \cdot f(x)$$

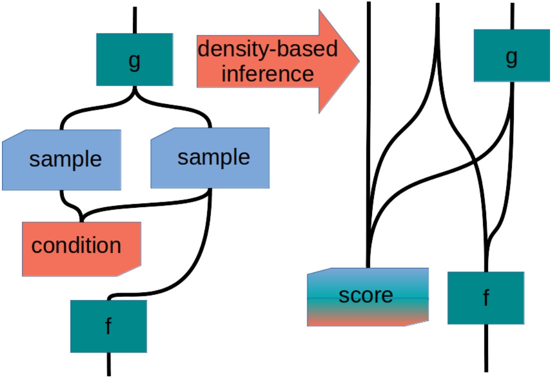
- integrals are hard!
- correct densities need care
- each fragment is an arbitrary (pure) program!
- correct answer isn't always clear

Compilation schemes

Peephole optimisations

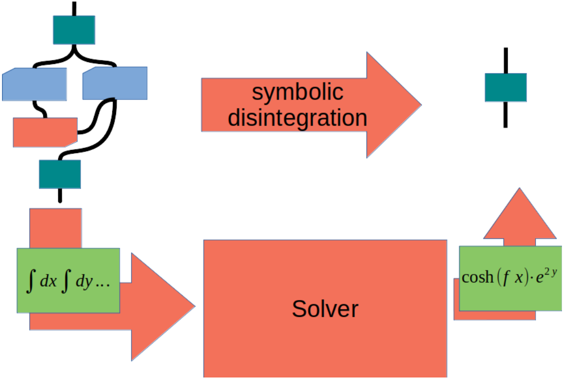


Compilation schemes



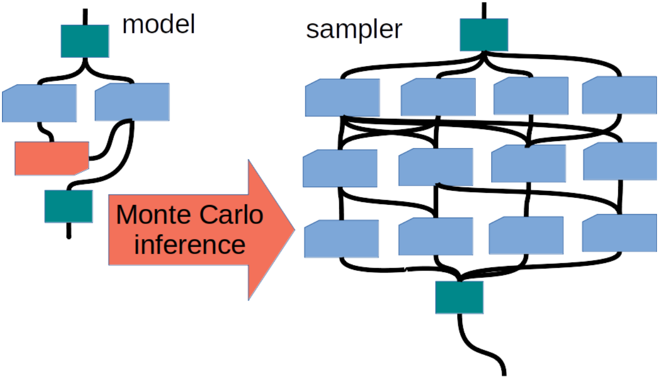
E.g.: Stan, SlicStan + enumeration of discrete variables
[Gorinova et al. POPL'19, and ongoing work]

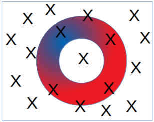
Compilation schemes



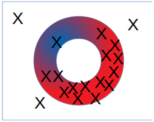
Hakaru [Narayanan et al. FLOPS'16]
Psi [Gehr-Misailovic-Vechev, CAV'16]

Compilation schemes





Baseline:
importance
sampling



statistically
efficient
sampling

Most systems. Anglican and Church
are purely Monte Carlo

Cost model?

No silver bullet!

- Inference is non-computable [Ackerman-Freer-Roy LICS'11]
- Steps towards bespoke inference:
  - Pyro: non-standard interpretations (effect handlers)
  - Monad-Bayes: modular inference building blocks (monad transformers)
  - Gen: Programmable inference (guides)

Summary

- ProbProg; machine-readable formalism for statistical modelling
- Key features:
  - Meaningful with small data
  - Model is descriptive
  - Deals with missing data
  - Quantifies uncertainty
  - Application-embedded
- Non-computable: need bespoke inference