A Live, Multiple-Representation Probabilistic Programming Environment for Novices

Maria Gorinova, Advait Sarkar, Alan Blackwell, Don Syme
What is Probabilistic Programming?

- Probabilistic graphical models:
  - Vertices are random variables
  - Edges are direct dependencies

- Probabilistic programming language:

```csharp
let Coin1 = new Bernoulli(0.5)
let Coin2 = new Bernoulli(0.5)
let BothHeads = Coin1 && Coin2
```
How is Probabilistic Programming different?

- **Conventional programming:** each variable has a single value – it holds a specific number, object, function vs.

- **Probabilistic programming:** each variable represents an entire probability distribution
How is Probabilistic Programming different?

• **Conventional programming:** each variable has a single value – it holds a specific number, object, function

  vs.

• **Probabilistic programming:** each variable represents an entire probability distribution
How is Probabilistic Programming different?

• Conventional programming: no side effects (in general)
  
  vs.

• Probabilistic programming: observations have side effects

```javascript
let Coin1 = new Boolean();
let Coin2 = new Boolean();
...
let BothHeads = Coin1 && Coin2
...
BothHeads = false
```
How is Probabilistic Programming different?

- **Conventional programming:** no side effects (in general)
- **Probabilistic programming:** observations have side effects
Aims

• Design a tool that communicates the different conceptual basis of probabilistic programming

• Do that by visualising:
  • The graphical models
  • The posterior distributions of random variables
```fsharp
let Coin1 = Variable.Bernoulli(0.5)
let Coin2 = Variable.Bernoulli(0.5)

let BothHeads = Coin1 && Coin2
```
let Coin1 = Variable.Bernoulli(0.5)
let Coin2 = Variable.Bernoulli(0.5)

let BothHeads = Coin1 && Coin2
let visitedAsia = Variable.Bernoulli(0.01)
let smokes = Variable.Bernoulli(0.3)

let hasBronchitis = Variable.New<bool>()
let hasLungCancer = Variable.New<bool>()
let hasTuberculosis = Variable.New<bool>()

hasBronchitis.SetTo(Variable.Bernoulli(0.3))

begin
   use ifAsia = Variable.If(visitedAsia)
   hasTuberculosis.SetTo(Variable.Bernoulli(0.05))
end

begin
   use ifNotAsia = Variable.IfNot(visitedAsia)
   hasTuberculosis.SetTo(Variable.Bernoulli(0.01))
end

begin
   use ifSmokes = Variable.If(smokes)
   hasLungCancer.SetTo(Variable.Bernoulli(0.1))
end

begin
   use ifNotSmokes = Variable.IfNot(smokes)
   hasLungCancer.SetTo(Variable.Bernoulli(0.01))
end

let hasTorC = hasTuberculosis ||| hasLungCancer
User Study Results

<table>
<thead>
<tr>
<th>Measurement</th>
<th>IDE Median</th>
<th>Text Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>191 s</td>
<td>220 s</td>
</tr>
<tr>
<td>Keystrokes</td>
<td>28.5</td>
<td>93</td>
</tr>
</tbody>
</table>
Thank you!

github.com/mgorinova/Infer.IDE
More Results: Exercise Phase

<table>
<thead>
<tr>
<th>Measurement</th>
<th>V</th>
<th>p-value</th>
<th>IDE median</th>
<th>Text median</th>
<th>Difference median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>611</td>
<td>0.004162</td>
<td>288 sec</td>
<td>359 sec</td>
<td>80 sec</td>
</tr>
<tr>
<td>Keystrokes</td>
<td>302</td>
<td>0.002044</td>
<td>15.5</td>
<td>79</td>
<td>22.5</td>
</tr>
<tr>
<td>Backspaces</td>
<td>367.5</td>
<td>0.02397</td>
<td>3.5</td>
<td>9</td>
<td>0</td>
</tr>
</tbody>
</table>

(a) Results for all Part 2 exercises.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>V</th>
<th>p-value</th>
<th>IDE median</th>
<th>Text median</th>
<th>Difference median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>205</td>
<td>0.2781</td>
<td>323 sec</td>
<td>337 sec</td>
<td>18 sec</td>
</tr>
<tr>
<td>Keystrokes</td>
<td>167</td>
<td>0.07112</td>
<td>34</td>
<td>90</td>
<td>64.5</td>
</tr>
<tr>
<td>Backspaces</td>
<td>188.5</td>
<td>0.2472</td>
<td>9</td>
<td>12</td>
<td>3</td>
</tr>
</tbody>
</table>

(b) Results for debugging exercises.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>V</th>
<th>p-value</th>
<th>IDE median</th>
<th>Text median</th>
<th>Difference median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>101</td>
<td>0.001667</td>
<td>245 sec</td>
<td>369 sec</td>
<td>92 sec</td>
</tr>
<tr>
<td>Keystrokes</td>
<td>12</td>
<td>0.002448</td>
<td>0</td>
<td>1.5</td>
<td>0</td>
</tr>
<tr>
<td>Backspaces</td>
<td>31</td>
<td>0.0331</td>
<td>0</td>
<td>36.5</td>
<td>0</td>
</tr>
</tbody>
</table>

(c) Results for observation exercises.
Results

• Increased performance in completing probabilistic programming exercises compared to a plain text editor:
  • IDE learners vs plain-text learners (*between users*)
    • Time difference (per part 1 exercise): 9 s
    • Keystrokes difference (per part 1 exercise): 64.5 keystrokes
  • IDE tasks vs plain-text tasks (*within user*)
    • Time difference (per part 2 exercise): 80 s
    • Keystrokes difference (per part 2 exercise): 22.5 keystrokes
User Study

• Aim: study the effect on learning PP concepts

• Structure:
  • **Learning Phase:** study theory with the help of a workbook. Complete 4 interleaved tasks
  • **Exercises Phase:** complete 8 additional exercises

<table>
<thead>
<tr>
<th>Part 1 (learning)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to:</td>
</tr>
<tr>
<td>• F#’s syntax</td>
</tr>
<tr>
<td>• Probabilistic programming</td>
</tr>
<tr>
<td>• Infer.NET</td>
</tr>
<tr>
<td>Interleaves 4 exercises</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part 2 (exercise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 additional exercises</td>
</tr>
</tbody>
</table>
Exercises

• Debugging:

  “You will be presented with a graphical model that should represent the problem described below. There are several mistakes in the code. Find and correct all mistakes.”

• Observation:

  “Please explain out loud what problem the model describes. Are variables X and Y independent? What is the probability of X being true?”
Experiment Structure

<table>
<thead>
<tr>
<th>Group A</th>
<th>Group B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning with the IDE</td>
<td>Learning without the IDE</td>
</tr>
</tbody>
</table>

- **Group A.1**: IDE, text editor
- **Group A.2**: IDE, text editor
- **Group B.1**: IDE, text editor
- **Group B.2**: IDE, text editor

**Order of Usage of Editors**
- **IDE first**: Group A.1, Group A.2
- **Text Editor first**: Group B.1, Group B.2

**Order of Exercises**
- **Straight order**: Group ID: A/B 00, Group ID: A/B 01
- **Reverse order**: Group ID: A/B 10, Group ID: A/B 11

**Between-subject study to evaluate as a learning tool**

**Within-subject study to evaluate usability**
High-Level Overview

Checking Phase

Source Code

Checker Module

RVs

Injection Phase

Injection

Graph Module
Results: Confidence