Seminar *Probabilistic Programming*

Joost-Pieter Katoen et al.

Winter Semester 2022/23

Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Outline

Overview

Probabilistic Programming

Probabilistic programs

Probabilistic programs = classical programs + probabilistic choice + conditioning

- choice: “execute program P with probability $\frac{2}{3}$ and Q with $\frac{1}{3}$”
- conditioning: “observe that value of variable x is positive”
- describe posterior probability distributions over variable output values

Applications

- Randomised algorithms (e.g., randomised Quicksort)
- Computer vision (e.g., image generation)
- Security
- Biology, coding theory, cryptographic protocols, machine learning, quantum computing, reliability analysis, ...

An Example

Virus infection

```
bool aliceInfectious = true
bool bobInfected = false
while aliceInfectious {
  prob 0.1 {
    bobInfected = true
  }
  prob 0.6 {
    aliceInfectious = false
  }
}
```

- What is the probability of Bob becoming infected?
- How long is Alice likely to be infectious?

Areas Covered in this Seminar

Topic areas

- Semantics [handbook+papers]
  - mathematical approaches to formally define precise meaning of programs
- Verification [handbook+papers]
  - providing correctness proofs for programs (termination, program equivalence...)
- Programming languages [handbook+papers]
  - new probabilistic and probabilistic extensions of classical programming languages
- Static analysis [papers]
  - slicing, resource consumption, ... based on source code

Outline

Aims of this Seminar

Goals

Aims of this seminar

- **Independent understanding** of a scientific topic
- Acquiring, reading and understanding **scientific literature**
  - given references sufficient in most cases
- Writing of your **own report** on this topic
  - far more that just a translation/rewording
  - usually an **“extended subset”** of original literature

    “subset”: present core ideas and omit too specific details (e.g., related work or optimisations)
    “extended”: more extensive explanations, examples, ...
    discuss contents with supervisor!
- **Oral presentation** of your results
  - can be “proper subset” of report
  - generally: less (detailed) definitions/proofs and more examples

Requirements on Report

Your report

- Independent writing of a report of **12–15 pages**
- First milestone: **detailed outline**
  - not: “1. Introduction/2. Main part/3. Conclusions”
  - rather: overview of structure (section headers, main definitions/theorems) and initial part of main section (one page)
- **Complete** set of references to all consulted literature
- **Correct citation** of important literature
- **Plagiarism**: taking text blocks (from literature or web) without source indication causes immediate **exclusion from this seminar**
- Font size **12pt** with “standard” page layout
  - **LaTeX template** will be made available on seminar web page
- **Language**: German or English
- We expect the **correct usage** of spelling and grammar
  - $\geq$ 10 errors per page $\Longrightarrow$ abortion of correction

Requirements on Talk

Your talk

- Talk of **30 minutes**
- Available: projector, presenter, [laptop]
- Focus your talk on the **audience**
- **Descriptive** slides:
  - $\leq$ 15 lines of text
  - use (base) colors in a useful manner
  - number your slides
- **Language:** German or English
- No spelling mistakes please!
- Finish **in time**. Overtime is bad
- Ask for **questions**
- Have **backup slides** ready for expected questions
- **LaTeX/beamer template** will be made available on seminar web page

Outline

Important Dates

Deadlines

- October 20: Topic preferences due
- November 21: Detailed outline due
- December 22: Full report due
- January 13: Presentation slides due
- February 6–9 (?): Seminar talks

Important

Missing a deadline causes **immediate exclusion** from the seminar

Important Dates

Selecting Your Topic

Procedure

- You obtain a list of topics of this seminar.
- Classified according to BSc/MSc level.
  - MSc students please choose at least one “M-only” topic
- Indicate at least three topics of interest at

  `https://terminplaner4.dfn.de/hQJa8kwGFBxnnLbc`
- Can indicate preference in the comments field
- We do our best to find an adequate topic-student assignment.
  - disclaimer: no guarantee for an optimal solution
- Assignment will be published on web page Oct 26.
- Then also your **supervisor** will be indicated.
- Please give language preference (unsure $\Longrightarrow$ German).

Withdrawal

- You have up to **three weeks** to refrain from participating in this seminar.
- Later cancellation (by you or by us) causes a **not passed** for this seminar and reduces your (three) possibilities by one.

Outline

Semantics

Semantics

1. *Semantics of Probabilistic Programs* (B/M)
   - semantics of an imperative language with discrete and continuous distributions
   - operational: reduction (Markov chain)
   - denotational: transformation of probability distributions
2. *Probabilistic Programs as Measures* (M)
   - compositional measure-theoretic semantics for statistical modelling languages
3. *Probabilistic λ-Calculi* (B/M)
   - probabilistic version of λ-calculus (core language for functional programming languages)
   - two variations: randomised and Bayesian λ-calculi

Verification

Verification

4. *Probabilistic Couplings from Program Logics* (M)
   - use of couplings for verifying probabilistic programs
   - enables clean proofs of probabilistic relational properties (equivalence, convergence, ...)
5. *Expected Runtime Analysis* (B)
   - weakest precondition calculus (á la Dijkstra) for finding expected run-times of probabilistic program
   - fully automatable for Bayesian networks
6. *Termination Analysis* (M)
   - methods for checking (positive) almost-sure termination
   - based on martingales (special type of stochastic process)
7. *Quantitative Analysis* (M)
   - quantitative analysis of probabilistic programs
   - uses concentration of measure inequalities to characterize how functions of random variables deviate from expected value

Programming Languages

Programming languages

8. *Luck: A Probabilistic Language for Testing* (B)
   - probabilistic domain-specific language for test generation
   - framework for property-directed testing of functional programs
9. *Rely: Programming Unreliable Hardware* (B)
   - programming language for reasoning about the probability that a program produces the correct result when executed on unreliable hardware
   - application: approximate computing
10. *Tabular: Probabilistic Inference from the Spreadsheet* (B)
    - domain-specific programming language for expressing probabilistic models and performing probabilistic inference over relational data
    - programs and data stored as spreadsheet tables (MS Excel interface)

Program Analysis

Program analysis

11. *Automated quantized inference for probabilistic programs with AQUA* (M)
    - introduces a tool and technique for exact inference
12. *Checking program equivalence* (M)
    - goal: decidability and complexity of equivalence checking
    - application: cryptography
13. *Slicing* (B/M)
    - aim: simplify programs before analysis
    - designing and proving correctness of slicing
14. *Learning termination proofs* (B/M)
    - aim: prove termination of a probabilistic program
    - how: using learning termination certificates by neural networks

The Topics

Semantics and inference

Semantics

15. *A separation logic for negative dependence* (M)
    - aim: a logic for reasoning about statistical independence
    - how: using a variant of separation logic
16. *Semantics for Probabilistic Programs with Nondeterminism* (B/M)
    - goal: add non-determinism to probabilistic programs
    - aim: how to do this semantically?
17. *Scenic* (B)
    - aim: a language for scene generation in computer vision
    - how: can learning of neural networks be improved?
18. *PSI: Exact inference for higher-order probabilistic programs.* (B)
    - aim: exact inference
    - how: using numerical mathematics
19. *Posterior inference in universal probabilistic programming.* (M)
    - aim: approximate posterior distribution using bounds
    - approach: using type systems

Outline

Some Final Hints

Hints

- Take your time to **understand** your literature.
- Be **proactive**! Look for **additional** literature and information.
- Discuss the content of your report with other students.
- Be **proactive**! Contact your supervisor **on time**.
- **Prepare** the meeting(s) with your supervisor.
- Forget the idea that you can prepare a talk in a day or two.

We wish you success and look forward to an enjoyable and high-quality seminar!