Probabilistic programming

Probabilistic programming (**PP**) is a programming paradigm in which probabilistic models are specified and inference for these models is performed automatically.[1] It represents an attempt to unify probabilistic modeling and traditional general purpose programming in order to make the former easier and more widely applicable.[2][3] It can be used to create systems that help make decisions in the face of uncertainty.

Programming languages used for probabilistic programming are referred to as "probabilistic programming languages" (PPLs).

Contents

Applications

Probabilistic programming languages

- Relational
- List of probabilistic programming languages

Difficulty

See also

Notes

External links

Applications

Probabilistic reasoning has been used for a wide variety of tasks such as predicting stock prices, recommending movies, diagnosing computers, detecting cyber intrusions and image detection.[4] However, until recently (partially due to limited computing power), probabilistic programming was limited in scope, and most inference algorithms had to be written manually for each task.

Nevertheless, in 2015, a 50-line probabilistic computer vision program was used to generate 3D models of human faces based on 2D images of those faces. The program used inverse graphics as the basis of its inference method, and was built using the Picture package in Julia.[4] This made possible "in 50 lines of code what used to take thousands".[5][6]

The Gen probabilistic programming library (also written in Julia) has been applied to vision and robotics tasks.[7]

More recently, the probabilistic programming systems Turing.jl has been applied in various pharmaceutical and economics applications.[8]

Probabilistic programming in Julia has also been combined with differentiable programming by combining the Julia package Zygote.jl with Turing.jl. [9]

Probabilistic programming languages

PPLs often extend from a basic language. The choice of underlying basic language depends on the similarity of the model to the basic language's ontology, as well as commercial considerations and personal preference. For instance, Dimple[10] and Chimple[11] are based on Java, Infer.NET is based on .NET Framework,[12] while PRISM extends from Prolog.[13] However, some PPLs such as WinBUGS and Stan offer a self-contained language, with no obvious origin in another language.[14][15]

Several PPLs are in active development, including some in beta test. The two most popular tools are Stan and PyMC3.[16]

Relational

A **probabilistic relational programming language** (PRPL) is a PPL specially designed to describe and infer with probabilistic relational models (PRMs).

A PRM is usually developed with a set of algorithms for reducing, inference about and discovery of concerned distributions, which are embedded into the corresponding PRPL.

List of probabilistic programming languages

Name	Extends from	Host language
Analytica[17]		C++
bayesloop[18][19]	Python	Python
CuPPL[20]	NOVA[21]	
Venture[22]	Scheme	C++
Probabilistic-C[23]	C	C
Anglican[24]	Clojure	Clojure
IBAL[25]	OCaml	
BayesDB[26]	SQLite, Python	
PRISM[13]	B-Prolog	
Infer.NET[12]	.NET Framework	.NET Framework
dimple[10]	MATLAB, Java	
chimple[11]	MATLAB, Java	
BLOG[27]	Java	
delSAT[28]	Answer set programming, SAT (DIMACS CNF)	
PSQL[29]	SQL	
BUGS[14]		
FACTORIE[30]	Scala	Scala
PMTK[31]	MATLAB	MATLAB
Alchemy[32]	C++	
Dyna[33]	Prolog	
Figaro[34]	Scala	Scala
Church[35]	Scheme	Various: JavaScript, Scheme
ProbLog[36]	Prolog	Python, Jython
ProBT[37]	C++, Python	
Stan[15]		C++
Hakaru[38]	Haskell	Haskell
BAli-Phy (software)[39]	Haskell	C++
ProbCog[40]		Java, Python
Gamble[41]		Racket
PWhile[42]	While	Python
Tuffy[43]		Java
PyMC3[44]	Python, Theano	Python
PyMC4[45]	Python, TensorFlow Probability	Python
Rainier[46][47]	Scala	Scala
greta[48]	TensorFlow	R

pomegranate[49]	Python	Python
Lea[50]	Python	Python
WebPPL[51]	JavaScript	JavaScript
Let's Chance[52]	Scratch	JavaScript
Picture[4]	Julia	Julia
Turing.jl[53]	Julia	Julia
Gen[54]	Julia	Julia
Low-level First-order PPL[55]	Python, Clojure, Pytorch	Various: Python, Clojure
Trolll[56]		Moscow ML
Edward[57]	TensorFlow	Python
TensorFlow Probability[58]	TensorFlow	Python
Edward2[59]	TensorFlow Probability	Python
Pyro[60]	PyTorch	Python
Saul[61]	Scala	Scala
RankPL[62]		Java
Birch[63]		C++
PSI[64]		D

Difficulty

Reasoning about variables as probability distributions causes difficulties for novice programmers, but these difficulties can be addressed through use of Bayesian network visualisations and graphs of variable distributions embedded within the source code editor.[65]

See also

- Statistical relational learning
- Inductive programming
- Bayesian programming

Notes

1. "Probabilistic programming does in 50 lines of code what used to take thousands" (http://phys.org/news/2015-04-probabilistic-lines-code-thousands.html). *phys.org*. April 13, 2015. Retrieved April 13, 2015.
2. "Probabilistic Programming" (https://web.archive.org/web/20160110035042/http://probabilistic-programming.org/wiki/Home). *probabilistic-programming.org*. Archived from the original (http://probabilistic-programming.org/wiki/Home) on January 10, 2016. Retrieved December 24, 2013.
3. Pfeffer, Avrom (2014), *Practical Probabilistic Programming*, Manning Publications. p.28. ISBN 978-1 6172-9233-0

4. "Short probabilistic programming machine-learning code replaces complex programs for computer-vision tasks" (http://www.kurzweilai.net/short-probabilistic-programming-machine-learning-code-replaces-complex-programs-for-computer-vision-tasks). *KurzweilAI*. April 13, 2015. Retrieved November 27, 2017.
5. Hardesty, Larry (April 13, 2015). "Graphics in reverse" (https://news.mit.edu/2015/better-probabilistic-programming-0413).
6. "MIT shows off machine-learning script to make CREEPY HEADS" (https://www.theregister.co.uk/2015/04/14/mit_shows_off_machinelearning_script_to_make_creepy_heads/).
7. "MIT's Gen programming system flattens the learning curve for AI projects" (https://venturebeat.com/2019/06/27/mits-gen-programming-system-allows-users-to-easily-create-computer-vision-statistical-ai-and-robotics-programs/). *VentureBeat*. June 27, 2019. Retrieved June 27, 2019.
8. *Predicting Drug-Induced Liver Injury with Bayesian Machine Learning* (https://pubs.acs.org/doi/10.1021/acs.chemrestox.9b00264), 2019
9. *∂P: A Differentiable Programming System to Bridge Machine Learning and Scientific Computing*, 2019, arXiv:1907.07587 (https://arxiv.org/abs/1907.07587)
10. "Dimple Home Page" (https://github.com/analog-garage/dimple). *analog.com*.
11. "Chimple Home Page" (https://github.com/analog-garage/chimple). *analog.com*.
12. "Infer.NET" (http://research.microsoft.com/en-us/um/cambridge/projects/infernet/). *microsoft.com*. Microsoft.
13. "PRISM: PRogramming In Statistical Modeling" (https://web.archive.org/web/20150301155729/http://rjida.meijo-u.ac.jp/prism/). *rjida.meijo-u.ac.jp*. Archived from the original (http://rjida.meijo-u.ac.jp/prism/) on March 1, 2015. Retrieved July 8, 2015.
14. "The BUGS Project - MRC Biostatistics Unit" (https://web.archive.org/web/20140314080841/http://www.mrc-bsu.cam.ac.uk/bugs/). *cam.ac.uk*. Archived from the original (http://www.mrc-bsu.cam.ac.uk/bugs/) on March 14, 2014. Retrieved January 12, 2011.
15. "Stan" (https://web.archive.org/web/20120903133321/http://mc-stan.org/). *mc-stan.org*. Archived from the original (http://mc-stan.org/) on September 3, 2012.
16. "The Algorithms Behind Probabilistic Programming" (http://blog.fastforwardlabs.com/2017/01/30/the-algorithms-behind-probabilistic-programming.html). Retrieved March 10, 2017.
17. "Analytica-- A Probabilistic Modeling Language" (http://www.analytica.com). *lumina.com*.
18. "bayesloop: Probabilistic programming framework that facilitates objective model selection for time-varying parameter models" (http://bayesloop.com/).
19. "GitHub -- bayesloop" (https://github.com/christophmark/bayesloop).
20. "Probabilistic Programming with CuPPL" (https://popl19.sigplan.org/event/lafi-2019-probabilistic-programming-with-cuppl). *popl19.sigplan.org*.
21. "NOVA: A Functional Language for Data Parallelism" (https://dl.acm.org/citation.cfm?id=2627375). *acm.org*.
22. "Venture -- a general-purpose probabilistic programming platform" (https://web.archive.org/web/20160125130827/http://probcomp.csail.mit.edu/venture/). *mit.edu*. Archived from the original (http://probcomp.csail.mit.edu/venture/) on January 25, 2016. Retrieved September 20, 2014.
23. "Probabilistic C" (https://web.archive.org/web/20160104201746/http://www.robots.ox.ac.uk/~brooks/probabilistic-c/). *ox.ac.uk*. Archived from the original (http://www.robots.ox.ac.uk/~brooks/probabilistic-c/) on January 4, 2016. Retrieved March 24, 2015.
24. "The Anglican Probabilistic Programming System" (https://github.com/probprog/anglican-infcomp). *ox.ac.uk*.
25. "IBAL Home Page" (https://web.archive.org/web/20101226131239/http://www.eecs.harvard.edu/~avi/IBAL/). Archived from the original (http://www.eecs.harvard.edu/~avi/IBAL/) on December 26, 2010.

26. "BayesDB on SQLite. A Bayesian database table for querying the probable implications of data as easily as SQL databases query the data itself" (https://github.com/probcomp/bayeslite). *GitHub*.
27. "Bayesian Logic (BLOG)" (https://web.archive.org/web/20110616214423/http://people.csail.mit.edu/milch/blog/). *mit.edu*. Archived from the original (http://people.csail.mit.edu/milch/blog/) on June 16, 2011.
28. "delSAT (probabilistic SAT/ASP)" (https://github.com/MatthiasNickles/delSAT/).
29. Dey, Debabrata; Sarkar, Sumit (1998). "PSQL: A query language for probabilistic relational data". *Data & Knowledge Engineering*. **28**: 107–120. doi:10.1016/S0169-023X(98)00015-9 (https://doi.org/10.1016%2FS0169-023X%2898%2900015-9).
30. "Factorie - Probabilistic programming with imperatively-defined factor graphs - Google Project Hosting" (https://code.google.com/p/factorie/). *google.com*.
31. "PMTK3 - probabilistic modeling toolkit for Matlab/Octave, version 3 - Google Project Hosting" (https://code.google.com/p/pmtk3/). *google.com*.
32. "Alchemy - Open Source AI" (http://alchemy.cs.washington.edu/). *washington.edu*.
33. "Dyna" (https://web.archive.org/web/20160117155947/http://dyna.org/). *www.dyna.org*. Archived from the original (http://www.dyna.org/) on January 17, 2016. Retrieved January 12, 2011.
34. "Charles River Analytics - Probabilistic Modeling Services" (http://www.cra.com/figaro). *cra.com*.
35. "Church" (https://web.archive.org/web/20160114182510/http://projects.csail.mit.edu/church/wiki/Church). *mit.edu*. Archived from the original (http://projects.csail.mit.edu/church/wiki/Church) on January 14, 2016. Retrieved April 8, 2013.
36. "ProbLog: Probabilistic Programming" (http://dtai.cs.kuleuven.be/problog). *dtai.cs.kuleuven.be*.
37. ProbaYes. "ProbaYes - Ensemble, nous valorisations vos données" (https://web.archive.org/web/20160305000751/http://www.probayes.com/fr/Bayesian-Programming-Book/downloads/). *probayes.com*. Archived from the original (http://www.probayes.com/fr/Bayesian-Programming-Book/downloads/) on March 5, 2016. Retrieved November 26, 2013.
38. "Hakaru Home Page" (https://hakaru-dev.github.io/). *hakaru-dev.github.io/*.
39. "BAli-Phy Home Page" (http://www.bali-phy.org/). *bali-phy.org*.
40. "ProbCog" (https://github.com/opcode81/ProbCog/wiki/Features). *GitHub*.
41. Culpepper, Ryan (January 17, 2017). "gamble: Probabilistic Programming" (https://github.com/rmculpepper/gamble) – via GitHub.
42. "PWhile Compiler" (https://github.com/zz5013/pwCompiler). *GitHub*.
43. "Tuffy: A Scalable Markov Logic Inference Engine" (http://i.stanford.edu/hazy/tuffy/home). *stanford.edu*.
44. PyMC devs. "PyMC3" (https://pymc-devs.github.io/pymc3/). *pymc-devs.github.io*.
45. Developers, PyMC (May 17, 2018). "Theano, TensorFlow and the Future of PyMC" (https://medium.com/@pymc_devs/theano-tensorflow-and-the-future-of-pymc-6c9987bb19d5). *PyMC Developers*. Retrieved January 25, 2019.
46. *stripe/rainier* (https://github.com/stripe/rainier), Stripe, August 19, 2020, retrieved August 26, 2020
47. "Rainier · Bayesian inference for Scala" (https://samplerainier.com/). *samplerainier.com*. Retrieved August 26, 2020.
48. "greta: simple and scalable statistical modelling in R" (https://greta-dev.github.io/greta/). *GitHub*. Retrieved October 2, 2018.
49. "Home — pomegranate 0.10.0 documentation" (https://pomegranate.readthedocs.io/en/latest/index.html). *pomegranate.readthedocs.io*. Retrieved October 2, 2018.
50. "Lea Home Page" (https://bitbucket.org/piedenis/lea). *bitbucket.org*.

51. "WebPPL Home Page" (http://dippl.org/). *github.com/probmods/webppl*.
52. "Let's Chance: Playful Probabilistic Programming for Children | Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems" (https://dl.acm.org/doi/abs/10.1145/3334480.3383071). *dl.acm.org*. doi:10.1145/3334480.3383071 (https://doi.org/10.1145%2F3334480.3383071). Retrieved August 1, 2020.
53. "The Turing language for probabilistic programming" (https://github.com/yebai/Turing.jl).
54. "Gen: A General Purpose Probabilistic Programming Language with Programmable Inference" (https://probcomp.github.io/Gen/). Retrieved June 17, 2019.
55. "LF-PPL: A Low-Level First Order Probabilistic Programming Language for Non-Differentiable Models" (https://github.com/bradleygramhansen/PyLFPPL). *ox.ac.uk*.
56. "Troll dice roller and probability calculator" (http://topps.diku.dk/torbenm/troll.msp).
57. "Edward – Home" (http://edwardlib.org/). *edwardlib.org*. Retrieved January 17, 2017.
58. TensorFlow (April 11, 2018). "Introducing TensorFlow Probability" (https://medium.com/tensorflow/introducing-tensorflow-probability-dca4c304e245). *TensorFlow*. Retrieved October 2, 2018.
59. " 'Edward2' TensorFlow Probability module" (https://github.com/tensorflow/probability/tree/master/tensorflow_probability/python/edward2). *GitHub*. Retrieved October 2, 2018.
60. "Pyro" (http://pyro.ai). *pyro.ai*. Retrieved February 9, 2018.
61. "CogComp - Home" (https://cogcomp.org/page/software_view/Saul).
62. Rienstra, Tjitze (January 18, 2018), *RankPL: A qualitative probabilistic programming language based on ranking theory* (https://github.com/tjitze/RankPL), retrieved January 18, 2018
63. "Probabilistic Programming in Birch" (http://birch-lang.org/). *birch-lang.org*. Retrieved April 20, 2018.
64. "PSI Solver - Exact inference for probabilistic programs" (https://psisolver.org/). *psisolver.org*. Retrieved August 18, 2019.
65. Gorinova, Maria I.; Sarkar, Advait; Blackwell, Alan F.; Syme, Don (January 1, 2016). *A Live, Multiple-Representation Probabilistic Programming Environment for Novices*. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. New York, NY, USA: ACM. pp. 2533–2537. doi:10.1145/2858036.2858221 (https://doi.org/10.1145%2F2858036.2858221). ISBN 9781450333627.

External links

- List of Probabilistic Model Mini Language Toolkits (http://anyall.org/blog/2009/12/list-of-probabilistic-model-mini-language-toolkits/)
- Probabilistic programming wiki (https://web.archive.org/web/20160110035042/http://probabilistic-programming.org/wiki/Home)
