Nordic probabilistic AI school

Introduction to probabilistic programming languages (PPLs)

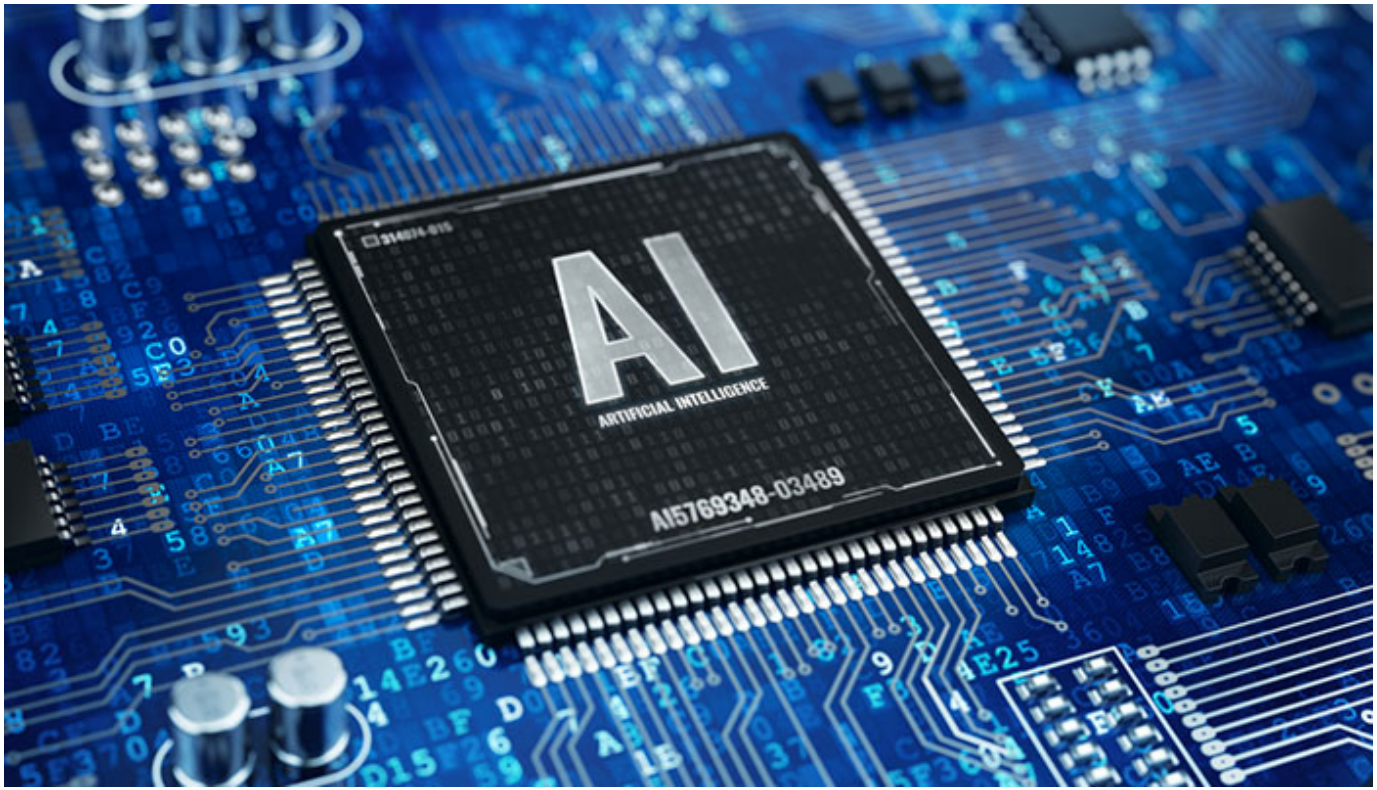
Andrés Masegosa and Thomas Dyhre Nielsen

June 12, 2023

Plan

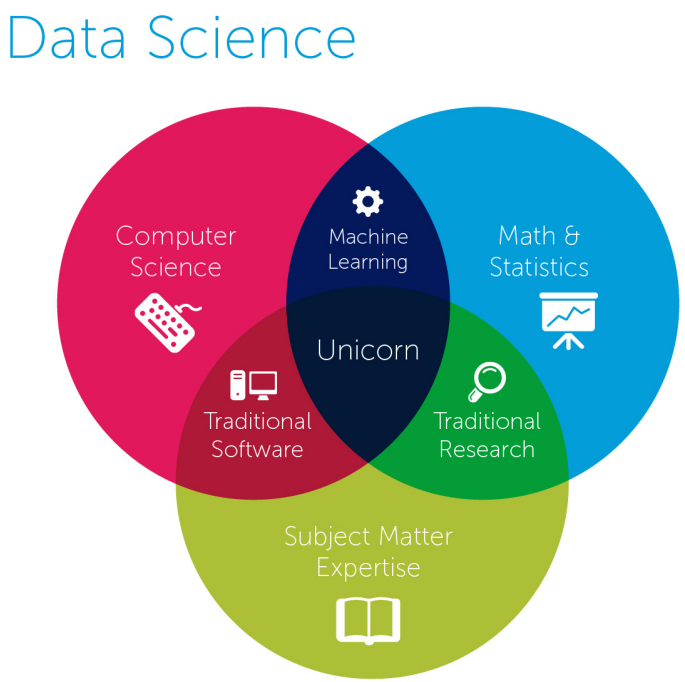
- **Day 1: Probabilistic programming**
  - Introduction to probabilistic programming
  - Probabilistic programming in Pyro
- **Day 2 (Before Lunch): Classical Variational Inference**
  - Introduction to Variational Inference
  - Mean-Field Approximation
  - Coordinate-ascent variational inference
- **Day 2 (After Lunch): Modern Variational Inference**
  - Black box variational inference
  - Variational inference in Pyro
- **Day 2 (Evening): Modern Variational Inference**
  - Variational AutoEncoders
  - Amortized Inference

Machine Learning Systems



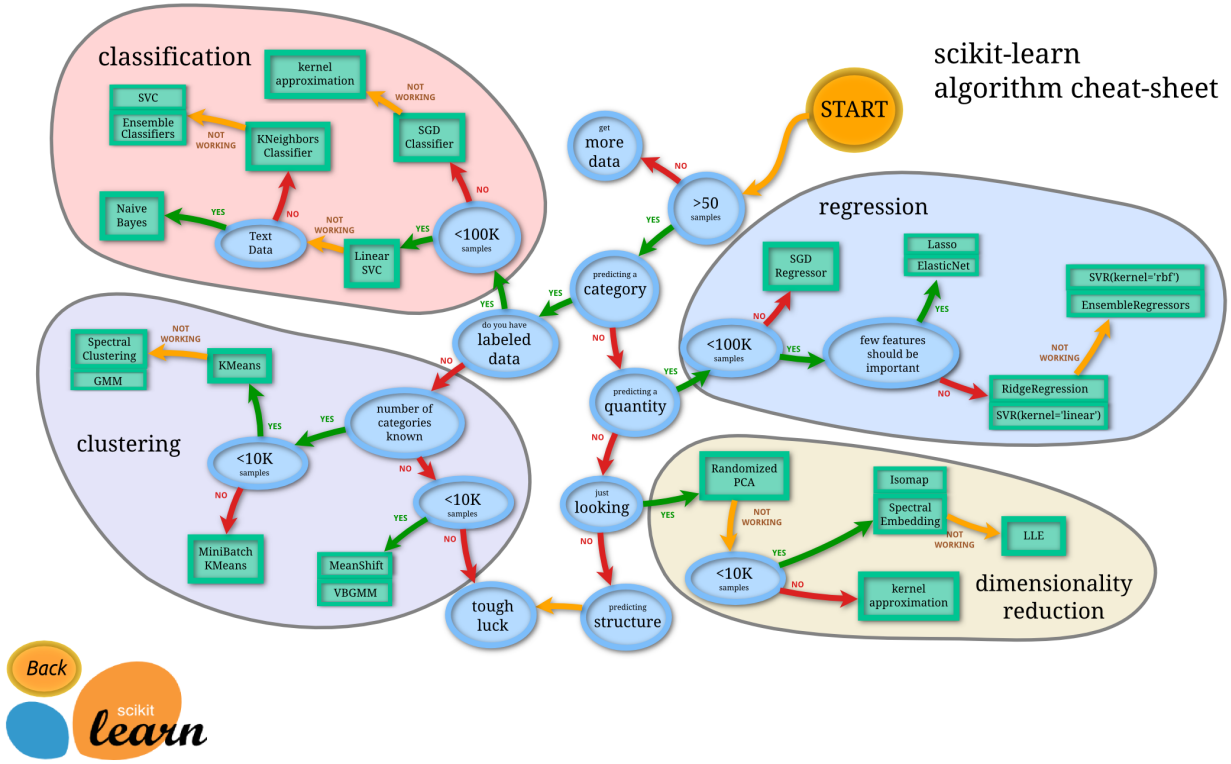
The development of **machine learning systems** requires enormous efforts.

- It can be a **highly technical** task.



The development of **machine learning systems** requires enormous efforts.

- It can be a **highly technical** task.
- It requires of **highly qualified experts.**

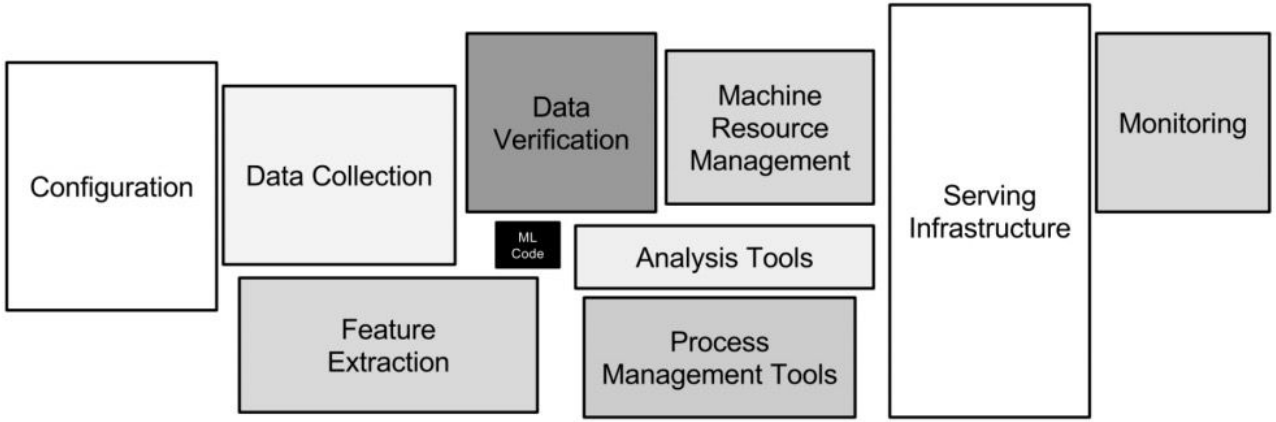


The development of **machine learning systems** requires enormous efforts.

- It can be a **highly technical** task.
- It requires of **highly qualified experts.**
- It is difficult to find the **ML model most suitable** for an application.

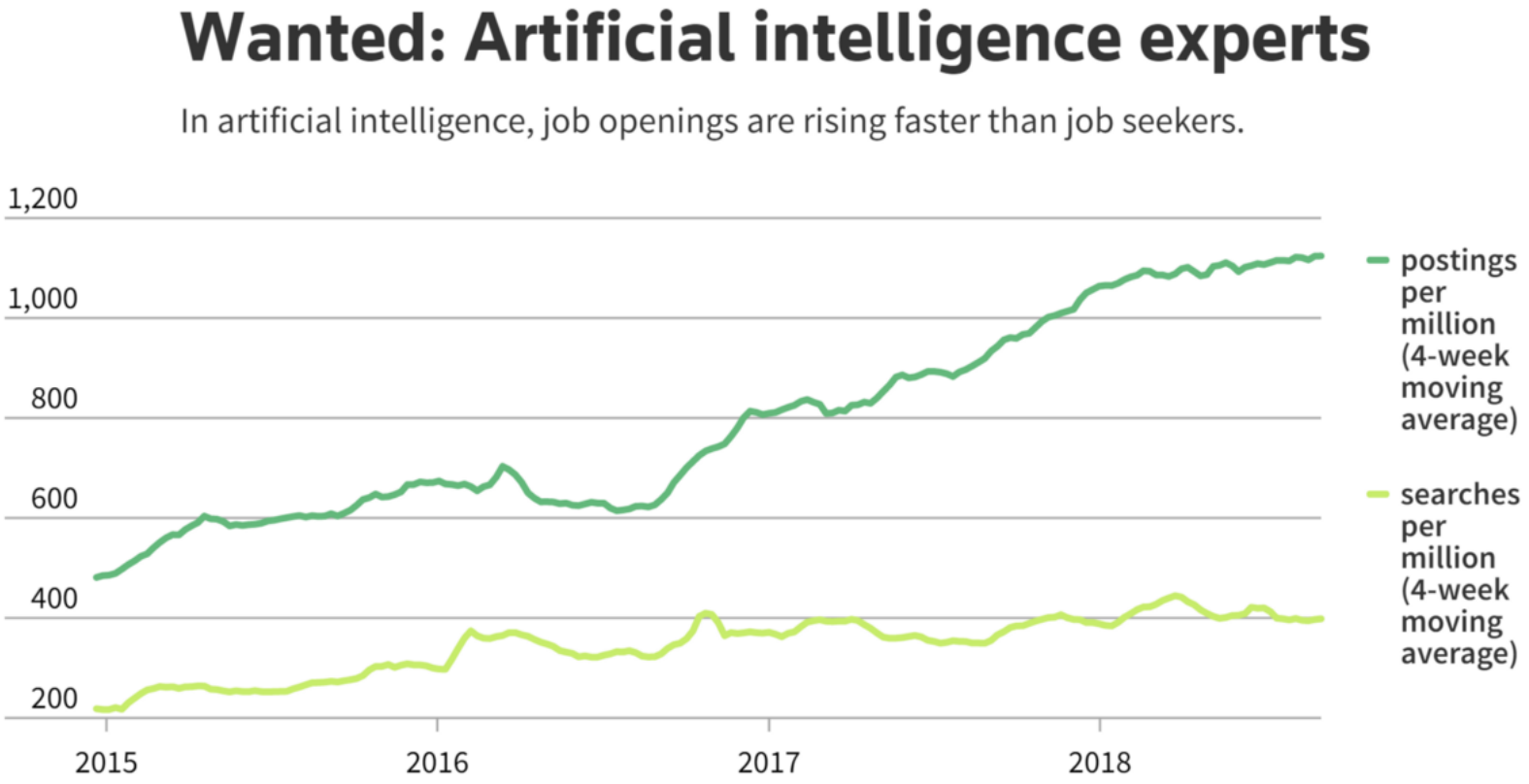
Hidden Technical Debt in Machine Learning Systems

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.



The development of **machine learning systems** requires enormous efforts.

- It can be a **highly technical** task.
- It requires of **highly qualified experts.**
- It is difficult to find the **ML model most suitable** for an application.
- Programming a ML model is a **complex task**; many problems are intermingled.

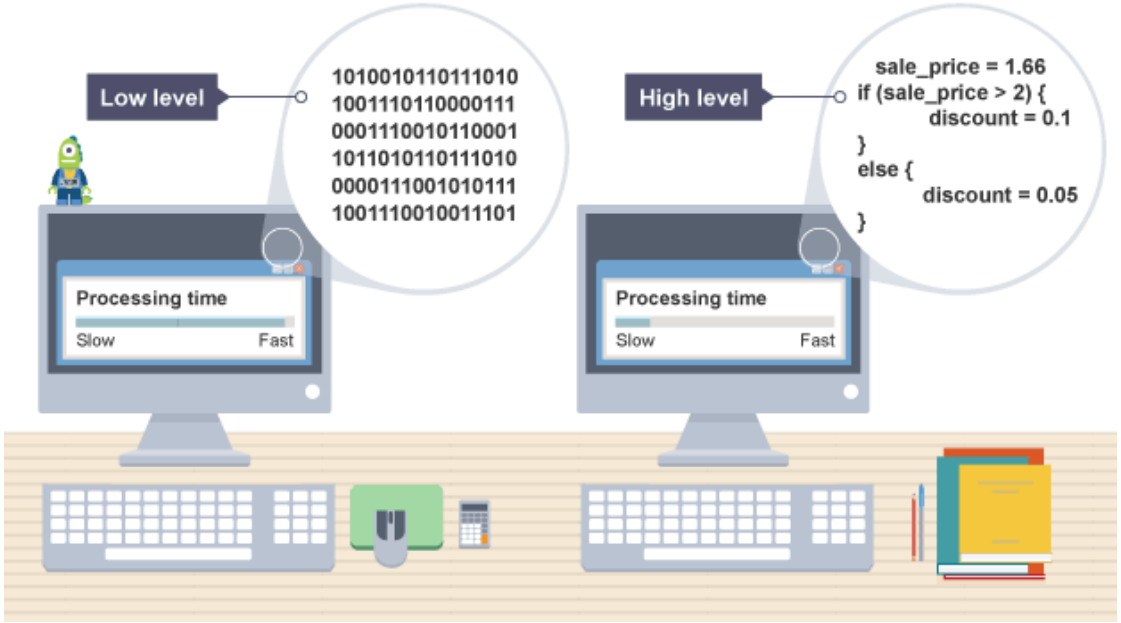


https://www.ml.cmu.edu/[...]good-news-for-job-seekers-with-ai-skills-there-is-a-shortage-of-qualified-workers.html

Consequences:

- Shortage of AI experts (and high salaries).
- Only big corporations have the resources for developing ML systems.

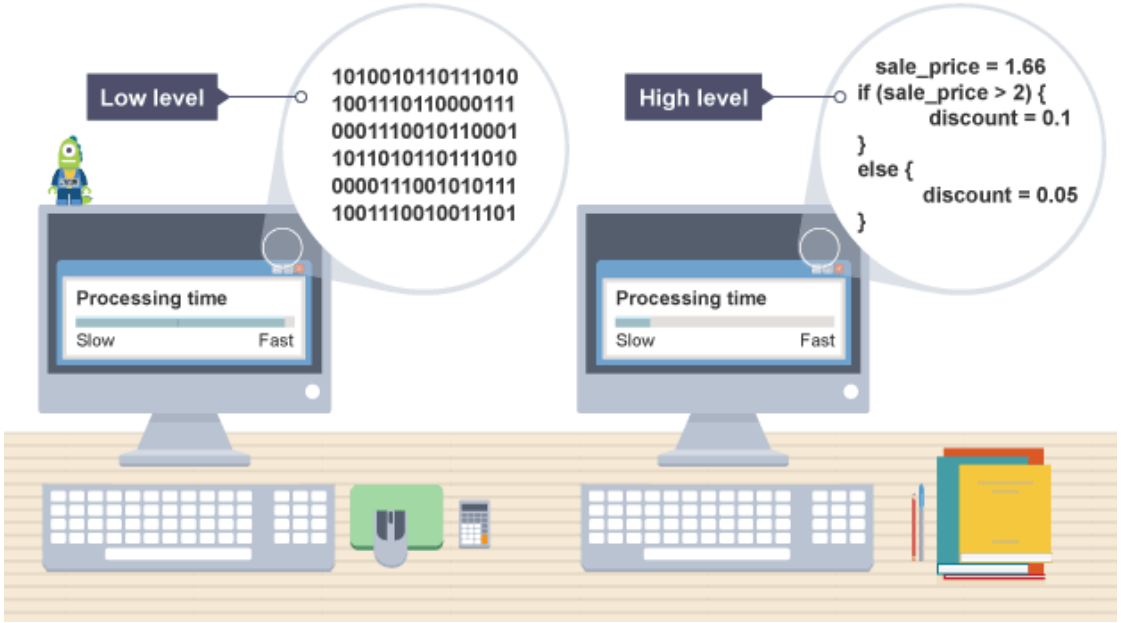
Programming Languages



Similar situation than 50 years ago:

- People used to program in low-level programming languages.
- Programming was complex and demand high-expertise.
- Focus on application and low-level hardware details.

Programming Languages



High-level programming languages brought many advantages:

- Programmers focused on the applications.
- Hardware Experts focused on compilers.
- High gains in productivity.
- “Democratization” of the software development.

Big Data and Machine Learning Ecosystem

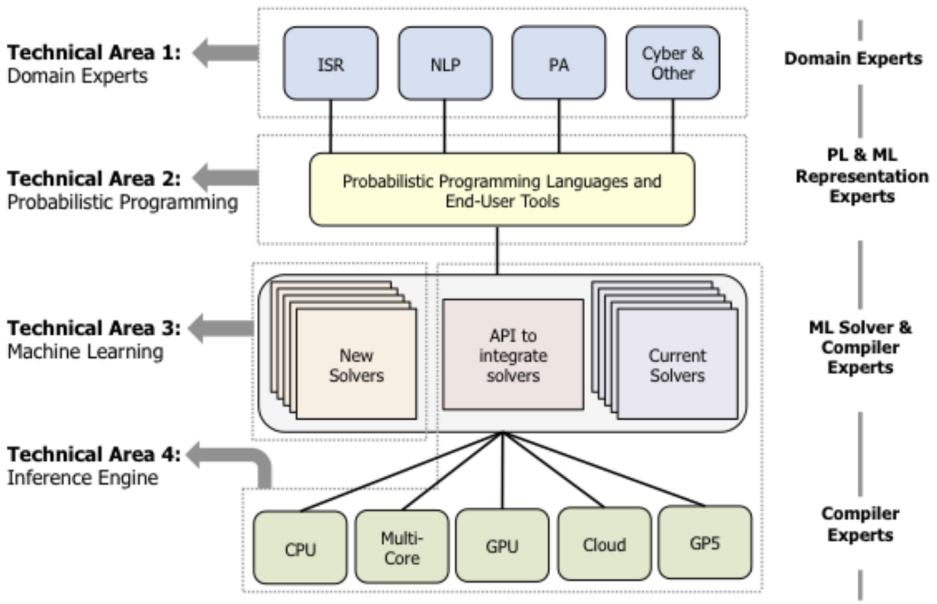
Claire D. Costa. Best Python Libraries for Machine Learning and Deep Learning.
https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c

Big Data and Machine Learning Libraries:

- **High-quality**, well-maintained, and open-source libraries
- Provide **high-level abstractions**.
- Hide **low level details** under the hood.
- Increase the **adoption** of the technologies.

What are the "high-level libraries" in Probabilistic AI?

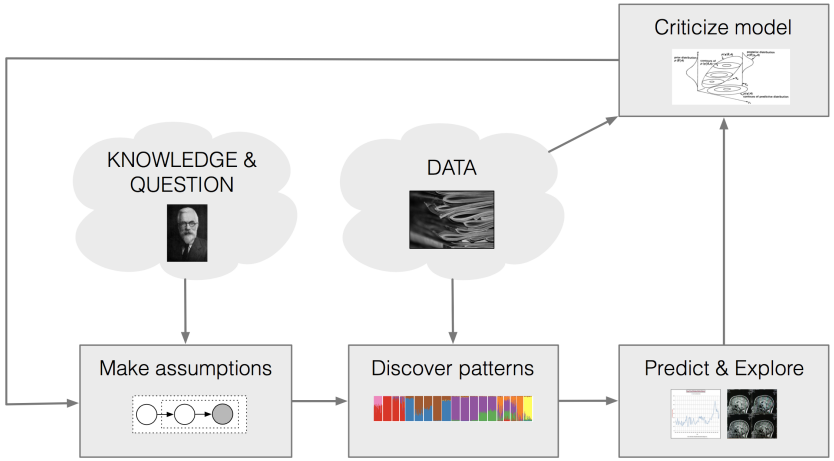
Why probabilistic programming languages (PPLs)?



PPLs as high-level programming languages for **probabilistic ML systems**:

- Stacked architecture.
- Different Domain Experts can code their models using the same language.
- ML experts can focus on the development of ML methods/algorithms (ML solvers).
- Compiler experts can focus on running these ML solvers on specialized hardware.

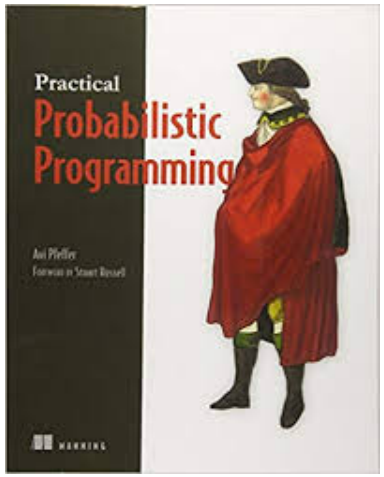
Box's Loop



[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

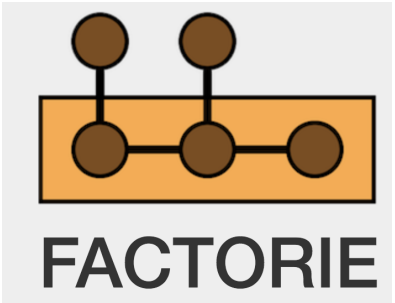
Benefits of PPLs for developing probabilistic machine learning systems:

- Simplify probabilistic machine learning model code.
- Reduce development time and cost to encourage experimentation.
- Reduce the necessary level of expertise.
- "Democratization" of the development of probabilistic ML systems.



1st Generation of PPLs :

- Bugs, WinBugs, Jags, Figaro, etc.
- Turing-complete probabilistic programming languages. (i.e. they can represent any computable probability distribution).
- Inference engine based on Monte Carlo methods.
- Not able to scale to large data samples/high-dimensional models.

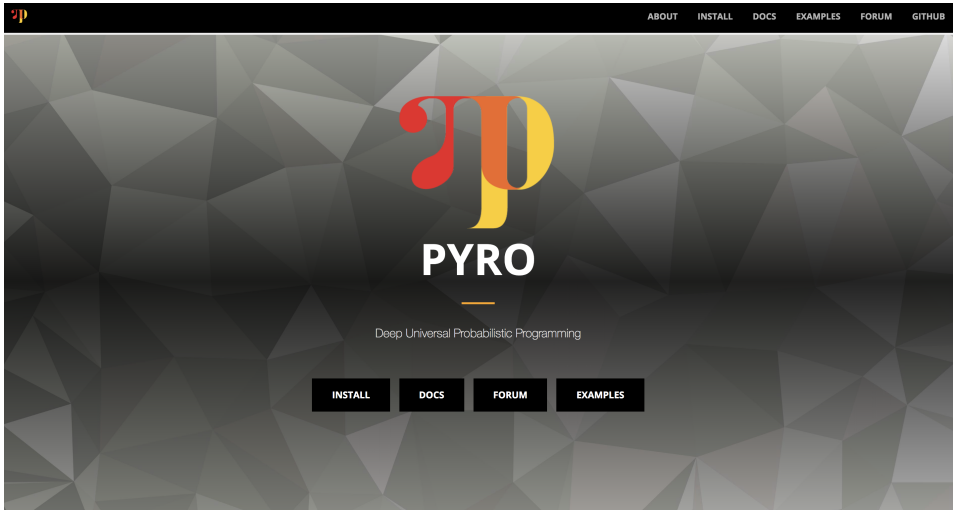


2nd Generation of PPLs :

- Infer.net, Factorie, Amidst, etc.
- Inference engine based on message passage algorithms and/or variational inference methods.
- Scale to large data samples/high-dimensional models.
- Restricted probabilistic model families (i.e. factor graphs, conjuage exponential family, etc.)

3rd Generation of PPLs :

- Pyro, Stan, PyMC, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- Scale to large data samples/high-dimensional models.
- Turing-complete probabilistic programming languages.
  - Strong focus on probabilistic models with **deep neural networks**.
- Most rely on deep learning frameworks (Pytorch, JAX, TensorFlow, etc).
  - Specialized hardware like GPUs, TPUs, etc.
  - Automatic differentiation methods.



Pyro's main features (`www.pyro.ai`) :

- Initially developed by UBER (the car riding company).
- Community of contributors and a dedicated team at Broad Institute (US).
- Rely on Pytorch (Deep Learning Framework).
- Enable GPU accelaration and distributed learning.

`https://github.com/PGM-Lab/2023-ProbAI`