Probabilistic programming (PP) is a programming paradigm in which probabilistic models are specified and inference for these models is performed automatically. It represents an attempt to unify probabilistic modeling and traditional general purpose programming in order to make the former easier and more widely applicable. 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).

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. 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.

Neverthless, 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. This made possible "in 50 lines of code what used to take thousands".

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

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

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

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


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/).


8. Predicting Drug-Induced Liver Injury with Bayesian Machine Learning (https://pubs.acs.org/doi/10.1021/acs.chemrestox.9b00264), 2019


18. “bayesloop: Probabilistic programming framework that facilitates objective model selection for time-varying parameter models” (http://bayesloop.com/).


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.


List of Probabilistic Model Mini Language Toolkits


[53. "The Turing language for probabilistic programming" (https://github.com/yebai/Turing.jl).]


[56. "Troll dice roller and probability calculator" (http://topps.diku.dk/torbenm/troll.msp).]


