Title :
Bayesian Inference and Let-Term Transformations

Topic :
probabilistic programming, bayesian inference, lambda-calculus, compilation of functional programs.

City and country :
Paris - France (IRIF)

Team or project in the lab :
PPS pole at IRIF (participation to the ANR project PPS)

Name and mail of the advisor :
Pagani Michele, pagani@irif.fr
(In case of goal A below, possible co-advisor : Pierre-Evariste Dagand, dagand@irif.fr)

Name and mail of the head of the department :
Magniez Frederic, magniez@irif.fr

General presentation of the topic (roughly 5 to 10 lines)

This internship is about *probabilistic programming* — a programming paradigm in which programs represent statistical models computing probabilistic distributions operationally. A major benefit of this approach is to describe inference methods as program transformations, to ease use and reuse, and to allow transfers of techniques between the programming-languages and machine-learning communities.

More precisely, this internship will focus on a special class of inference algorithms based on Bayesian networks, such as the Variable Elimination algorithm, and it will study how to express them as program transformations of a special class of let-terms, corresponding to a fragment of simply typed lambda-calculus.

Objective of the internship (roughly 10 to 20 lines)

Bayesian inference aims to provide efficient algorithms to update the likelihood of a random variable according to certain observations (e.g. refreshing the likelihood of having a good grade in a computer science class, knowing how many hours a student spent coding for exam preparation). A crucial technique is to factor polynomials into products following some diagrammatical representations of random variables, such as Bayesian networks and factor graphs (see chapter 8 of [1] or chapters 4-7 of [2]).

In a recent preprint [3], we show how the Variable Elimination algorithm can be expressed as program transformations (basically, rewritings over lambda-terms), which transform a global definition of a variable into a local definition by swapping and nesting let-in expressions. The goal of this internship is quite flexible and can be developed following practical and/or theoretical directions :

A) practical goals: (i) to implement the transformations described in [3] on a core language consisting basically in a simply-typed lambda-calculus and test it on some relevant examples; (ii) to implement some of the heuristics given in chapter 9 of [2] for approaching an optimal transformation;

B) theoretical goals: (i) to compare the transformations in [3] with the “let-floating” transformations adopted in optimising compilers of functional languages, as e.g. in [4,5], looking in particular for relevant examples where the two techniques diverge; (ii) to extend this approach to other inference algorithm, like the Message Passing (or Belief Propagation) algorithm.

Bibliographic references

[1] C. M. Bishop, "Pattern Recognition and Machine Learning", Springer 2006
[2] A. Darwiche, “Model and Reasoning with Bayesian Networks”, Cambridge University Press 2009
[3] C. Faggian, T. Ehrhard, M. Pagani, “The Variable Elimination Algorithm Wears Linear Lambda-Calculus”, available on request.
[4] S. P. Jones, W. Partain, A. Santos, “Let-floating: Moving Bindings to Give Faster Programs”, https://dl.acm.org/doi/10.1145/232629.232630
[5] S. P. Jones, A. Santos, “A transformation-Based Optimiser for Haskell” https://www.sciencedirect.com/science/article/pii/S0167642397000294?via%3Dihub

Expected ability of the student

Expertise in two or more of the following topics: functional programming, lambda-calculus, linear logic, bayesian inference, probabilistic graphical models, statistical machine learning, statistics.