On Sufficiency of Mutants

Akbar Siami Namin and James H. Andrews
Department of Computer Science
The University of Western Ontario, Canada.

Doctoral Symposium ICSE 2007 - Minneapolis, USA

May 21, 2007
Motivation
Challenge
Problem Statement
Possible Solution
Our Approach
Preliminary Result
Conclusion
Future Work
It is about the cost
Current testing techniques are still expensive

We need more effective and less expensive testing techniques

How to assess the efficiency and effectiveness of the new techniques?

- Efficiency: relatively easy to assess by defining some metrics (e.g., time, cost, used resources, etc.)
- Effectiveness: Not easy to address!!!
Experimental study is a tool to assess and compare the effectiveness of different techniques.

Experimental study needs prepared subjects.

Preparing subjects in software testing needs:

1. A program
2. Some known faulty versions
3. A relatively large number of test cases
Software and Experiments
Motivation

Problem: In software testing research there are only a few well prepared subjects

- Finding programs with known real faults is difficult

- Not enough data: making the experiments and statistical analysis less significant

- Is there any solution?
Using mutation to generate faulty versions.

What is mutation and mutation operators:

**Fault** - An incorrect part of a code.
- Faulty Code: `while (a > b) do {...}`
- Correct Code: `while (a < b) do {...}`

**Mutant** - Small modification of a code by mutation operator.
- Correct Code: `while (a > b) do {...}`
- Mutated Code: `while (a < b) do {...}`
So, what is the problem?
The Problem

- Many mutation operators (Good or Bad?)
  - Proteum, A mutant generator for C, implements 108 operators
  
- Good: Simulating defects more closely

- Bad: Enormous number of mutants
  
  - E.g., A simple program with 137 lines of code can have 4935 mutants

- Infeasible Computation!!!
Number of Mutants vs. Net Lines of Code

Number of Mutants

Net Lines of Code

tcas
totinfo
printtokens
printtokens2
schedule2
schedule
replace
Is there any solution?
Finding a minimal subset of mutation operators (problem?)

◆ The minimal subset might not be necessarily the best

Finding a sufficient subset of mutants called sufficient mutants (How?)

Offutt et al. worked on selecting one of several fixed subsets of 21 mutation operators

Wong et al. suggested defining constrained mutation so that the most critical and useful operators will be used in experiments
Statistical Techniques

Outline
Motivation
Challenge
Problem Statement
Possible Solution
Our Approach
Our Approach Metrics
Correlation Analysis
Cluster Analysis
All-Subset Regression
Preliminary Result
Conclusion
Future Work
Our Approach

1. Finding sufficient subset through statistical techniques
   • Correlation Analysis
   • Cluster Analysis
   • All-Subset Regression Analysis

2. Applying multiple-linear regression on the identified enough subset
Cost of each operator $\mu_i$:

$$C_i = \sum_j \frac{\#\text{Mutants Generated By } \mu_i}{\#\text{All Mutants}}$$

for $j \in \{\text{Subject Programs}\}$

For each test suite:

1. $Am_i$: Detection ratio of the mutants generated by the operator $\mu_i$
2. $AM$: Detection ratio of the mutants generated by all operators
Correlation Analysis: A useful technique

- Correlation between two variables (How?)
  1. Constructing correlation matrix
  2. Comparing each pair of variables
  3. Eliminating one with high cost if there is a high correlation
     - What is high correlation? $\text{cor}(X, Y) \geq 0.90$ (Standard Guilford Scalling)

- Calling the remaining variables the sufficient set

- Applying multiple-linear regression on the sufficient set
Cluster Analysis: A similarity technique

- Clustering variables based on their similarities (How?)
  1. Constructing proximity matrix
     - Computing the nearest neighbor
  2. Generating a dendrogram
  3. Eliminating one with high cost if there is a high correlation
     - What is high correlation? $cor(X, Y) \geq 0.90$

- Calling the remaining variables the sufficient set

- Applying multiple-linear regression on the sufficient set
Select a small subset of a large set of variables that model a response variable (How?)

1. Constructing all regression models exhaustively
   - Number of variables involved in each model chosen by the analyst

2. The result is already a multiple-linear regression

Calling the variables involved in the chosen model the sufficient set
Preliminary Result
Preliminary Result

- Applied the techniques to tcas.c program so far
- 300 test suites for each group
- Applied only to random-based test suites
## Preliminary Result

<table>
<thead>
<tr>
<th>Mutants</th>
<th>core-SUB</th>
<th>core-EBC</th>
<th>core-CA</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-DirVarAriNeg</td>
<td>44</td>
<td>I-IndVarAriNeg</td>
<td>19</td>
</tr>
<tr>
<td>I-DirVarRepReq</td>
<td>220</td>
<td>I-IndVarLogNeg</td>
<td>19</td>
</tr>
<tr>
<td>I-IndVarLogNeg</td>
<td>19</td>
<td>II-ArgRepReq</td>
<td>5</td>
</tr>
<tr>
<td>I-IndVarRepReq</td>
<td>91</td>
<td>u-OALN</td>
<td>2</td>
</tr>
<tr>
<td>u-OABN</td>
<td>3</td>
<td>u-OASN</td>
<td>2</td>
</tr>
<tr>
<td>u-OLSN</td>
<td>34</td>
<td>u-OCNG</td>
<td>8</td>
</tr>
<tr>
<td>u-VDTR</td>
<td>111</td>
<td>u-OULLN</td>
<td>17</td>
</tr>
<tr>
<td>u-VGSR</td>
<td>794</td>
<td></td>
<td></td>
</tr>
<tr>
<td>u-VTWD</td>
<td>74</td>
<td></td>
<td></td>
</tr>
<tr>
<td>II-ArgLogNeg</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>1393</td>
<td>Total</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Total</td>
</tr>
</tbody>
</table>
The Model for Cluster Analysis

Outline
Motivation
Challenge
Problem Statement
Possible Solution
Our Approach
Preliminary Result

Conclusion
Future Work
Conclusion
Addressing the expensive cost of mutation by three statistical analysis

- Correlation Analysis
- Cluster Analysis
- All-Subset Regression Analysis

Which one is better?

- In terms of efficiency: Cluster Analysis
- In terms of accuracy: All-Subset Regression Analysis
Future Work
Future Work

- Applying techniques to all Siemens subject programs
- Verifying the result with other subject programs
- Applying the techniques to different programming languages using different mutant generators