

STA4516

Topics in Probabilistic Programming

Instructor

Prof Daniel M. Roy

daniel.roy@utoronto.ca (please include "STA4516" in your email's subject line or body)

Office hours: SS 6026C, by appointment.

Time and Location

Date/Time: Fall 2015, Tuesdays, 2--5pm from Oct. 27 through Dec. 8

Room: SS 2116 (Sidney Smith Hall)

Overview

Probabilistic programming is an emerging area of machine learning, statistics, and computer science, and can be characterized as the systematic study of algorithmic processes that describe and transform uncertainty. Probabilistic programming languages give users the ability to describe complex probabilistic models with code, rather than formulas, while universal inference engines automate the task of implementing inference algorithms for models described by probabilistic code. This class will discuss the key ideas behind probabilistic programming languages and systems, as well as give students a glimpse at the theoretical foundations of probabilistic programming itself.

Pre-requisites

Mathematical maturity and some background in probability/measure theory, functional programming, and analysis recommended.

Structure of the course

The course will be structured around weekly readings and student presentations of papers. Each week there will be a lecture on a new subject.

- Week 1 - Oct 27
 - Student background survey
 - High level overview of course
- Week 2 - Nov 3
 - Representing Conditional Independence
 - Exact and Lightweight MCMC Algorithms
- No class on Nov 10 -----
- Week 3 - Nov 17
 - Students should prepare by having completed the ProbMods and DIPPL online tutorials, in addition to the papers being presented by students.
 - Representing Bayesian nonparametric models
 - **Student Presentations**

- Week 4 - Nov 24
 - Students should prepare by starting to read the paper by Arno Pauly, using Wikipedia to look up unfamiliar terms. Getting through the first 3-4 pages would be an excellent start.
 - Introduction to Computable Analysis
 - **Student Presentations**
- Week 5 - Dec 1
 - Computable Probability Theory
 - **Student Presentations**
- Week 6 - Dec 8
 - Open problems
 - **Student Presentations**

See *Reading* (below) for papers associated with each week's topic.

Grading

For students taking the class for credit, their grade will be based on:

- 33% class participation, including scribing*;
- 33% in-class presentation of a paper;
- 33% project report.

Students auditing will be expected to participate actively (by asking and answering questions, and doing weekly readings) and even *scribe*, if necessary. Students auditing who would like to present a paper are welcome to do so, time permitting. Everyone is welcome to submit a research project.

Class participation

Students should remain attentive, ask questions, and answer questions during class. Participation also involves reading over those papers being presented that week, and coming with questions. By Week 3, students should have completed the online tutorials (ProbMods and DIPPL.) Class participation also involves scribing.

What is scribing?

Scribing requires that the student take detailed notes during lecture, and produce a LaTeX version of these notes. The notes should be free from typos, and should allow a reader to follow the logic of the presentation. These will be distributed to the students. A LaTeX template for scribing is available from the course website and must be used.

Presentations

Students are expected to make paper presentations, which can be tackled in groups of up to 2 students collaboratively, although two people are expected to cover the material in greater depth and sophistication. (Groups of 3 must receive my prior approval.) A list of scheduled presentations will be available on the course website. Papers other than those that have been

marked with yellow asterisks (*) require prior approval. Students are encouraged to find papers in areas of interest to them.

Presentation structure and notes

Presentations can either be slide presentations or chalk talks. Presentations should last 30 minutes. Up to fifteen minutes of questions will push this to 45 minutes in total. (Rehearse your presentations to check for timing. A rough guide is no more than one slide per minute.) In the case of a slide presentation, PDF slides will be placed on the website after the class. In the case of a chalk-talk, the presenter is responsible for producing notes for their own talk. These can be the hand-written notes if they are clearly written and suitable for publication on the website, otherwise they should be LaTeX'd. Notes should be submitted by the Friday following the presentation.

Project Report

Projects are meant to be short and low-risk, but potentially the start to something more interesting. The deliverables are:

1. One-page summaries of projects, due by email by Wednesday, November 11 (note there is no class that week due to a study break). Summaries allow me to give early advice.
2. Project reports, due December 8, by email, by midnight, in PDF format, with proper citations and bibliographies, free from grammatical and spelling errors. I take plagiarism extremely seriously. (Any code produced should be submitted alongside the PDF, e.g., in a tarball or zip file.)

Project reports should ideally be produced in LaTeX, and can use any LaTeX style, but I am looking for what would amount to ~4 pages worth of material had the paper been written in the NIPS style.

The prototypical probabilistic programming project is to find a statistical model in the literature, and implement (or approximate) it in one (or ideally two different) probabilistic programming system(s). The goal would be to reproduce a key plot/figure in a research paper that used the model on some data. A good project along these lines would carefully document the challenges that arose in using each probabilistic programming systems, including any approximations that had to be made in order to get useful results. An excellent project might then use the flexibility of the probabilistic programming ecosystem to propose a slightly improved model and get some preliminary results. Another contribution that would make an excellent project, would be to show how a family of related statistical models can be represented by a single probabilistic programming library. There are many other ideas that would make an excellent project.

Side note: Note that many probabilistic programming systems are alpha-level software and may contain bugs. Developers may not be able to respond to you quickly enough to meet deadlines. Other systems are more mature. I recommend starting with a more mature system, and, once you have enough results to write up research paper. Alternatively, start early. Note also that

some systems cannot handle large data sets because the resulting algorithm is too slow or too memory intensive. Start small, and build up.

Another prototypical research project is to perform a literature review of several related papers. A good project along these lines would identify strengths and weaknesses of the papers, and propose (actually) interesting avenues for further exploration. It's hard to summarize what an excellent project would be. I'd probably want to learn something in the course of reading the review.

For those who are theoretically inclined, I welcome your own ideas, although I would recommend that you come to me early so that we can discuss them.

Policy on Late Work

Research projects are due by email on the last day of class (11:59pm Toronto time, Dec 8). Every day of delay thereafter results in a 33% deduction. Presentations cancelled later than Friday noon will be counted as missed. Missed presentations can be made up if there are slots available for 75%. Extenuating circumstances will be handled on a case by case basis.

Course Webpage

Blackboard will be used to manage the course list and grades, but the course information and links will be available at <http://danroy.org/teaching/2015/STA4516/>

Accessibility

Students with diverse learning styles and needs are welcome in this course. Please feel free to approach me or Accessibility Services so we can assist you in achieving academic success in this course. If you have not registered with the Accessibility Services and have a disability, please visit the Accessibility Services website at <http://www.accessibility.utoronto.ca> for information on how to register.

Reading

There is no required course text (indeed, a suitable text doesn't even exist yet). Students are responsible for reading the papers being presented each week. Papers marked with * are approved for presenting. Students may suggest other articles (listed here, or otherwise), but my prior approval is required.

Online Tutorials and Books

- **Probabilistic Models of Cognition** (using WebChurch)
 - By Noah Goodman, Joshua Tenenbaum, with Timothy O'Donnell, Tomer Ullman, et al.
 - <https://probmods.org/>
- **The Design and Implementation of Probabilistic Programming Languages** (using WebPPL)
 - By Noah Goodman and Andreas Stuhlmüller
 - <http://dippl.org/>

- **Towards common-sense reasoning via conditional simulation: Legacies of Turing in Artificial Intelligence**
 - (with Cameron Freer and Josh Tenenbaum)
 - Turing's Legacy (ASL Lecture Notes in Logic), 2012.
 - <http://arxiv.org/abs/1212.4799>
- **Practical Probabilistic Programming (using Figaro)**
 - <https://www.manning.com/books/practical-probabilistic-programming>

Seminal Papers

- **The Design and Implementation of IBAL: A General-Purpose Probabilistic Language**
 - Avi Pfeffer
 - Introduction to Statistical Relational Learning, 2005
 - <http://www.gelberpfeffer.net/Papers/ibal-ch.pdf>
- **BLOG: Probabilistic Models with Unknown Objects**
 - Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong, Andrey Kolobov
 - Introduction to Statistical Relational Learning, 2005
 - <http://research.microsoft.com/en-us/um/people/akolobov/papers/blog-chapter.pdf>
- **Church: a language for generative models**
 - Noah Goodman, Vikash Mansinghka, Daniel M. Roy, Keith Bonawitz, and Joshua Tenenbaum
 - UAI 2008.
 - <http://arxiv.org/abs/1206.3255>
- **Lightweight Implementations of Probabilistic Programming Languages Via Transformational Compilation**
 - David Wingate, Andreas Stuhlmüller, and Noah D. Goodman
 - AISTATS 2011
 - <https://stuhlmueeller.org/papers/lightweight-mcmc-aistats2011.pdf>

PhD Theses

- **Brian Milch, 2006. Probabilistic Models with Unknown Objects**
 - University of California, Berkeley.
 - <http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-174.pdf>
- **Vikash K. Mansinghka, 2009. Natively probabilistic computation**
 - Massachusetts Institute of Technology
 - <http://dspace.mit.edu/handle/1721.1/47892>
 - MIT/EECS George M. Sprowls Doctoral Dissertation Award
- **Daniel M. Roy, 2011. Computability, inference and modeling in probabilistic programming**
 - Massachusetts Institute of Technology
 - <http://dspace.mit.edu/handle/1721.1/66463>
 - MIT/EECS George M. Sprowls Doctoral Dissertation Award

- **Andreas Stuhlmüller, 2015. Modeling Cognition with Probabilistic Programs: Representations and Algorithms**
 - Massachusetts Institute of Technology
 - <https://stuhlmueeller.org/papers/stuhlmueeller-phdthesis.pdf>

Algorithms

- * **Coarse-to-Fine Sequential Monte Carlo for Probabilistic Programs**
 - Andreas Stuhlmüller, Robert X.D. Hawkins, N. Siddharth, Noah D. Goodman
 - <http://arxiv.org/abs/1509.02962>
- * **C3: Lightweight Incrementalized MCMC for Probabilistic Programs using Continuations and Callsite Caching**
 - Daniel Ritchie, Andreas Stuhlmüller, Noah D. Goodman
 - <http://arxiv.org/abs/1509.02151>
- * **Output-Sensitive Adaptive Metropolis-Hastings for Probabilistic Programs**
 - Tolpin, D., van de Meent, J.-W., & Paige, F., Brooks Wood.
 - ECML PKDD 2015.
 - <http://www.robots.ox.ac.uk/~fwood/assets/pdf/Tolpin-ECMLPKDD-2015.pdf>
- * **Particle Gibbs with Ancestor Sampling for Probabilistic Programs**
 - van de Meent, J.-W., Yang, H., Mansinghka, V., & Wood, F.
 - AISTATS 2015
 - <http://www.robots.ox.ac.uk/~fwood/assets/pdf/vandeMeent-AISTATS-2015.pdf>
- * **A Compilation Target for Probabilistic Programming Languages**
 - Paige, B., & Wood, F. (2014)
 - ICML (2014)
 - <http://www.robots.ox.ac.uk/~fwood/assets/pdf/Paige-ICML-2014.pdf>
- * **Generating Efficient MCMC Kernels from Probabilistic Programs**
 - L. Yang, P. Hanrahan, & N. D. Goodman. (2014). In AISTATS
 - <https://web.stanford.edu/~ngoodman/papers/aistats2014-shred.pdf>
- * **A New Approach to Probabilistic Programming Inference**
 - Wood, F., van de Meent, J. W., & Mansinghka, V.
 - AISTATS (2014)
 - <http://www.robots.ox.ac.uk/~fwood/assets/pdf/Wood-AISTATS-2014.pdf>
- **Venture: a general purpose probabilistic programming platform with efficient stochastic inference.**
 - Mansinghka, V., Radul, A.
 - <http://arxiv.org/pdf/1404.0099v1.pdf>
- * **Learning Stochastic Inverses**
 - A. Stuhlmüller, J. Taylor, & N. Goodman.
 - (2013). In Advances in Neural Information Processing Systems
 - <https://web.stanford.edu/~ngoodman/papers/inverses-nips-2013.pdf>
- * **A dynamic programming algorithm for inference in recursive probabilistic programs**
 - Andreas Stuhlmüller and Noah D. Goodman.

- In Second Statistical Relational AI workshop at UAI 2012 (StaRAI-12), 2012.
- <http://arxiv.org/abs/1206.3555>
- * **Nonstandard Interpretations of Probabilistic Programs for Efficient Inference**
 - David Wingate, Noah D. Goodman, Andreas Stuhlmüller, and Jeffrey M. Siskind (2011)
 - Advances in Neural Information Processing Systems (NIPS) 24
 - <https://stuhlmueeller.org/papers/nonstandard-interpretations-nips2011.pdf>

Domain Specific Languages and Applications

- * **Generating Design Suggestions under Tight Constraints with Gradient-based Probabilistic Programming**
 - D. Ritchie, S. Lin, N. D. Goodman, & P. Hanrahan. (2015).
 - In Proceedings of Eurographics 2015.
 - <http://cocolab.stanford.edu/papers/RitchieEtAl2015-Eurographics.pdf>
- * **Picture: an imperative probabilistic programming language for scene perception.**
 - Kulkarni, T., Kohli, P., Tenenbaum, J., Mansinghka, V.
 - CVPR 2015 (Best Paper Honorable Mention)
 - http://mrkulk.github.io/www_cvpr15/
 - See also preceding paper at NIPS: <http://probcomp.csail.mit.edu/gppp/>
- * **Reasoning about Reasoning by Nested Conditioning: Modeling Theory of Mind with Probabilistic Programs**
 - A. Stuhlmüller, & N. D. Goodman. (2013). J. Cognitive Systems Research
 - <https://web.stanford.edu/~ngoodman/papers/StuhlmueellerGoodman-CoqSys-2013.pdf>
- **A stochastic programming perspective on nonparametric Bayes**
 - Daniel M. Roy, Vikash Mansinghka, Noah Goodman, and Joshua Tenenbaum
 - ICML Workshop on Nonparametric Bayesian, 2008.

Combining Logic and Probability in AI

- * **Semantics and Inference for Recursive Probability Models**
 - Avi Pfeffer and Daphne Koller
 - AAAI 2000
 - <http://www.gelberpfeffer.net/Papers/recursive.ps>
- * **The magic of logical inference in probabilistic programming.**
 - Gutmann et al.
 - Theory and Practice of Logic Programming 11.4-5 (2011): 663-680.
 - <http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=8320688&fileId=S1471068411000238>
- **Effective Bayesian Inference for Stochastic Programs.**
 - Daphne Koller, David McAllester, and Avi Pfeffer.
 - AAAI (1997)
 - <http://www.gelberpfeffer.net/Papers/aaai97func.ps>

- **Introduction to Statistical Relational Learning**
 - Lise Getoor and Ben Taskar
 - <http://www.cs.umd.edu/srl-book/>
 - **This book contains many excellent chapters and links to an older tradition combining logic and probability.**
- [And many many many more predating probabilistic programming in AI/ML]

Theory (Computable Analysis and Probability Theory)

- **On the topological aspects of the theory of represented spaces**
 - Arno Pauly
 - <http://arxiv.org/abs/1204.3763>
- * **Posterior distributions are computable from predictive distributions**
 - Cameron Freer and Daniel M. Roy
 - Proc. Artificial Intelligence and Statistics (AISTATS), 2010.
- * **Noncomputable conditional distributions**
 - (with Nate Ackerman and Cameron Freer)
 - Proc. Logic in Computer Science (LICS), 2011.
 - See also:
 - **On the computability of conditional probability**
 - (with Nate Ackerman and Cameron Freer)
 - arXiv:1005.3014
- * **Computable de Finetti measures**
 - (with Cameron Freer)
 - Annals of Pure and Applied Logic, 2012.
 - See also:
 - **Computable exchangeable sequences have computable de Finetti measures**
 - Cameron E. Freer and Daniel M. Roy
 - Proc. Computability in Europe (CiE), 2009.
- * **On computability and disintegration**
 - (with Nate Ackerman and Cameron Freer)
 - arXiv:1509.02992

Theory (Programming Languages)

- * Alessandra Di Pierro and Herbert Wiklicky. **Probabilistic analysis of programs: A weak limit approach.** In Ugo dal Lago, editor, 3rd International Workshop on Foundational and Practical Aspects of Resource Analysis (FOPARA 2013), volume 8552 of Lecture Notes in Computer Science. Springer, 2014.
 - See also: Alessandra Di Pierro and Herbert Wiklicky. **Semantics of probabilistic programs: A weak limit approach.** In Chung chieh Shan, editor, Programming Languages and Systems - Proceedings of the 11th Asian Symposium, APLAS 2013,

volume 8301 of Lecture Notes in Computer Science, pages 241-256. Springer International Publishing, 2013.

- * Sooraj Bhat, Johannes Borgström, Andrew D. Gordon, and Claudio Russo. **Deriving probability density functions from probabilistic functional programs.** In 19th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS). Springer, 2013. EAPLS Best Paper Award for ETAPS 2013.
- * Patrick Cousot and Michael Monerau. **Probabilistic abstract interpretation.** In Helmut Seidl, editor, Programming Languages and Systems, number 7211 in Lecture Notes in Computer Science, pages 169-193. Springer Berlin Heidelberg, January 2012.
- * Norman Ramsey and Avi Pfeffer. **Stochastic lambda calculus and monads of probability distributions.** Proc. of the 29th ACM SIGPLAN-SIGACT Symp. on Principles of Program. Lang., pages 154-165, 2002.
- * Achim Jung and Regina Tix. **The troublesome probabilistic powerdomain.** In Third Workshop on Computation and Approximation, volume 13 of Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers, 1998.
- * Claire Jones. **Probabilistic non-determinism.** PhD thesis, University of Edinburgh, Edinburgh, Scotland, UK, 1989.
- * Claire Jones and Gordon Plotkin. **A probabilistic powerdomain of evaluations.** In Proc. of the Fourth Ann. Symp. on Logic in Comp. Sci., pages 186-195. IEEE Press, 1989.
- * Gordon Plotkin. **Probabilistic powerdomains.** In Proceedings CAAP, 1982.
- * Dexter Kozen. **Semantics of probabilistic programs.** Journal of Computer System Sciences, 22:328-350, 1981.
- * Nasser Saheb-Djahromi. **Probabilistic LCF.** In Mathematical Foundations of Comput. Sci., 1978 (Proc. Seventh Sympos., Zakopane, 1978), volume 64 of Lecture Notes in Comput. Sci., pages 442-451. Springer, Berlin, 1978.

Theory (Probability)

- **On the topological aspects of the theory of represented spaces**
 - Arno Pauly
 - <http://arxiv.org/abs/1204.3763>
- **Joint Measurability and the One-way Fubini Property for a Continuum of Independent Random Variables**
 - P J Hammond and Y Sun
 - <http://web.stanford.edu/~hammond/fubini.pdf>
- **Stochastic Processes Depending on a Continuous Parameter**
 - J L Doob
 - <http://www.ams.org/journals/tran/1937-042-01/S0002-9947-1937-1501916-1/S0002-9947-1937-1501916-1.pdf>

Additional reading

A larger (but still far from complete) collection of research articles on probabilistic languages and probabilistic programming can be found here:

<http://probabilistic-programming.org/research/>

Relevant work in statistics, probability, and other fields

- General state space Markov chains and MCMC algorithms
 - Gareth Roberts and Jeff Rosenthal
 - <http://arxiv.org/pdf/math/0404033.pdf>

Selection of Probabilistic Programming Languages and Systems

- **WebPPL** (descendant of Church)
 - <http://webppl.org/>
 - See also “Play space” editor for WebChurch: <https://probmods.org/play-space.html>
- **Venture** (descendant of Church)
 - <http://probcomp.csail.mit.edu/venture/>
- **Stan**
 - <http://mc-stan.org/>
- **Figaro**
 - <https://www.cra.com/work/case-studies/figaro>
 - <https://github.com/p2t2/figaro>
- **Anglican** (descendant of Church)
 - <http://www.robots.ox.ac.uk/~fwood/anglican/>
- **BLOG**
 - <http://bayesianlogic.github.io/>
- **Infer.NET**
 - <http://research.microsoft.com/en-us/um/cambridge/projects/infernet/>

Many more can be found listed at <http://probabilistic-programming.org>