Probabilistic programming

John Winn and Tom Minka

Machine Learning Summer School

September 2009

Microsoft® Research Cambridge

Goals of Probabilistic Programming

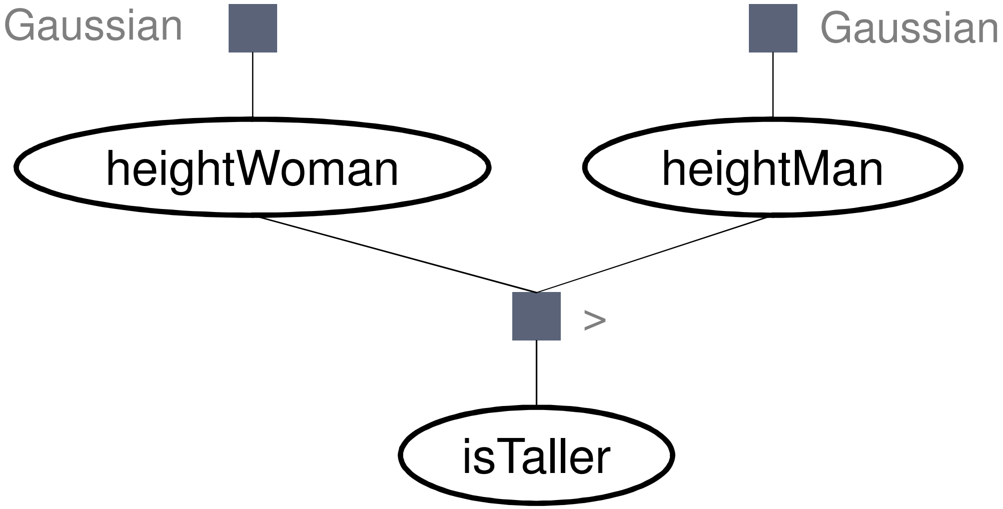
- Make it easier to do probabilistic inference in custom models
- If you can write the model as a program, you can do inference on it
- Not limited by graphical notation
- Libraries of models can be built up and shared

A big area of research!

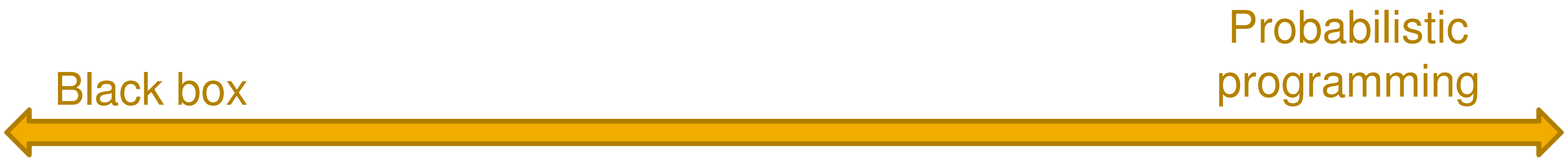
Heights example

- Suppose we take a woman at random and a man at random from the UK population
- The woman turns out to be taller than the man
- What is the probability of this event?
- What is the posterior distribution over the woman’s height?
- What is the posterior distribution over the man’s height?

Heights example



Machine learning software



Black box

Probabilistic programming

Weka

Bayes Net software (e.g Hugin)

SVM libraries

gR

BNT

VIBES

BUGS

Infer.NET

Church

Hierarchical Bayes Compiler

CSOFT probabilistic language

- A representation language for probabilistic models.
- Takes C# and adds support for:
  - random variables
  - constraints on variables
  - inference
- Can be embedded in ordinary C# to allow integration of deterministic + stochastic code

CSoft – random variables

- Normal variables have a fixed single value.

  e.g. `int length=6,`
  `bool visible=true.`

- Random variables have uncertain value specified by a probability distribution.

  e.g. `int length = random(Uniform(0,10))`
  `bool visible = random(Bernoulli(0.8)).`

- Introduce random operator which means ‘is distributed as’.

CSOFT –constraints

- We can define constraints on random variables, e.g.

```
constrain(visible==true)
constrain(length==4)
constrain(length>0)
constrain(i==j)
```

- The constrain(b) operator means ‘we constrain b to be true’.

Csoft – inference

- The infer() operator gives the posterior distribution of one or more random variables.
- Example:

```
int i = random(Uniform(1,10));
bool b = (i*i>50);
Dist bdist = infer(b);//Bernoulli(0.3)
```

- Output of infer() is always *deterministic* even when input is *random*.

Heights example in CSOFT

```
double heightMan = random(Gaussian(177,64));
double heightWoman = random(Gaussian(164,64));
Bernoulli dist = infer(heightWoman > heightMan);
constrain(heightWoman > heightMan);
Gaussian distWoman = infer(heightWoman);
Gaussian distMan = infer(heightMan);
```

- First infer is computed without the constraint
- Later infers are computed with the constraint

Sampling interpretation

- Imagine running the program many times, where
  - `random(dist)` is an ordinary function that draws a random number from dist
  - `constrain(b)` stops the run if b is not true
  - `infer(x)` collects the value of x into a persistent memory (one for each use of infer in the program)
    - If enough x's have been stored, return their distribution
    - Otherwise stop the run (i.e. wait until enough samples are collected)
- This defines the meaning of a Csoft program

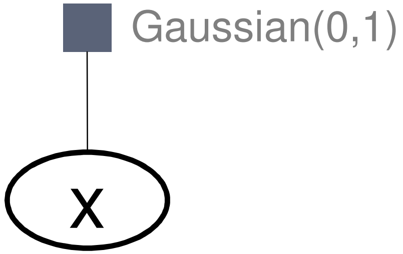
Probabilistic programs & graphical models

Random variables

Probabilistic program

```
double x = random(Gaussian(0,1));
```

Graphical model

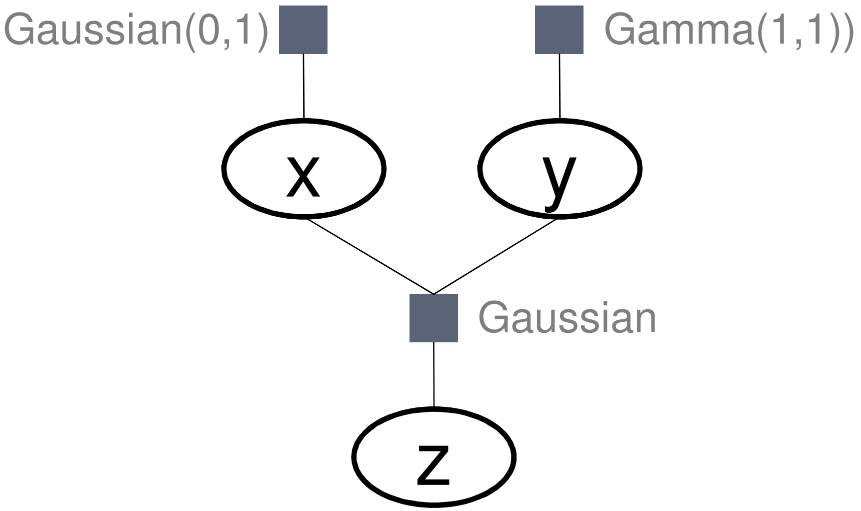


Bayesian networks

Probabilistic program

```
double x = random(Gaussian(0,1));
double y = random(Gamma(1,1));
double z = random(Gaussian(x,y));
```

Graphical model

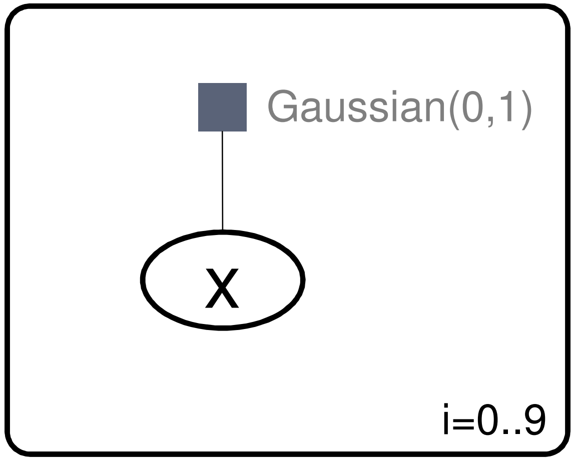


Loops/plates

Probabilistic program

```
for(int i=0;i<10;i++) {
  double x = random(Gaussian(0,1));
}
```

Graphical model

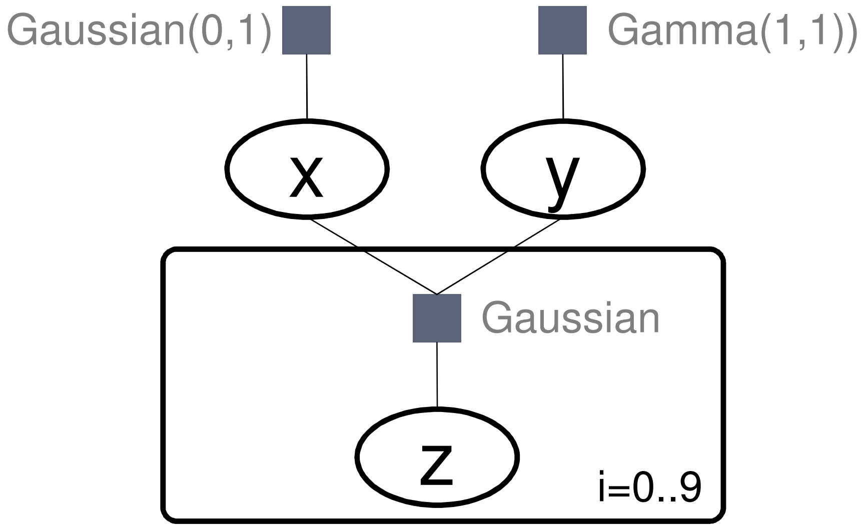


Loops/plates II

Probabilistic program

```
double x = random(Gaussian(0,1));
double y = random(Gamma(1,1));
for(int i=0;i<10;i++) {
  double z = random(Gaussian(x,y));
}
```

Graphical model

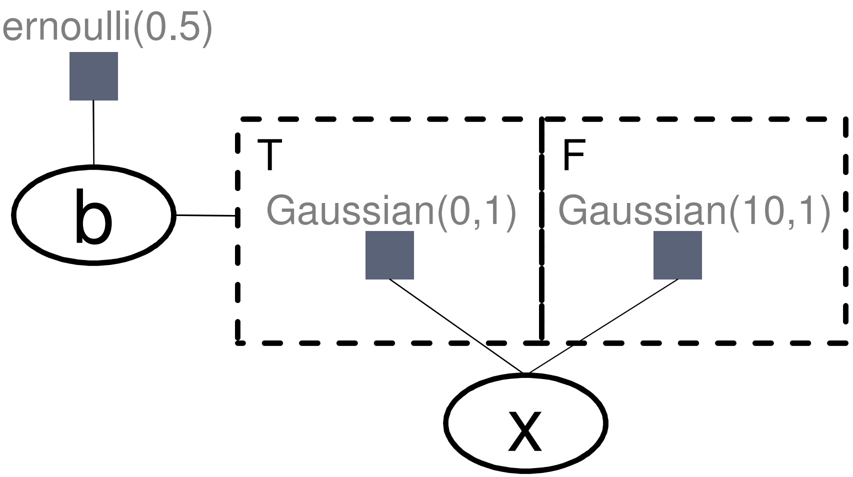


If statement/gates

Probabilistic program

```
bool b = random(Bernoulli(0.5)); double x;
if (b) {
  x = random(Gaussian(0,1));
} else {
  x = random(Gaussian(10,1));
}
```

Graphical model



Gates (Minka and Winn, NIPS 2008)

Other language features

Probabilistic program

- Functions/recursion
- Indexing
- Jagged arrays
- Mutation: `x=x+1`
- Objects
- …

Graphical model

No common equivalent

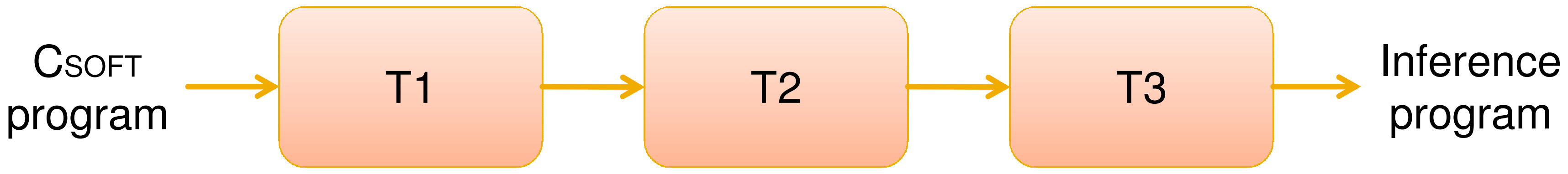
Needs of Probabilistic Programming

- Flexible and general inference algorithms
- Modelling constructs that integrate nicely with inference
  - E.g. *Gates* (Minka and Winn, NIPS 2008)
- Compiler technology for probabilistic constructs
- Automatic scheduling of fixed-point updates
- Automatic parallelization
- …

Probabilistic programming in Infer.NET

Infer.NET

- C*ompiles* probabilistic programs into inference code.
- No in-memory factor graphs = no overhead
- Consists of a chain of code transformations:



CSOFT program

T1

T2

T3

Inference program

- Calling `infer` invokes this chain automatically

Infer.NET

- Model is specified using C#, with operators overloaded to look like Csoft
- C# code is internally converted into Csoft
- Inference compiler works only with Csoft
- In a future version, it will be possible to program in Csoft directly
- Free for academic use
  http://research.microsoft.com/infernet

Random variables in Infer.NET

Probabilistic program

```
double x = random(Gaussian(0,1));
```

C# code

```
Variable<double> x = Variable.Gaussian(0,1);
```

Bayesian networks

Probabilistic program

```
double x = random(Gaussian(0,1));
double y = random(Gamma(1,1));
double z = random(Gaussian(x,y));
```

C# code

```
Variable<double> x = Variable.Gaussian(0,1);
Variable<double> y = Variable.Gamma(1,1);
Variable<double> z = Variable.Gaussian(x,y);
```

Inference in Infer.NET

Probabilistic program

```
double x = random(Gaussian(0,1));
Dist xdist = infer(x);
```

C# code

```
Variable<double> x = Variable.Gaussian(0,1);
InferenceEngine engine = new InferenceEngine();
IDistribution<double> xdist = engine.Infer(x);
// or
Gaussian xdist = engine.Infer<Gaussian>(x);
```

Loops/plates

Probabilistic program

```
for(int i=0;i<10;i++) {
  double x = random(Gaussian(0,1));
}
```

C# code

```
Range i = new Range(10);
using(Variable.ForEach(i)) {
  Variable<double> x = Variable.Gaussian(0,1);
}
```

Loops/plates II

Probabilistic program

```
double[] x = new double[10];
for(int i=0;i<10;i++) {
  x[i] = random(Gaussian(0,1));
}
```

C# code

```
Range i = new Range(10);
VariableArray<double> x = Variable.Array<double>(i);
using(Variable.ForEach(i)) {
  x[i] = Variable.Gaussian(0,1);
}
```

If statement/gates

Probabilistic program

```
bool b = random(Bernoulli(0.5)); double x;
if (b) {
  x = random(Gaussian(0,1));
} else {
  x = random(Gaussian(10,1));
}
```

C# code

```
Variable<bool> b = Variable.Bernoulli(0.5);
Variable<double> x = Variable.New<double>();
using(Variable.If(b)) {
  x.SetTo( Variable.Gaussian(0,1) );
} using(Variable.IfNot(b)) {
  x.SetTo( Variable.Gaussian(10,1) );
}
```

Indexing by random integer

Probabilistic program

```
bool[] b = new bool[2] { true, false };
int i = random(Discrete(0.4,0.6));
bool c = b[i];    // Bernoulli(0.4)
```

C# code

```
VariableArray<bool> b = Variable.Array<bool>(range);
b.ObservedValue = new bool[2] { true, false };
Variable<int> i = Variable.Discrete(range,0.4,0.6);
Variable<bool> c = Variable.New<bool>();
using(Variable.Switch(i)) {
  c.SetTo( b[i] );
}
```

On to the practical!

http://mlg.eng.cam.ac.uk/mlss09/material.htm