StiCProb: A Novel Feature Mining Approach using Conditional Probability

Yutian Tang, and Dr. Hareton Leung
The Hong Kong Polytechnic University, Hong Kong
@SANER 2017
more: www.chrisyttang.org/loong
Motivation

- legacy –> product line

1. How to locate the feature?
2. How to measure?

\[\text{feature} \leftrightarrow \text{element}\]
How to locate?

1. Select seeds
2. Annotate features
Basic

Programming elements

Relationship

Feature

Annotation

\[ E \]

\[ R \subseteq E \times E \]

\[ F \]

\[ A \subseteq E \times F \]
Basic (cond’)

mutual exclusion

\[ M \subseteq F \times F \]

implications

\[ \Rightarrow \subseteq F \times F \]

full annotation

\[ A \subseteq E \times F \]

\[ A^* = \left\{ (e, f) \mid (e, f) \in A, g \Rightarrow^* f \right\} \]

\[ (e, f) \]

\[ (e, f) \mid (e, f) \in A, g \Rightarrow f \]
Annotate Features

Annotation State

\[ S(A^*, f, i) = \{ e \mid (e, f) \in A^* \} \]

mining process for a single feature
Feature-Element Correlation Coefficient

$$S(A^*, f, i)$$

Annotation State (i)

$$S(A^*, f, i + 1)$$

Annotation State (i+1)

Prog. Elements Cand.
Feature-Element Correlation Coefficient (cond’)

Prog. Elements Cand.

feature $f$

$S(A^*, f, i) \quad \Rightarrow \quad p(e | S(A^*, f, i)) \Rightarrow S(A^*, f, i + 1)$
Feature-Element Correlation Coefficient (cond')

\[ t \in S(A^*, f, i) \]
Feature-Element Correlation Coefficient (cond')

**Program P**

1. \( s = 0 \)
2. \( i = 0 \)
3. while \((i < 5)\)
4. \( if \ (i < 3) \)
5. \( t = 1 \)
6. \( else \ t = i - 2 \)
7. \( s = s + t \)
8. \( i = i + 1 \)
9. return \( s \)

**Slicing Scope**

\[
sscope(e) = e \cup \left\{ s \mid s \xrightarrow{df} e, s \in E \right\}
\]
Feature-Element Correlation Coefficient (cond')

Binding

\[ bind(e) = \text{def}(e) \cup \text{use}(e)^* \quad \text{in} \quad \text{scope}(e) \]

Context Binding

\[ \text{contbind}(e, [c]) \]

* All def and use within e
Context Binding

Context Binding @ Method Invocation

\[ l = r_0 . m(r_1, r_2, ..., r_n) \]

\[ \text{contbind}(m,[r_1,\ldots,r_n]) = \text{dispatch}(p_i = r_i) \rightarrow \text{bind}(m) \]

Context Binding

Context Binding @ Method Invocation

main(){
    A a = new A();
csite 1: z = wrapper(a);
}

wrapper (A b){
    y = bar(b); return y;
}

bar (A c){
    x = c.f; return x;
}

Context Binding

Context Binding @ Overriding

Animal
:eat(food: Food)

Cow
:eat(food: Grass)

1. def and use in m
2. def in parent class/interface, used in m
Context Binding

Context Binding @ Overriding

\[
\text{def}(m) \cup \text{use}(m)
\]

\[
\text{def}(m) \quad \text{use}(m) \\
\text{def}(p(1)) \rightarrow \text{use}(m) \\
\text{def}(p(2)) \rightarrow \text{use}(m) \\
\vdots \\
\text{def}(p(n)) \rightarrow \text{use}(m)
\]

\text{def}(c) - \text{def}(m)
Context Binding

Context Binding @ Overriding

defined in m: \( \text{def}(m) \)

used in m: \( \text{use}(m) \)

1. used in m and defined in m
2. used in m and not defined in m

\[
\text{contbind}(m,[p_1,\ldots,p_n]) = \text{def}(m) \cup \bigcup_{i=1}^{n} (\text{use}(m) \leadsto \text{def}(p_i))
\]

\( \leadsto \) used to specify the source of the context
Context Binding

Context Binding @ Overriding: Example

```
1 public class FlyingCar implements OperateCar {
2     public int startEngine(int encryptedValue) {
3         OperateCar.super.startEngine(OperateCar.encryptedValue);
4     }
5 }
6
public interface OperateCar {
7     int encryptedValue = 1;
8     default public int startEngine(int value) {...}
9 }
```

c\text{contextbind}(\text{startEngine}) = \{\text{encryptedValue}, \text{OperateCar.encryptedValue}\}
Context Binding

Context Binding @ Inheritance

\[ \text{contbind}(c, [p_1, \ldots, p_n]) = \text{bind}(c) \cup \bigcup_{i=1}^{n} \text{def}(p_i) \]
Feature-Element Correlation Coefficient (cond')

Context Binding

\[ \text{contbind}(e, [c]) \]

by default:  \textit{context-aware points-to analysis}
Feature-Element Correlation Coefficient (cond’)

\[ S(A^*, f, i) = \{ e \mid (e, f) \in A^* \} \]

\[ S(A^*, f, i) = \bigcup_{a \in S(A^*, f, i)} \text{contextbind}(a) \]
StiCProb

1. Build Program DB

2. Build uniqueness table

3. Annotate features
StiCProb: Uniqueness Table

Element $s$ and $t$ with a relation $r$

$$s \xrightarrow{r} t$$

$$U \left( E,T,R,P_{\text{forward}},P_{\text{backward}} \right)$$

$P_{\text{forward}}$

The uniqueness of $t$ to $s$ if $s$ has been annotated to a feature $f$

$$s \in S(A^*, f, i)$$

$$S(A^*, f, i) = \{ e | (e, f) \in A^* \}$$
StiCProb: Uniquess Table(\text{cond}')

\[ P_{\text{forward}} \]

the uniqueness of \texttt{t to s} if s has been annotated to a feature f

\[ s \in S(A^*, f, i) \]
\[ S(A^*, f, i) = \{ e | (e, f) \in A^* \} \]

\[ P_{\text{forward}} \left( s \xrightarrow{r} t \middle| (s, f) \in A^* \right) = \frac{\text{contbind}(t, [s])}{\text{contbind}(s)} \]
StiCProb: Uniquess
Table\textsuperscript{(cond’)}

\( P_{\text{backward}} \)

the uniqueness of \textit{s} to \textit{t} if \textit{t} has been annotated to a feature \textit{f}

\[
P_{\text{backward}}(s \xrightarrow{r} t \mid (t, f) \in A^*) = \frac{\bigcup i \xrightarrow{r} t \text{ contbind}(t, [i])}{\bigcup i \xrightarrow{r} t \text{ contbind}(t, [s])}
\]

\[
\bigcup i \xrightarrow{r} t \text{ contbind}(t, [i])
\]

\( s \in S(A^*, f, i) \)

\( S(A^*, f, i) = \{e \mid (e, f) \in A^*\} \)

\( \bigcup i \xrightarrow{r} t \text{ contbind}(t, [i]) \)

a collection of context binding from all prog. elements, which have relation \( r \) with \textit{t}.
Algorithm 1: StiCProb feature mining approach

**Input:** seeds, fm, threshold, U  
**Output:** all annotation states for features Sset in fm

1. Create a set of annotation states as  
   \[Sset = \bigcup_{f \in \text{features}} S(A^*, f)\];
2. Assign seeds to each feature as \(S(A^*, f) = \text{seeds}(f)\);
3. Create feature set features with all features in fm;
4. While features not NULL do
   a. For feature f in features do
      i. Create set waitList = \(\emptyset\);
      ii. Create candidate set \(C(S, f) = \emptyset\) for f;
      iii. Add all elements have relations with elements in \(S(A^*, f)\) to \(C(S, f)\);
      iv. For element m in \(C(S, f)\) do
         a. If there is a relation \(r\) from m to the element e in \(S(A^*, f)\) then
            i. Let value = \(p_{\text{backward}}(m \xrightarrow{r} e)\) \(\mid (e, f) \in A^*\);
         else
            i. Let value = \(p_{\text{forward}}(e \xrightarrow{r} m)\) \(\mid (e, f) \in A^*\);
         if value > threshold then
            i. Add m to waitList;
      Update \(S(A^*, f) \leftarrow S(A^*, f) \cup \text{waitList}\);
   if StopCheck(f) is TRUE then
      i. Remove f from features;
5. Return Sset;
## Case Study

<table>
<thead>
<tr>
<th>Projects</th>
<th>LOC</th>
<th>#features</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prevalyer</td>
<td>8,009</td>
<td>5</td>
<td>object persistence library</td>
</tr>
<tr>
<td>MobileMedia</td>
<td>4,653</td>
<td>6</td>
<td>mobile</td>
</tr>
<tr>
<td>Lampiro</td>
<td>44,584</td>
<td>*2</td>
<td>message client</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>~120K</td>
<td>7</td>
<td>modeling tool</td>
</tr>
</tbody>
</table>
Case Study

Experimental Setting:

- **seeds**: FLAT3 tool
- **tool**: Loong Eclipse plugin
- **feature model**: benchmark
- **benchmark**

Related Approaches:

- Type system, Topology analysis, Text comparison

Measurement:

- precision recall f-score
Case Study

Other Settings:

# seeds: 3

threshold: 0.6
Experimental Result

StiCProb with threshold $t = 0.6$

<table>
<thead>
<tr>
<th>Project</th>
<th>Feature</th>
<th>Feature Size</th>
<th>Mining Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LOC</td>
<td>FR</td>
</tr>
<tr>
<td>Prevayler</td>
<td>Censor</td>
<td>105</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Gzip</td>
<td>165</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Monitor</td>
<td>240</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Replication</td>
<td>1487</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>Snapshot</td>
<td>263</td>
<td>29</td>
</tr>
<tr>
<td>MobileM.</td>
<td>CopyMedia</td>
<td>79</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Sorting</td>
<td>85</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Favorites</td>
<td>63</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>SMS Trans.</td>
<td>714</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Music</td>
<td>709</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>Photo</td>
<td>493</td>
<td>35</td>
</tr>
</tbody>
</table>

**LOC:** line of code, **FR:** count of distinct code fragments, **IT:** number of iteration, **Prec.:** precision
# Experimental Result (cond’)

StiCProb with threshold \( t = 0.6 \)

<table>
<thead>
<tr>
<th>Project</th>
<th>Feature</th>
<th>Feature Size</th>
<th>Mining Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LOC</td>
<td>FR</td>
</tr>
<tr>
<td>MobileM.</td>
<td>M.Transfer</td>
<td>153</td>
<td>4</td>
</tr>
<tr>
<td>Lampiro</td>
<td>Compre.</td>
<td>5155</td>
<td>33</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>Cognitive</td>
<td>16319</td>
<td>285</td>
</tr>
<tr>
<td></td>
<td>Activity</td>
<td>2282</td>
<td>115</td>
</tr>
<tr>
<td></td>
<td>State</td>
<td>3917</td>
<td>115</td>
</tr>
<tr>
<td></td>
<td>Collab.</td>
<td>1579</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Sequence</td>
<td>5379</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Use-Case</td>
<td>2712</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>Deployment</td>
<td>3147</td>
<td>57</td>
</tr>
</tbody>
</table>

**LOC**: line of code, **FR**: count of distinct code fragments, **IT**: number of iteration, **Prec.**: precision.
Experimental Result (cond’)

Recall Performance
SP: StiCProb (t = 0.6)
TP: topology analysis

Precision Performance
TS: type system
TC: text comparison
Experimental Result

*f-score*

SP: StiCProb ($t = 0.6$)
TP: topology analysis
TS: type system
TC: text comparison
**Experimental Result**

**Runtime**

- **SP**: StiCProb ($t = 0.6$)
- **TP**: topology analysis
- **TS**: type system
- **TC**: text comparison

---

**Graphs**

- **Prevalyer**: SP: 2, TS: 1, TP: 2, TC: 71
- **MobileMeida**: SP: 2, TS: 2, TP: 3, TC: 21
- **Lampiro**: SP: 1, TS: 1, TP: 13, TC: 135
- **ArgoUML**: SP: 2, TS: 1, TP: 254, TC: 5,415

---

**Legend**

- **SP**: StiCProb
- **TS**: type system
- **TP**: topology analysis
- **TC**: text comparison
Discussion

Seeds:

1. seeds provided by FLAT3 might be not correct

2. number of seeds

3. granularity of seeds: coarse granularity could improve the recall performance, but sometimes at the cost of precision.
Discussion

Thresholds:

threshold: 0.6 —-> 0.8

precision: 83% —-> 85%

The threshold contributes less to the performance.
Thanks
Loong Plugin

- Download: http://www.chrisyttang.org/loong/
- Source code: https://github.com/csytang/Loong
- Experimental results: https://drive.google.com/folderview?id=0B9l0qvk6pnW0ZDRYMmxI/QhRb0U&usp=sharing
- Online Tutorial: http://www.chrisyttang.org/loong/
Discussion

• Need of req. specification $\leftrightarrow$ seeds selection/poor naming

• Variants of our approach? or better solutions?

• - weighted graph $\rightarrow$ graph clustering

• ?