To Reify or not To Reify

is the graph is everything?

BUGS PPL uses the graph as an organization principle

Statistical model described by a joint probability distribution function

PDF is a complex function

Can be represented as a graph

Graph shows factorization of PDF

BUGS does local computation of the graph

Two types of computation

Meta computation — computation about what computation to do

Numerical computation – involving calculating probability distributions or summary statistics (conjugacy)

The structure of the graph gives conditional independence relations

BUGS works out the parents of each node in graph

Given the parents possible to find the children of each node in graph

Then have markov blanket

Can do Gibbs type algorithms

Differentiating the joint PDF

Equivalent to differentiating conditional distributions

More efficient differentiation algorithms?

Can do HMC and ADVI

PPLs and source code transformation

Derive imperative code from the PPL specification of model

Easy if only want to be able to calculate the joint PDF

For HMC and ADVI need derivatives of the joint PDF

Various ways of calculating these derivatives

PPLs and source code transformation

No explicit graph

No markov blanket

How to do Gibbs type algorithms?

Good points in not having explicit graph

PPLs and source code transformation

Could calculate the conditional distributions by executing the imperative code that calculates the joint PDF

This is wasteful for many models

Basic idea is to add control structures to the imperative code that evaluates the joint PDF so that only certain statements get executed

Modifying the imperative code for the joint PDF

Take a two level approach:

Firstly add a case statement so that each statement in the PPL gives rise to a branch in the case statement.

Secondly for loop statements in the PPL specify which elements of the loop are need by a vector of integers

Modifying the imperative code for the joint PDF

Give the procedure to calculate the joint PDF an argument consisting of a vector of vector of integers.

The first integer of each vector of integers tells which statement in the PDF need executing, the following integers what elements of a loop statement need to be executed.

The vector of vector of integers tells how to evaluate the conditional distribution

Modifying the imperative code for the joint PDF

Can use the same idea to write code for conjugacy

Can use the same idea to write classification code

Need to impose an order on the statements in the PPL or more precisely the order of statements in the imperative code.

Would prefer to rearrange the PPL so that statements in needed order.

How to find the integer arrays

Very simple algorithm

Based on abstract interpretation

Maybe not very efficient for big models

Embarrassingly parallel

Early results

Only one model working – the Rats model

Inference is 25% faster than the BUGS software

100K samples takes 2s compared with 2.5 in BUGS

For a bigger model (Methadone) with 400K observations and 20K parameters finding the arrays of integers takes 64s on one core and the storage required for the model is around half that used by BUGS

Early results

The prototype software makes heavy use of the BUGS software.

Source code is produced by walking the AST of the BUGS software

Source code is produced in the Component Pascal language. Should be possible to produce code in other languages for example Julia or C.

The source code produced is readable and well structured but verbose