Programming with "Big Code":
Lessons, Techniques, Applications

Pavol Bielik, Veselin Raychev, Martin Vechev
Department of Computer Science
ETH Zurich

Work @ ETH Zurich

Work on “Big Code” started a few years ago


Prof. Martin Vechev


Prof. Andreas Krause


Veselin Raychev


Pavol Bielik


Svetoslav Karaivanov


Christine Zeller


Pascal Roos

Code Completion with Statistical Language Models, PLDI 2014
Machine Translation for Programming Languages, Onward 2014
Predicting Program Properties from “Big Code”, POPL 2015
Fast and Precise Statistical Code Completion, ETH TR
Statistical Feedback Generation for Programs, ETH TR
Programming with Big Code: Lessons, Techniques and Applications, SNAPL 2015

Applications

[PLDI 14]
SLANG: Code Completion

```
Intent i = new Intent();
           ?
ctx.sendBroadcast(i);
```

```
Context ctx;
Activity currentActivity;

void test(boolean no_bars) {
    WebView view = new WebView(ctx);
    if (no_bars) {
        view.setVerticalScrollBarEnabled(false);
        view.setHorizontalScrollBarEnabled(false);

    }
    view.

}
```

```
currentActivity.setContentView(View view) : void - A
view.loadUrl(String url) : void - WebView
```

All of these benefit from the “Big Code” and lead to applications not possible with previous techniques

Applications

[PLDI 14]
SLANG: Code Completion

```
Intent i = new Intent();
           ?
ctx.sendBroadcast(i);
```

[Onward 14]
Programming Language Translation

```
P( Java | C# )
P( C# | Java )
P( Java )
```

All of these benefit from the “Big Code” and lead to applications not possible with previous techniques

Applications

[PLDI 14]
SLANG: Code Completion

```
Intent i = new Intent();
           ?
ctx.sendBroadcast(i);
```

[Onward 14]
Programming Language Translation

```
P( Java | C# )
P( C# | Java )
P( Java )
```

[submitted]
Statistical Feedback Generation

```
...                          likely error
for x in range(a):
    print a[x]
```

All of these benefit from the “Big Code” and lead to applications not possible with previous techniques

Applications

[PLDI 14]
SLANG: Code Completion

```
Intent i = new Intent();
          ?
ctx.sendBroadcast(i);
```

[Onward 14]
Programming Language Translation

```
P( Java | C# )
P( C# | Java )
P( Java )
```

[POPL 15]
JSNice: Deobfuscation
Type Prediction

[submitted]
Statistical Feedback Generation

likely error

```
...
for x in range(a):
    print a[x]
```

All of these benefit from the “Big Code” and lead to applications not possible with previous techniques

Probabilistic Programming Systems: Dimensions

- Applications
- Intermediate Representation
- Analyze Program (PL)
- Train Model (ML)
- Query Model (ML)

Probabilistic Programming Systems: Dimensions

Applications

Intermediate Representation

Analyze Program (PL)

Train Model (ML)

Query Model (ML)

What is a generic metric for code?

✔ Cross Entropy → ✗ Code Completion

✔ BLEU Score → ✗ Program Translation

Traditional metrics might not be indicative of client performance

Probabilistic Programming Systems: Dimensions

Applications

Intermediate Representation

Analyze Program (PL)

Train Model (ML)

Query Model (ML)

What is the best program representation?

Probabilistic Programming Systems: Dimensions

Applications

Intermediate Representation

Analyze Program (PL)

Train Model (ML)

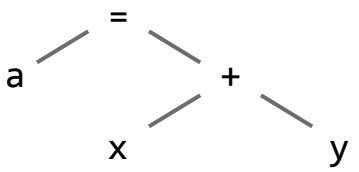
Query Model (ML)

What is the best program representation?

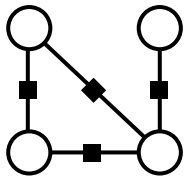
Sequences

```
req    → {<open, 0>, <send, 0>}
source → {..., <open, 2>}
```

Trees



Graphical Models



Feature Vectors

```
req    → (0,0,1,1,0)
source → (1,0,0,0,0)
...
```

Probabilistic Programming Systems: Dimensions

Applications

Intermediate Representation

Analyze Program (PL)

Train Model (ML)

Query Model (ML)

What is the best program representation?

Choosing the right representation is crucial

Feedback Generation: Sequence representations

Allamanis et. al. [2013]	46.4%
Hsiao et. al. [2014]	50.8%
Incorporate semantic information	75.3%
Incorporate dataflow analysis	**86.3%**

Probabilistic Programming Systems: Dimensions

Applications

Intermediate Representation

Analyze Program (PL)

Train Model (ML)

Query Model (ML)

How to extract program representation?

SLANG (APIs): alias and typestate analysis
JSNice (Variable Names): scope and alias analysis
Feedback Generation: alias, control-flow and typestate analysis

```
req.open("GET", source, false);
```

```
req    → {<open, 0>, <send, 0>}
source → {..., <open, 2>}
```

Probabilistic Programming Systems: Dimensions

Applications

Intermediate Representation

Analyze Program (PL)

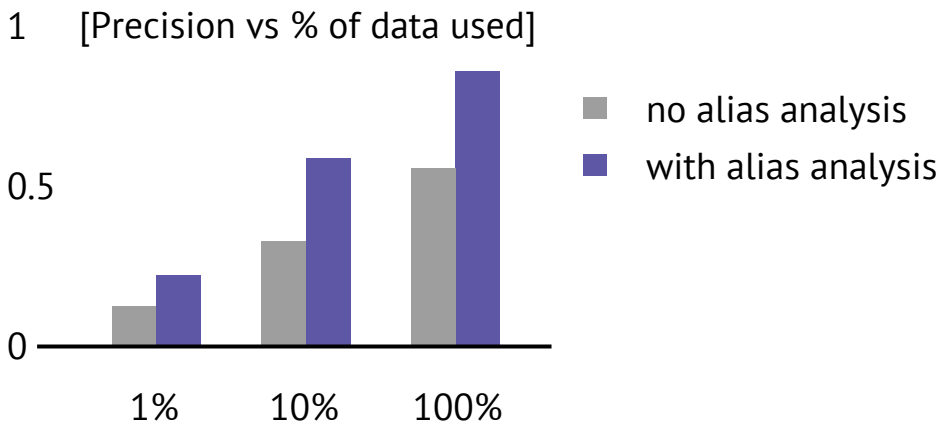
Train Model (ML)

Query Model (ML)

How to extract program representation?

SLANG (APIs): alias and typestate analysis
JSNice (Variable Names): scope and alias analysis
Feedback Generation: alias, control-flow and typestate analysis

Design scalable yet precise enough algorithms



Probabilistic Programming Systems: Dimensions

Applications

Intermediate Representation

Analyze Program (PL)

Train Model (ML)

Query Model (ML)

What is the suitable probabilistic model?

N-gram language model

Probabilistic context-free grammars

Neural networks

(Structured) Support vector machine

Conditional Random Fields

...

Probabilistic Programming Systems: Dimensions

Applications

Intermediate Representation

Analyze Program (PL)

Train Model (ML)

Query Model (ML)

What is the suitable probabilistic model?

N-gram language model

Probabilistic context-free grammars

Neural networks

(Structured) Support vector machine

Conditional Random Fields

Structured prediction is critical

Baseline	25.3%
Independent	54.1%
Structured	**63.4%**

Programming with "Big Code"

Applications	Code completion, Program synthesis, Translation, Deobfuscation, Feedback generation
Intermediate Representation	Sequences (sentences), Translation Table, Graphical Models, Trees, Feature Vectors
Analyze Program (PL)	alias analysis, control-flow analysis, scope analysis, typestate analysis
Train Model (ML)	Neural Networks, SVM, Structured SVM, N-gram language model
Query Model	argmax P(y | x), $y \in \Omega$

Programming with "Big Code"

Applications	Code completion Deobfuscation	Program synthesis	Translation Feedback generation
Intermediate Representation	Sequences (sentences) Trees	Translation Table	Graphical Models Feature Vectors
Analyze Program (PL)	alias analysis scope analysis	control-flow analysis typestate analysis	
Train Model (ML)	Neural Networks N-gram language model	SVM	Structured SVM
Query Model	argmax P(y | x) $y \in \Omega$		Greedy MAP Inference

More information and tutorials at: http://www.nice2predict.org/ http://www.srl.inf.ethz.ch/spas.php

NICE 2 Predict

General framework

http://www.nice2predict.org/

We have open-sourced our prediction engine and we are extending it with new capabilities

Upcoming PLDI'15 tutorial

Programming with "Big Code"

Applications	Code completion Deobfuscation	Program synthesis	Translation Feedback generation
Intermediate Representation	Sequences (sentences) Trees	Translation Table	Graphical Models Feature Vectors
Analyze Program (PL)	alias analysis scope analysis	control-flow analysis typestate analysis	
Train Model (ML)	Neural Networks N-gram language model	SVM	Structured SVM
Query Model	argmax P(y | x) $y \in \Omega$		Greedy MAP Inference

More information and tutorials at: http://www.nice2predict.org/ http://www.srl.inf.ethz.ch/spas.php

NICE 2 Predict