Probabilistic Models of Source Code: An Overview

Miltos Allamanis, University of Edinburgh
November, 2016
Developers implicitly embed **knowledge** in code that may be useful for the same or other projects.

Mine the hidden knowledge to create **smart** software engineering tools.
Probabilistic reasoning is a *principled* way of handling *ambiguities* and *partial information*. 
Natural Language Processing (NLP)

Some Knowledge of Linguistics

Data: Corpora of Text, Speech etc

Models of Aspects of Natural Language

❯ Resolve language ambiguities with principled probabilistic models of language.
❯ Learn models from annotated corpora.

Parsing

Named Entity Recognition

Machine Translation....
Natural Language Processing with Machine Learning

Resolve language ambiguities with principled probabilistic models of language.
Learn models from annotated corpora.
Machine Learning Models of Source Code

“All models are wrong, some are useful” - George Box

Software Engineers

Codebases

Machine Learning Models of Aspects of Source Code

Software Engineering Tools
Outline

Very Brief Intro to Machine Learning

A Taxonomy of Models

Applications
Machine Learning

Machine Learning Model
Designed by humans

Model Parameters
Learned from data
Finding a good model

Underfitting

Overfitting

graphs from http://antianti.org/?p=175
Learning Model Parameters

❯ Optimize objective function in training set
❯ Use computational methods of optimization
Learning the Distribution of Model Parameters

> Learn the distribution of parameters given some data.
> Use computational methods

\[ p(\mu \mid Data) = \frac{p(Data \mid \mu) \cdot p(\mu)}{p(Data)} \]
A Spectrum of Common Machine Learning Tasks

Classification
- Supervised

Clustering
- Unsupervised
A Spectrum of Common Machine Learning Tasks

Joint Classification

Learning Features
Automatic Evaluation in Machine Learning

Imperfect measures of performance such as

❯ Prediction Accuracy
❯ Model Fit

❯ Quantify performance in a **reproducible** manner

❯ **Drive** improvement of systems in a **measurable** way
Machine Learning Models
Natural Languages vs Programming Languages
Part II
A Taxonomy of Models
```csharp
for (int i = 0; i < 10; i++){
    Console.WriteLine(i);
}
```
Probabilistic Models of Source Code

**Code Generating Models**

$P_D(\mathcal{C} | C(\mathcal{C}))$

- Language Models
- Translation Models
- Multimodal Models

**Code Representational Models**

$P_D(\pi | f(\mathcal{C}))$

- Structured Prediction
- Distributed Representations

**Pattern Mining Models**

$P_D(f(\mathcal{C})) = \sum_l P_D(f(\mathcal{C}) | l) P(l)$

- Latent Variable Models

---

Probabilistic Generative Models

- Code Generating Models
- Code Representational Models
- Pattern Mining Models
Simplifying assumptions about how code is generated
Language Models

Assign a non-zero probability to every piece of valid code

$$P(c)$$
n-gram Language Models

code snippet:

```java
public void execute(Runnable task) {
    if (task == null)
        throw new NullPointerException();
    ForkJoinTask<?> job;
    if (task instanceof ForkJoinTask<?>)
        job = (ForkJoinTask<?>) task;
    else
        job = new ForkJoinTask.AdaptedRunnableAction(task);
    externalPush(job);
}
```
public void execute(Runnable task) {
    if (task == null)
        throw new NullPointerException();
    ForkJoinTask<?> job;
    if (task instanceof ForkJoinTask<?>) // avoid re-wrap
        job = (ForkJoinTask<?>) task;
    else
        job = new ForkJoinTask.AdaptedRunnableAction(task);
    externalPush(job);
}
n-gram Language Models

public void execute(Runnable task) {
    if (task == null)
        throw new NullPointerException();
    ForkJoinTask<?> job;
    if (task instanceof ForkJoinTask<?>) // avoid re-wrap
        job = (ForkJoinTask<?>) task;
    else
        job = new ForkJoinTask.AdaptedRunnableAction(task);
    externalPush(job);
}

\[
P(t_0 \ldots t_M) = \prod_{m=0}^{M} P(t_m | t_{m-1} \ldots t_{m-n+1})
\]
n-gram Language Models

```java
public void execute(Runnable task) {
    if (task == null)
        throw new NullPointerException();
    ForkJoinTask<?> job;
    if (task instanceof ForkJoinTask<?>) // avoid re-wrap
        job = (ForkJoinTask<?>) task;
    else
        job = new ForkJoinTask.AdaptedRunnableAction(task);
    doSubmit(job);
}
```
Cache n-gram Language Models

\[ P(t_i|h, \text{cache}) = \lambda \cdot P_{n\text{-gram}}(t_i|h) + (1 - \lambda) \cdot P_{\text{cache}}(t_i|h) \]
Syntactic model of source code, i.e. model how AST is generated
Tree Generation Model: Probabilistic Context Free Grammars (PCFG)

\[ E \rightarrow E + E \quad \text{(prob 0.7)} \]
\[ E \rightarrow T \quad \text{(prob 0.3)} \]
\[ F \rightarrow (E) \quad \text{(prob 0.1)} \]
\[ T \rightarrow F \ast F \quad \text{(prob 0.6)} \]
\[ T \rightarrow F \quad \text{(prob 0.4)} \]
\[ F \rightarrow id \quad \text{(prob 0.9)} \]
Generating from a PCFG
Generating from a PCFG

ForStatement
  Initialization
  Expression
  Expression
  Body
Generating from a PCFG
Generating from a PCFG

ForStatement
  - Initialization
  - Expression
  - Expression
  - Body

Single Variable Declaration
  - Type
  - Name
  - Initializer
Generating from a PCFG

ForStatement
- Initialization
  - Single Variable Declaration
    - Type
      - int
    - Name
    - Initializer
  - Expression
  - Expression
- Body
Generating from a PCFG

ForStatement

- Initialization
  - Single Variable Declaration
    - Type: int
    - Name: i
    -Initializer

- Expression
- Expression
- Body

Infix Expression
- Left Operand
- Operator: <
- Right Operand: Numeric Literal: 10
ForStatement
  -- Initialization
    Single Variable Declaration
      Type
        int
      Name
        i
      Initializer
        Numeric Literal
          0
    Left Operand
    Operator
      ???
    Right Operand
  Expression
    Infix Expression
      Expression
      Expression
  Body
Graph Language Models
Evaluation Metrics for Language Models

Log Probability

\[ Q(c, P_D) = -\log_2(P_D(c)) \]

Cross Entropy

\[ H(c, P_D) = -\frac{1}{M} \log_2 P_D(c) \]
\( t^* = \arg \max P_D(t|s) = \arg \max P_D(s|t)P_D(t) \)
<table>
<thead>
<tr>
<th>Subphase</th>
<th>Output</th>
<th>Example output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel data collection</td>
<td>Method pairs</td>
<td>C#: Console.WriteLine(&quot;Hello World!&quot;);</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Java: System.out.println(&quot;Hello World!&quot;);</td>
</tr>
<tr>
<td>Word alignment</td>
<td>Aligned method pairs</td>
<td>C#: Console.WriteLine(&quot;Hello World!&quot;);</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Java: System.out.println(&quot;Hello World!&quot;);</td>
</tr>
<tr>
<td>Phrase table construction</td>
<td>Phrase table</td>
<td>C# phrase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Console</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WriteLine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Multimodal Models of Source Code

“get the first letter of each word in string and uppercase”

Non-Code Modality

Synthesize/Score Code Snippet

```
string s;
string[] words = s.ToUpper().split(' ');  
string[] firstLetters = new string[words.Length];
for (int i=0; i < words.Length; i++) {
    firstLetters[i] = words.Substring(0,1);
}
```
- all elements to small letters for each line and sort
- all elements to small letters for each line and order
- all elements to lowercase for each line and sort
- all elements to lowercase for each line and order
- all elements lowercase for each line and sort
- all elements lowercase for each line and order
- all elements to lower case for each line and sort
- all elements to lower case for each line and order
- all elements to small letters for each new line and sort

```csharp
string result = String.Join("\n", input_string.Split("\n\n").Select((string x) => x.ToLower()).OrderBy(x => x));
```
Representational Models of Code

\[ P_D(\pi | f(c)) \]
Structured Prediction

Output variables are mutually dependent or constrained.
function chunkData(e, t) {
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t) {
    if (i + t < r) {
      n.push(e.substring(i, i + t));
    } else {
      n.push(e.substring(i, r));
    }
  }
  return n;
}

(a) JavaScript program with minified identifier names

Unknown properties (variable names):
? ? t ? n ? r ? i ?

Known properties (constants, APIs):
0 [ ] length push ...

function chunkData(str, step) {
  var colNames = [];
  var len = str.length;
  var i = 0;
  for (; i < len; i += step) {
    if (i + step < len) {
      colNames.push(str.substring(i, i + step));
    } else {
      colNames.push(str.substring(i, len));
    }
  }
  return colNames;
}

(e) JavaScript program with new identifier names and types

Raychev et al. “Predicting Program Properties from Big Code”, 2015
Localized vs Distributed (Vector) Representations

\[ P_D(\pi | f(c)) \]

\[ \mathbb{R}^D \]
Distributed Representations

Allamanis et al., “Suggesting accurate method and class names”, 2015
Distributed Representations

Allamanis et al., “Suggesting accurate method and class names”, 2015
Distributed Representations

Allamanis et al., “Suggesting accurate method and class names”, 2015
Setters vs Getters in bigbluebutton

\[ W(“woman”) - W(“man”) \approx W(“aunt”) - W(“uncle”) \]
\[ W(“woman”) - W(“man”) \approx W(“queen”) - W(“king”) \]
Pattern Mining Models of Code

\[ P_D(f(c)) = \sum_l P_D(f(c)|l)P(l) \]
Latent Variable Models
Movshovitz-Attias & Cohen, “KB-LDA: Jointly learning a knowledge base of hierarchy, relations, and facts”, 2015
Part III
Applications of Models of Source Code
Machine learning helps each application area by learning to resolve (some) ambiguities from data.
Machine Learning Models
Applications: A (Very) Brief Tour
Recommender Systems
Proksch et al. 2015  Bruch et al. 2009
<table>
<thead>
<tr>
<th>Token Role</th>
<th>Construction Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data type $T$</td>
<td>TYPE[$T$]</td>
</tr>
<tr>
<td>Variable $x$</td>
<td>VAR[typeof($x$)]</td>
</tr>
<tr>
<td>Literal $v$</td>
<td>LIT[typeof($v$)]</td>
</tr>
<tr>
<td>Function decl $m$</td>
<td>FUNC[typeof($m$), lexeme($m$), paralist($m$), rettype($m$)]</td>
</tr>
<tr>
<td>Function call $m$</td>
<td>CALL[typeof($m$), lexeme($m$), paracount($m$), rettype($m$)]</td>
</tr>
<tr>
<td>Parameter $x$</td>
<td>PARA[typeof($x$)]</td>
</tr>
<tr>
<td>Field $f$</td>
<td>FIELD[typeof($f$), lexeme($f$)]</td>
</tr>
<tr>
<td>Operator $o$</td>
<td>OP[name($o$)]</td>
</tr>
<tr>
<td>Cast ($T$)</td>
<td>CAST[$T$]</td>
</tr>
<tr>
<td>Keyword</td>
<td>To corresponding reserved token</td>
</tr>
<tr>
<td>Block open &amp; close</td>
<td>To corresponding reserved token</td>
</tr>
<tr>
<td>Special literal</td>
<td>To corresponding reserved token</td>
</tr>
<tr>
<td>Unknown</td>
<td>To special lexical token LEX</td>
</tr>
</tbody>
</table>
Fowkes & Sutton, “Parameter-Free Probabilistic API Mining across GitHub”, 2016
Statistical Code Migration
<table>
<thead>
<tr>
<th>Subphase</th>
<th>Output</th>
<th>Example output</th>
</tr>
</thead>
</table>
| Parallel data collection | Method pairs    | C#:
                               Console . WriteLine ( "Hello World!" );
                               Java:
                               System . out . println ( "Hello World!" ); |
| Word alignment           | Aligned method pairs | C#:
                               Console . WriteLine ( "Hello World!" );
                               Java:
                               System . out . println ( "Hello World!" ); |
| Phrase table construction | Phrase table    | ![Phrase table](phrase_table.png)                   |

\[ \mathbf{t}^* = \arg \max P_{\mathcal{D}}(\mathbf{t}|\mathbf{s}) = \arg \max P_{\mathcal{D}}(\mathbf{s}|\mathbf{t}) P_{\mathcal{D}}(\mathbf{t}) \]

Raychev et al. Onwards 2014

Nguyen et al. FSE 2013, ASE 2015
Code Defects
```java
1 for (int i = 0; i < scorers.length; i++) { 
2    if (scorers[i].nextDoc() == NO_MORE_DOCS) 
3        lastDoc = NO_MORE_DOCS; 
4    return; 
5 } 
6 ```
Netty (2013-08-20)
File: ThreadPerChannelEventLoopGroup.java

Entropy dropped after bugfix: **4.6257**

```java
if (isTerminated()) {
    // Before (entropy = 5.96485):
    - terminationFuture.setSuccess(null);
    // After (entropy = 1.33915):
    + terminationFuture.trySuccess(null);
```

Function `to_s` has appeared 12 times and `split` has appeared 29 times, and they've appeared 0 times together.

Diagram:

Source Files → Tokenization → Token Sequences → N-gram Model Building → Ranked Token Sequences → Bug Detection → Potential Bugs

References:
- Fast et al. CHI 2014
- Wang et al. ASE 2016
1. FILE * fp1 = fopen("myfile.txt", "r");
2. FILE * fp2 = fdopen(fd, "w");
3. fread(buffer, n, 1, fp1);
4. fwrite(buffer, n, 1, fp2);
5. fclose(fp1);
6. fclose(fp2);
Code Synthesis
Archive your missed calls from Android to Google Drive

```
=SUMIFS(WeeklyHours[totalpay], WeeklyHours[location], "capitol hill", WeeklyHours[title], "barista")
```

### Quirk et al. ACL 2015, 2016

### Gulwani & Marron, ICMD 2014
Allamanis et al. “Bimodal modelling of source code and natural language” ICML 2015
Number of Examples for Learning the Test Task

- No Ranking
- LearnRank

Benchmarks:

1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97 101 105 109 113 117 121

Number of Examples:

0 1 2 3 4 5 6

Singh & Gulwani “Predicting a Correct Program in Programming By Example” 2015
Code Summarization
protected int code;
protected String reason;

/** Construct StatusLine */
public StatusLine(int c, String r)
{  code=c;
   reason=r;
}

/** Create a new copy of the request-line*/
public Object clone()
{
}

/** Indicates whether some other Object is "equal to" this StatusLine
public boolean equals(Object obj)
{

public String toString()
{

public int getCode()
{

public String getReason()
{   }
Summarization via Method Name Prediction

<table>
<thead>
<tr>
<th>Target</th>
<th>Attention Vectors</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_1 ) set</td>
<td>( \alpha = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>( \kappa = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td></td>
</tr>
<tr>
<td>( m_2 ) use</td>
<td>( \alpha = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td>0.974</td>
</tr>
<tr>
<td></td>
<td>( \kappa = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td></td>
</tr>
<tr>
<td>( m_3 ) browser</td>
<td>( \alpha = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td>0.969</td>
</tr>
<tr>
<td></td>
<td>( \kappa = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td></td>
</tr>
<tr>
<td>( m_4 ) cache</td>
<td>( \alpha = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td>0.583</td>
</tr>
<tr>
<td></td>
<td>( \kappa = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td></td>
</tr>
<tr>
<td>( m_5 ) END</td>
<td>( \alpha = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td>0.066</td>
</tr>
<tr>
<td></td>
<td>( \kappa = \langle s \rangle { \text{this}.\text{use Browser Cache} = \text{use Browser Cache}; } \langle /s \rangle )</td>
<td></td>
</tr>
</tbody>
</table>

**Attention Visualization**

1. Source Code (C#):
   ```csharp
   public int TextWidth(string text) {
       TextBlock t = new TextBlock();
       t.Text = text;
       return (int)Math.Ceiling(t.ActualWidth);
   }
   ```
   **Descriptions:**
   a. Get rendered width of string rounded up to the nearest integer
   b. Compute the actual textwidth inside a textblock

2. Source Code (C#):
   ```csharp
   var input = "Hello";
   var regEx = new Regex("World");
   return !regEx.IsMatch(input);
   ```
   **Descriptions:**
   a. Return if the input doesn't contain a particular word in it
   b. Lookup a substring in a string using regex
Coding Conventions
ForkJoinTask<?> job; if (task instanceof ForkJoinTask<?>) // avoid re-wrap job = (ForkJoinTask<?>) task; else job = new ForkJoinTask.AdaptedRunnableAction(task); externalPush(job);

Suggestions
1. job (30%)
2. task (20%)
3. tsk (15%)
function chunkData(e, t) {
    var n = [];
    var r = e.length;
    var i = 0;
    for (; i < r; i += t) {
        if (i + t < r) {
            n.push(e.substring(i, i + t));
        } else {
            n.push(e.substring(i, r));
        }
    }
    return n;
}

// str: string, step: number, return: Array */
function chunkData(str, step) {
    var colNames = [];
    var len = str.length;
    var i = 0;
    for (; i < len; i += step) {
        if (i + step < len) {
            colNames.push(str.substring(i, i + step));
        } else {
            colNames.push(str.substring(i, len));
        }
    }
    return colNames;
}

(a) JavaScript program with minified identifier names

Unknown properties (variable names):

(b) Known and unknown name properties

Known properties (constants, APIs):

(c) Dependency network

(d) Result of MAP inference

Raychev et al. POPL 2015
\[ \psi = \text{tree}(x, \lambda i_1, i_2, i_3, i_4 \rightarrow \exists t. \text{ls}(i_2, t, \lambda i_5, i_6, i_7, i_8 \rightarrow \top) * \text{ls}(t, t, \lambda i_9, i_{10}, i_{11}, i_{12} \rightarrow \top)). \]
Probabilistic Models of Source Code

What was *not* discussed

- Non-Probabilistic Models
- Models with *no* Learning Component
- Pure Information Retrieval Models
- Off-the-shelf models: topic models, etc.
- Things that don’t Involve Code

Bag-of-Word Representations

with hand-crafted features
Challenges Ahead...
```javascript
scope.$watch(watchExpr, function ngSwitchWatchAction(value) {
    var i, ii;
    for (i = 0, ii = previousElements.length; i < ii; ++i) {
        previousElements[i].remove();
    }
    previousElements.length = 0;

    for (i = 0, ii = selectedScopes.length; i < ii; ++i) {
        var selected = selectedElements[i];
        selectedScopes[i].$destroy();
        previousElements[i].$animate.leave(selected, function() {
            previousElements.splice(i, 1);
        });
    }
}

selectedElements.length = 0;
selectedScopes.length = 0;

if (!selectedTranscludes || ngSwitchController.cases['!' + value] || ngSwitchController.cases['=' + value]) {
    scope.$eval(attr.change);
    for (var selectedTransclude in selectedTranscludes, function(selectedTransclude) {
        var selectedScope = scope.$new();
        selectedScopes.push(selectedScope);
        scope.$apply();
    });
```

(the end)