Today

A bird’s view of the area

Course organization
  objectives, your tasks, grading, howto’s
Learning and probabilistic models based on Big Data have revolutionized entire fields.

Natural Language Processing (e.g., machine translation)

Computer Vision (e.g., image captioning)

Medical Computing (e.g., disease prediction)

Can we bring this revolution to programmers?
Machine Learning for Programs

Task ➔ Statistical Programming Tool ➔ Solution

Probabilistic model

number of repositories

15 million repositories
Billions of lines of code
High quality, tested, maintained programs

last 5 years
Why now?

**Advances in Programming Languages**
[Automated Reasoning, Synthesis, Constraint Solving]

**Advances in Machine Learning**
[Deep Learning, Graphical Models, Language Models]

Data
[> 15 million public repositories]

Confluence of streams

Machine learning-based programming tools
new rules, new ideas, new opportunities
Sample efforts in this space

**Academia**

@ETH:  [http://plml.ethz.ch](http://plml.ethz.ch)

@Technion:  [http://www.cs.technion.ac.il/~yahave/prime/](http://www.cs.technion.ac.il/~yahave/prime/)

@Rice:  [https://www.cs.rice.edu/~sc40/](https://www.cs.rice.edu/~sc40/)

@MIT:  [http://people.csail.mit.edu/fanl/](http://people.csail.mit.edu/fanl/)

@Microsoft:  
[https://www.microsoft.com/en-us/research/project/program/](https://www.microsoft.com/en-us/research/project/program/)

@Edinburgh:  [http://homepages.inf.ed.ac.uk/csutton/](http://homepages.inf.ed.ac.uk/csutton/)

**Start-ups**

[DEEP CODE](http://deepcode.ai) [ETH spin-off]

[CODOTA](http://codota.com)

[PRODO.AI](https://prodo.ai/)

[NEAR](http://near.ai)

DARPA’s funded the 40M USD MUSE program on this topic
Seminal work in the area

Learning from Large Codebases,

ETH Medal for Outstanding PhD thesis

ACM Doctoral Dissertation, Honorable Mention Award
Only 3rd time in 40-year history of ACM that a PhD from Europe wins this award.
“...regarded as having the potential to open up several promising new avenues of research in the years to come...”
source: https://awards.acm.org/about/2016-doctoral-dissertation

ACM SIGPLAN and Communications of the ACM Research Highlights, 2018
“...significantly advance the state-of-the-art in statistical reasoning on programs and offer the first concrete evidence of the tremendous promise of the overall approach...”
source: http://www.sigplan.org/Highlights/Papers/
@ETH

Probabilistically likely solutions to problems hard to solve otherwise

Publications

- Program Synthesis for Character Level Language Modeling, ICLR'17
- Learning a Static Analyzer from Data, CAV'17
- Statistical Deobfuscation of Android Applications, ACM CCS’16
- Probabilistic Mode for Code with Decision Trees, ACM OOPSLA’16
- PHOG: Probabilistic Mode for Code, ACM ICML’16
- Learning Programs from Noisy Data, ACM POPL’16
- Predicting Program Properties from “Big Code”, ACM POPL’15
- Code Completion with Statistical Language Models, ACM PLDI’14
- Machine Translation for Programming Languages, ACM Onward’14

ML Engines

- apk-deguard.com
- jsnice.org
- nice2predict.org

more: http://plml.ethz.ch

http://deepcode.ai
## Dimensions:
### Machine Learning for Programming

<table>
<thead>
<tr>
<th>Applications</th>
<th>Code completion</th>
<th>Deobfuscation</th>
<th>Program synthesis</th>
<th>Feedback generation</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate Representation</td>
<td>Sequences (sentences)</td>
<td>Translation Table</td>
<td>Trees</td>
<td>Graphical Models (CRFs)</td>
<td>Feature Vectors</td>
</tr>
<tr>
<td>Analyze Program (PL)</td>
<td>typestate analysis</td>
<td>scope analysis</td>
<td>control-flow analysis</td>
<td>alias analysis</td>
<td></td>
</tr>
<tr>
<td>Train Model (ML)</td>
<td>Neural Networks</td>
<td>SVM</td>
<td>PCFG</td>
<td>Structured SVM</td>
<td></td>
</tr>
<tr>
<td>Query Model (ML)</td>
<td>( \text{argmax} \ P(y \mid x) )</td>
<td>( y \in \Omega )</td>
<td>Greedy</td>
<td>MAP inference</td>
<td></td>
</tr>
</tbody>
</table>
ML to write code

```
Camera camera = Camera.open();
camera.setDisplayOrientation(90);
camera.unlock();
SurfaceHolder holder = getHolder();
holder.addCallback(this);
holder.setType(SurfaceHolder.STP);
MediaRecorder r = new MediaRecorder();
r.setCamera(camera);
r.setAudioSource(MediaRecorder.AS);
r.setVideoSource(MediaRecorder.VS);
r.setOutFormat(MediaRecorder.MPEG4);
```

Statistical language models
Recurrent neural networks

Typestate analysis
Alias analysis
ML to Translate Between Languages

C#  Java

Console.WriteLine("Hi")
...

System.out.println("Hi");
...

Phrase-based Statistical Machine Translation

Prefix Parsing of Context-Free Grammars
ML for code deobfuscation

```javascript
function FZ(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;
}

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;
}
```
JSNice.org

- Every country
- 200,000 users
- Top ranked tool

This Page Amsterdam @thispage_ams · Jul 16
Do you write ugly JavaScript code? Not to worry. JSNice will make it look like you are a superstar coder. Yail - buff.ly/1HR4JL7

Ingvar Stepanyan @RReverser · Aug 6
JSNice.org became my must-have tool for code deobfuscation.

Brevity @seekbrevity · Jul 28
JSNice is an amazing tool for de-minifying javascript files. JSNice.org, its great for learning and reverse engineering.

Alvaro Sanchez @alvasavi · Jun 10
This is gold.
Statistical renaming, Type inference and Deobfuscation. jsnice.org

Alex Vanston @mvdot · Jun 7
I've been looking for this for years: JS NICE buff.ly/1pQ5qr #javascript #unminify #deobfuscate #makethereadable

Kamil Tomšík @cztomsik · Jun 6
tell me how this works!
de-minify javascript incl. args, vars & jsdoc impressive! jsnice.org
Learning Phase

- Program analysis
- SSVM learning
  - max-margin training
  - 7M feature functions for names
  - 70K feature functions for types

Prediction Phase

- Program analysis
- MAP inference
- Transform

```
var n = [];
var r = e.length;
var i = 0;
for (; i < r; i += t)
  if (i + t < r)
    n.push(e.subs(i, i + t));
  else
    n.push(e.subs(i, r));
return n;
```

```
var colNames = [];
var len = str.length;
var i = 0;
for (; i < len; i += step)
  if (i + step < len)
    colNames.push(str.subs(i, i + step));
  else
    colNames.push(str.subs(i, len));
return colNames;
```
package a.b.c
class a extends SQLiteHelper
{
    SQLiteDatabase b; public
a(Context ctx) { b =
    getWritableDatabase(); } 
Cursor c(String str) { return b.rawQuery(str); }}

package com.example.dbhelper

class DBHelper extends
SQLiteHelper {
    SQLiteDatabase db;
    public DBHelper(Context ctx) {
        db = getWritableDatabase();
    }
    Cursor execSQL(String str) {
        return db.rawQuery(str);
    }
}

Funny Reddit post/comment

[-] Tycon712 • 3 points 2 days ago
Can someone tell me what the point of using Proguard is if there are tools out there like this?

[permalink embed pocket]

[-] theheartbreakpug • 6 points 2 days ago
As far as I know, this is brand new. I asked the creator of ProGuard a week ago how hard it
is to unobfuscate code after it's run through proguard. He said it strips all the names out of
the code so it's essentially impossible. I'm super impressed by what they've done here.
Nice2Predict.org: scalable structured prediction framework

fully, open sourced, Apache license

used by various groups worldwide

Fast, Approximate MAP inference
Fast, Parallel, Structured SVM and Pseudo-Likelihood Training
Arbitrary factors and indicator functions
### OPPORTUNITY

**Advances in Programming Languages**
- Automated Reasoning, Synthesis, Constraint Solving

**Advances in Machine Learning**
- Deep Learning, Graphical Models, Language Models

**Data**
- > 15 million public repositories

---

### RICH PROBLEM SPACE

**Applications**
- Code completion
- Program synthesis
- Feedback generation
- Translation

**Intermediate Representation**
- Sequences (sentences)
- Translation Table
- Graphical Models (CRFs)
- Feature Vectors

**Analyze Program (PL)**
- typestate analysis
- scope analysis
- control-flow analysis
- alias analysis

**Train Model (ML)**
- Neural Networks
- PCFGs
- N-gram language model
- SVM
- Structured SVM

**Query Model (ML)**
- argmax P(y | x)
  \[ y \in \Omega \]
- Greedy MAP inference

---

### NEW PROBABILISTIC MODELS

1. **Pick** a structure of interest, e.g., trees:

   \[
   T\text{Cond} \;::= \;\varepsilon \mid \text{WriteOp} \; T\text{Cond} \mid \text{MoveOp} \; T\text{Cond}
   \]

   \[
   \text{MoveOp} \;::= \;\text{Up}, \text{Left}, \text{Right}, \text{DownFirst}, \text{DownLast}, \text{NextDFS}, \text{PrevDFS}, \text{NextLeaf}, \text{PrevLeaf}, \text{NextNodeType}, \text{PrevNodeValue}
   \]

   \[
   \text{WriteOp} \;::= \;\text{WriteValue}, \text{WriteType}, \text{WritePos}
   \]

2. **Define** a DSL for expressing functions:
   (can be Turing complete)

   \[
   f_{\text{best}} = \arg\min_{f \in \text{DSL}} \text{cost}(D, f)
   \]

3. **Synthesize** \( f_{\text{best}} \in \text{DSL} \) from Dataset D:

4. **Use** \( f_{\text{best}} \) on new structures:

   \[
   f_{\text{best}} \;\rightarrow\; y
   \]

---

### PRACTICAL IMPACT

**SLANG**
- Deep3

**Deep3**
- De-mind equity javascript incl. args, vars & #deep
- Syntactic and semantic analysis, Type inference
- 20 Best Practices for Web Designers of Year 2014
- JavaScript, Type Inference
- NICE 2 Predict

**NICE 2 Predict**
- One of the most powerful tools for static code analysis and reverse engineering
- Slang
- Deep3

---
Course objectives

Introduction to the emerging area of learning from Big Code.

Learn how to read and evaluate papers in the area

Learn how to make good technical presentations
Basic Information

• Instructors
  – Prof. Dr. Martin Vechev (martin.vechev@inf.ethz.ch)
  – Dr. Veselin Raychev (veselin.raychev@inf.ethz.ch)

• Meetings
  – once a week, 2 presentations per meeting
  – next meeting: March 5
  – Presentation schedule will be posted on course
How it works: your tasks

- select a paper, get date
- study paper
- create presentation
- meet advisor, get feedback

By this Friday, send e-mail to Veselin (veselin.raychev@inf.ethz.ch) with your 5 choices

give final presentation: 30 min
answer questions: 15 min

Participation:
ask good questions, attend all classes
Study paper

select a paper, get date

study paper

create presentation

meet advisor, get feedback

3 ‘C’s of reading:

Carefully: lookup unknown terms, read cited papers

Critically: find limitations, flaws

Creatively: think of improvements

Write: key ideas, try examples by hand

Consult with TA / Instructors, if questions email also fine

give final presentation: 30 min
answer questions: 15 min

Participation:
ask good questions, attend all classes
Create presentation

- select a paper, get date
- study paper
- create presentation
- meet advisor, get feedback

- Explain motivation for the work
- Clearly present the technical solution and results
- Use your own example, not the one in the paper
- Outline limitations / improvements
- Focus on the key/crucial concepts
- Do not present all of the details

- give final presentation: 30 min
- answer questions: 15 min

Participation:
- ask good questions, attend all classes
Meet advisor, get feedback

- select a paper, get date
- study paper
- create presentation
- meet advisor, get feedback
  - clear technical questions on the paper
  - get feedback on draft presentation
  - meeting mandatory a week or so before presentation

- give final presentation: 30 min
- answer questions: 15 min

Participation:
- ask good questions, attend all classes
Grading

- select a paper, get date
- study paper
- create presentation
- meet advisor, get feedback

Quality of your final presentation:
- how well you understood the material?
- how well you presented it?
- how well you answered the questions?

We will take into account paper difficulty.

Participation:
- did you ask good questions?
- attendance will be taken (do not miss classes)

Give final presentation: 30 min
Answer questions: 15 min

Participation:
- ask good questions, attend all classes
Useful Presentation Links

“How to give strong technical presentations”, by Markus Püschel


– Please read the above slides!

“Even a geek can speak”, by Joey Asher