Exploring Visual Programming Concepts for Probabilistic Programming Languages

Gabriel Candal

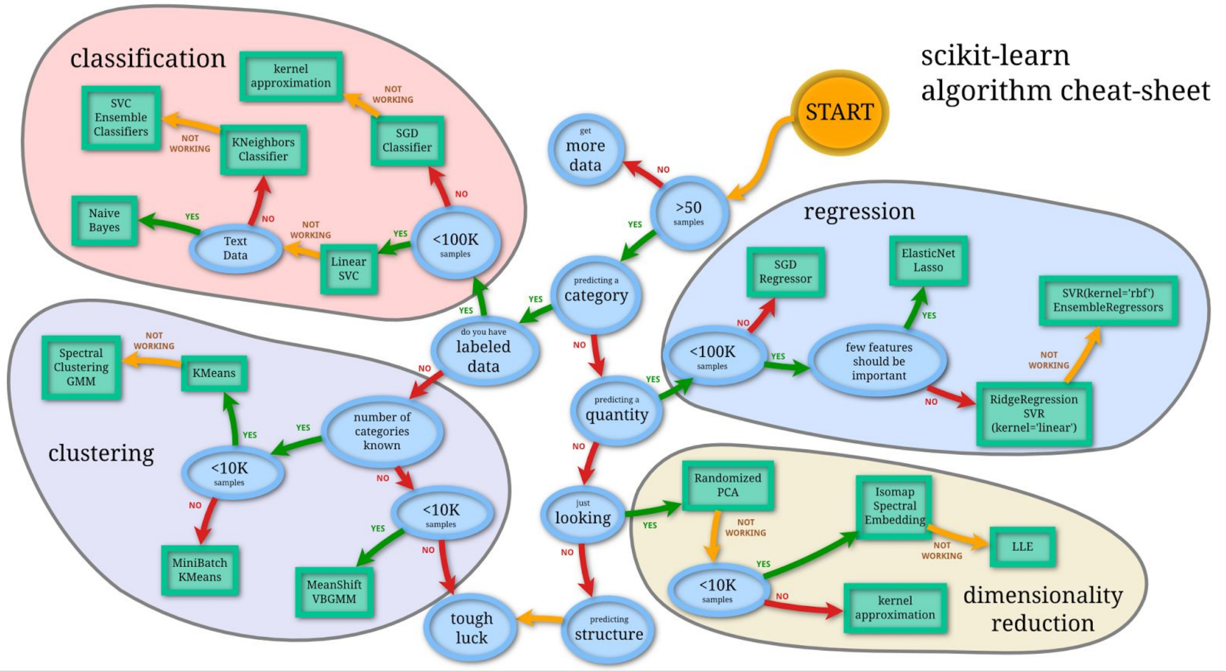
Reasoning under uncertainty

Required in some domains:

- Computer Vision
- Cryptography
- Biology
- Fraud detection
- Internet ads bid/placement
- Recommender systems

Reasoning under uncertainty

Traditional approach: use prepackaged and very generic machine learning models



Traditional approach

Problems?

- Doesn’t fully leverage domain knowledge
- Must fit your data into the model (feature extraction+normalization+transform.)
- May be hard to explain the results

Solution?

- Build your own model!

Building your own model

Usually based on Bayesian reasoning:

p(cause | observable) = p(observable | cause) * p(cause) / p(observable)

- **Model unknown causes with random variables**
- Specify how unknown causes relate to observable variables
- **Feed data** into observable variables
- Invert the story: **infer**!

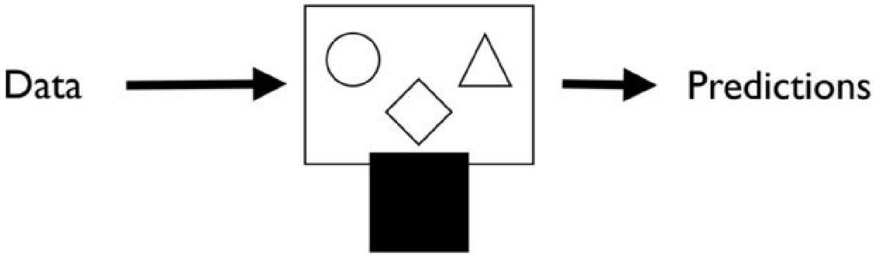
Building your own model

Problem?

- Must write inference by yourself: non-trivial, demanding and error-prone

Solution?

- Openbox models
- Blackbox inference engine



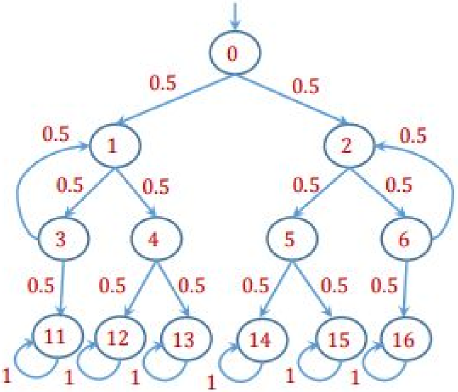
From Olivier Grisel's SciPy 2013 Keynote

Probabilistic Programming Languages (PPLs)

Why PPLs

- **Unambiguous** way to communicate a model
- **Re-use** inference engine
- Use **different inference** methods easily
- **More general** than graphical models (such as Bayesian and Markov Networks)

```
int x = 0;
while (x < 11) {
  bool coin = Bernoulli(0.5);
  if(x=0)
    if (coin) x = 1 else x = 2;
  else if (x=1)
    if (coin) x = 3 else x = 4;
  else if (x=2)
    if (coin) x = 5 else x=  6;
  else if (x=3)
    if (coin) x = 1 else x = 11;
  else if (x=4)
    if (coin) x = 12 else x = 13;
  else if (x=5)
    if (coin) x = 14 else x = 15;
  else if (x=6)
    if (coin) x = 16 else x = 2;
}
return (x);
```

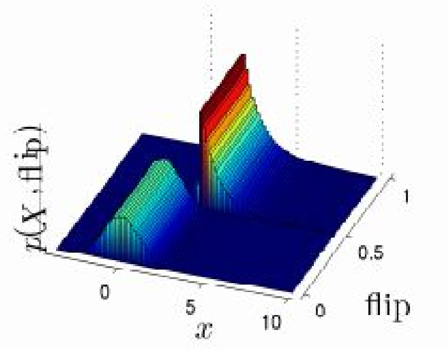


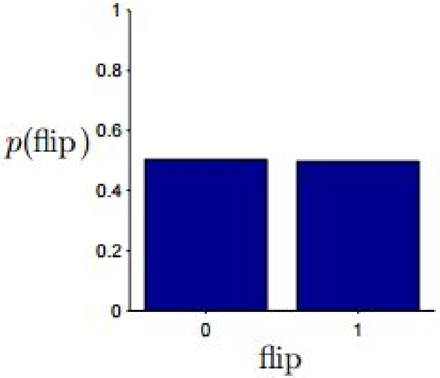
From: Andrew D. Gordon and Thomas A. Henzinger and Aditya V. Nori and Sriram K. Rajamani. Probabilistic Programming. International Conference on Software Engineering (ICSE Future of Software Engineering), 2014.

Why PPLs - example

```
flip = rand < 0.5
if flip
    x = randg + 2    % Random draw from Gamma(1,1)
else
    x = randn        % Random draw from standard Normal
end
```

Implied distributions over variables

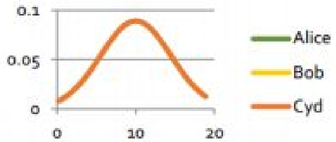


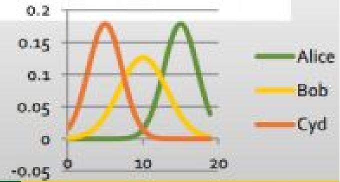


From: David Duvenaud and James Lloyd. Introduction to probabilistic programming. University of Cambridge, 2013.

Why PPLs - Microsoft Xbox Live True Skill example

```
// prior distributions, the hypothesis
let skill() = sample (Gaussian(10.0,20.0))
let Alice,Bob,Cyd = skill(),skill(),skill()
// observe the evidence
let performance player = sample (Gaussian(player,1.0))
observe (performance Alice > performance Bob) //Alice beats Bob
observe (performance Bob > performance Cyd) //Bob beats Cyd
observe (performance Alice > performance Cyd) //Alice beats Cyd
// return the skills
Alice,Bob,Cyd
```



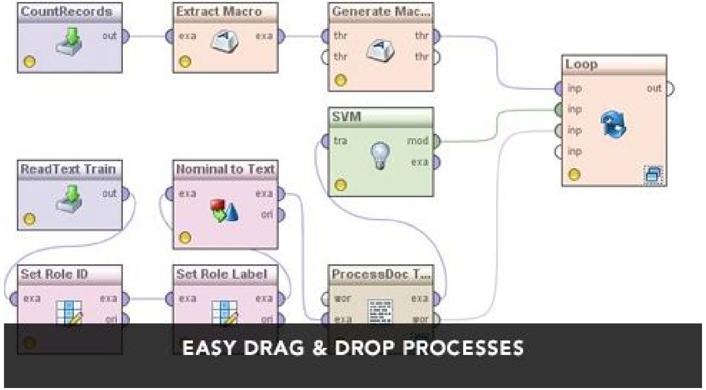


MICROSOFT RESEARCH

From: T Minka and J Winn and J Guiver and D Knowles. Infer .NET 2.5. Microsoft Research Cambridge, 2012.

PPLs shortcomings

- Forces the users to learn yet another syntax



From: rapidminer.com

- Interface doesn’t resemble common data analysis tools (Excel, RapidMiner, Weka Knowledge Flow, ...)

Solution:

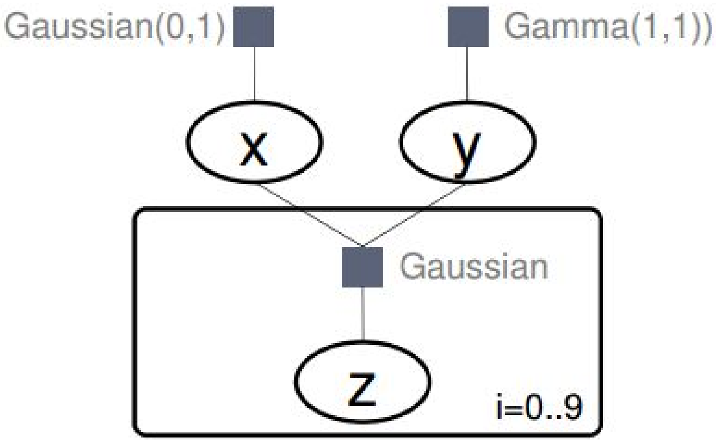
Visual Programming Concepts for Probabilistic Programming Languages

Visual Programming Concepts for PPLs

Visual Programming Concepts for PPLs

How?

- Node-based programming editor → compiles to PPL syntax



From: John Winn and Tom Minka. Probabilistic Programming. Machine Learning Summer School, 2009.

```
x = Gaussian(0, 1)

y = Gamma(1, 1)

for(i in 0..9)
  z = Gaussian(x, y)
```

Visual Programming Concepts for PPLs

- Hypothesis:
  - More intuitive
  - Easier to learn
  - Faster to develop in
- Tricky issues:
  - Functions
  - Recursion
  - Arrays of arrays of ...
  - Mutation: x=x+1
  - Objects

References

- scikit-learn. Choosing the right estimator. http://scikit-learn.org/stable/tutorial/machine_learning_map/
- Olivier Grisel. SciPy 2013 Keynote: Trends in Machine Learning and the SciPy community.
- Rob Knies. Infer.NET: Machine Learning Tailor-Made. Inside Microsoft Research, 2013.
- David Duvenaud and James Lloyd. Introduction to probabilistic programming. University of Cambridge, 2013.
- John Winn and Tom Minka. Probabilistic Programming. Machine Learning Summer School, 2009.
- Andrew D. Gordon and Thomas A. Henzinger and Aditya V. Nori and Sriram K. Rajamani. Probabilistic Programming. International Conference on Software Engineering (ICSE Future of Software Engineering), 2014.
- T Minka and J Winn and J Guiver and D Knowles. Infer .NET 2.5. Microsoft Research Cambridge, 2012.