Exploiting Redundant Test Cases in Fault Localisation: Good or Bad?

Alexandre Perez
alexandre.perez@fe.up.pt

Nuno Cardoso, José Campos, Rui Abreu
nunopcardoso@gmail.com, jose.carlos.campos@fe.up.pt,
ruic@computer.com

Department of Informatics Engineering
Faculty of Engineering, University of Porto

November 19, 2013
1 Spectrum-based Reasoning

2 Redundancy at the Test Level: Impacts?

3 Minimising Coincidental Correctness

4 Conclusions
A hit spectra is a pair \((ob\!s, e)\):

- \(ob\!s_i\) Activity of components in transaction \(i\).
- \(e_i\) Outcome of transaction \(i\) (pass or fail).

<table>
<thead>
<tr>
<th>(i)</th>
<th>(ob!s_i)</th>
<th>(e_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1 0 1 1</td>
<td>1</td>
</tr>
</tbody>
</table>
A hit spectra is a pair \((\text{obs}, e)\):

- \(\text{obs}_i\) Activity of components in transaction \(i\).
- \(e_i\) Outcome of transaction \(i\) (pass or fail).

### Spectrum-based Reasoning:

- **Different** from statistical fault localisation approaches.
- **Generate** sets of components that would explain the observed erroneous behaviour.
- **Rank** the candidates according to their likelihood of being faulty.
• **Generate** sets of components that would explain the observed erroneous behaviour.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$obs_i$</th>
<th>$e_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1 0 1 1</td>
<td>1</td>
</tr>
</tbody>
</table>

1 Rui Abreu and Arjan J. C. van Gemund. “A Low-Cost Approximate Minimal Hitting Set Algorithm and its Application to Model-Based Diagnosis”. In: SARA. 2009.

• **Generate** sets of components that would explain the observed erroneous behaviour.

\[
\begin{array}{cccc}
  i & obs_i & e_i \\
  1 & 1 & 1 & 0 & 1 \\
  2 & 1 & 0 & 1 & 1 \\
\end{array}
\]

- A **minimal candidate** is a set of components that cover all failing transactions.
- **STACCATO**\(^1\) & **MHS2**\(^2\).

---

\(^1\)Rui Abreu and Arjan J. C. van Gemund. “A Low-Cost Approximate Minimal Hitting Set Algorithm and its Application to Model-Based Diagnosis”. In: *SARA*. 2009.

Barinel\(^3\) approach:

- For each candidate \(d\) under a set of observations \((\text{obs}, e)\), the posterior probability is calculated using Naïve Bayes rule\(^4\).

\[
\Pr(d \mid \text{obs}) = \Pr(d) \times \frac{\Pr(\text{obs} \mid d)}{\Pr(\text{obs})}
\]

---

\(^3\) Rui Abreu, Peter Zoeteweij, and Arjan J. C. van Gemund. “Spectrum-Based Multiple Fault Localization”. In: ASE. 2009, pp. 88–99.

\(^4\) Conditional independence is assumed throughout the process.
Barinel$^3$ approach:

- For each candidate $d$ under a set of observations $(obs, e)$, the posterior probability is calculated using Naïve Bayes rule$^4$.

- $Pr(d)$ is used to make larger candidates less probable.

\[
Pr(d \mid obs) = Pr(d) \times \frac{Pr(obs \mid d)}{Pr(obs)}
\]


$^4$Conditional independence is assumed throughout the process.
**BARINEL**\(^3\) approach:

- For each candidate \(d\) under a set of observations \((obs, e)\), the posterior probability is calculated using Naïve Bayes rule\(^4\).

- \(Pr(d)\) is used to make larger candidates less probable.

\[
Pr(d \mid obs) = \frac{Pr(d) \times Pr(obs \mid d)}{Pr(obs)}
\]

- \(Pr(obs)\) is not considered for ranking purposes (does not depend on \(d\)).

---

\(^3\)Rui Abreu, Peter Zoeteweij, and Arjan J. C. van Gemund. “Spectrum-Based Multiple Fault Localization”. In: ASE. 2009, pp. 88–99.

\(^4\)Conditional independence is assumed throughout the process.
Barinel\(^3\) approach:

- For each candidate \(d\) under a set of observations \((obs, e)\), the posterior probability is calculated using Naïve Bayes rule\(^4\).

\[
\Pr(d \mid obs) = \frac{\Pr(obs \mid d) \times \Pr(d)}{\Pr(obs)}
\]

- \(\Pr(d)\) is used to make larger candidates less probable.

- \(\Pr(obs)\) is not considered for ranking purposes (does not depend on \(d\)).

- \(\Pr(obs \mid d)\) is used to bias the probability based on the run-time observations.

\(^3\)Rui Abreu, Peter Zoeteweij, and Arjan J. C. van Gemund. “Spectrum-Based Multiple Fault Localization”. In: ASE. 2009, pp. 88–99.

\(^4\)Conditional independence is assumed throughout the process.
Diagnostic Candidate Ranking

\[
\Pr(\text{obs}|d) = \prod_{\text{obs}_i \in \text{obs}} \begin{cases} 
G(\text{obs}_i, d) & \text{if } e_i = 0 \\
1 - G(\text{obs}_i, d) & \text{if } e_i = 1
\end{cases}
\]

\(G(\text{obs}_i, d)\) is estimated:

- Using maximum likelihood estimation under parameters \(\{g_j|j \in d\}\)^5.
- NFGE^6: uses a feedback loop to update the health estimates of each component.

---


1. Spectrum-based Reasoning

2. Redundancy at the Test Level: Impacts?

3. Minimising Coincidental Correctness

4. Conclusions
Redundant Test Cases

At the spectra level of abstraction:

- Tests are redundant if they share similar activity patterns.
- Can exonerate faulty components.

---


8 George K. Baah, Andy Podgurski, and Mary Jean Harrold. “Mitigating the confounding effects of program dependences for effective fault localization”. In: FSE. 2011, pp. 146–156.
Redundant Test Cases

At the spectra level of abstraction:

- Tests are redundant if they share similar activity patterns.
- Can exonerate faulty components.

Coincidental correctness\(^7,8\):

- Occurs when passing test cases execute faulty components and no failure is triggered.
- Can be caused by incorrect or relaxed test oracles.
- Can occur due to the abstraction of program traces used.

---


\(^8\) George K. Baah, Andy Podgurski, and Mary Jean Harrold. “Mitigating the confounding effects of program dependences for effective fault localization”. In: FSE. 2011, pp. 146–156.
Consider the following hit-spectra matrix:

<table>
<thead>
<tr>
<th></th>
<th>( obs_i )</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 0</td>
<td>e_i</td>
</tr>
<tr>
<td>2</td>
<td>0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0</td>
<td>0</td>
</tr>
</tbody>
</table>

- After candidate generation: \( D = \langle \{1, 2\}, \{1, 3\}, \{2, 3\} \rangle \)
- Diagnostic Ranking:
  - \( Pr(\{2, 3\}|obs) = 0.66 \)
  - \( Pr(\{1, 2\}|obs) = 0.17 \)
  - \( Pr(\{1, 3\}|obs) = 0.17 \)
After a redundant test case:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\text{obs}_i$</th>
<th>$e_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0</td>
<td>0</td>
</tr>
</tbody>
</table>

- $\Pr(\{2, 3\}|\text{obs}) = 0.66$
- $\Pr(\{1, 2\}|\text{obs}) = 0.17$
- $\Pr(\{1, 3\}|\text{obs}) = 0.17$
### Redundant Test Cases – Example

After a redundant test case:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$obs_i$</th>
<th>$e_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$i$</th>
<th>$obs_i$</th>
<th>$e_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1 0 1</td>
<td>0</td>
</tr>
</tbody>
</table>

- $\Pr(\{2, 3\}|obs) = 0.66$
- $\Pr(\{1, 2\}|obs) = 0.17$
- $\Pr(\{1, 3\}|obs) = 0.17$
- $\Pr(\{2, 3\}|obs) = 0.59$
- $\Pr(\{1, 2\}|obs) = 0.35$
- $\Pr(\{1, 3\}|obs) = 0.06$
Redundant Test Cases – Example

After a redundant test case:

<table>
<thead>
<tr>
<th></th>
<th>$obs_i$</th>
<th>$e_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- $\Pr(\{2, 3\}|obs) = 0.66$
- $\Pr(\{1, 2\}|obs) = 0.17$
- $\Pr(\{1, 3\}|obs) = 0.17$
Redundant Test Cases – Example

After a redundant test case:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$obs_i$</th>
<th>$e_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0</td>
<td>0</td>
</tr>
</tbody>
</table>

$\Pr(\{2, 3\}|obs) = 0.66$

$\Pr(\{1, 2\}|obs) = 0.17$

$\Pr(\{1, 3\}|obs) = 0.17$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$obs_i$</th>
<th>$e_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0 1 1</td>
<td>0</td>
</tr>
</tbody>
</table>

$\Pr(\{1, 2\}|obs) = 0.33$

$\Pr(\{1, 3\}|obs) = 0.33$

$\Pr(\{2, 3\}|obs) = 0.33$
Table of Contents

1. Spectrum-based Reasoning
2. Redundancy at the Test Level: Impacts?
3. Minimising Coincidental Correctness
4. Conclusions
Marsi et al.⁹ remove coincidentally correct test cases by:

- Selecting a set of suspicious statements executed by all failing tests (called CCEs);
- Clustering tests into two groups based on the similarity of the executed statements to the CCEs.

---

Miao et al.\textsuperscript{10} use a similar clustering approach:

- Uses hard \textit{k-Means} clustering with $k = |T| \times p$.
- If a passing test is in the same cluster as a failing one, it is labeled as coincidentally correct.

Minimising Coincidental Correctness – Related work

*Miao et al.*\(^{10}\) use a similar clustering approach:

- Uses hard *k*-Means clustering with \( k = |T| \times p \).
- If a passing test is in the same cluster as a failing one, it is labeled as coincidentally correct.

Two strategies:

- **Cleaning Strategy**: Coincidental test cases are removed from the original test suite.
- **Relabelling Strategy**: The outcome of coincidental test \( i \) is changed to failing \( (e_i = 1) \).

---

k-Means: data elements are clustered into k distinct clusters.
Fuzzy c-Means: membership values represent the strength of the association between a data element and a cluster.
Work in progress.

Introduces the concept of **assertion confidence**:

- No longer assuming that all assertions are equally trustworthy.
- **Fuzzy memberships** of coincidentally correct tests can represent confidence.

\[
Pr(obs_i, c_i|d) = (1 - c_i) + (c_i \cdot Pr(obs_i|d))
\]
Work in progress.

Introduces the concept of assertion confidence:
- No longer assuming that all assertions are equally trustworthy.
- Fuzzy memberships of coincidentally correct tests can represent confidence.

\[ \Pr(\text{obs}_i, c_i|d) = (1 - c_i) + (c_i \cdot \Pr(\text{obs}_i|d)) \]

Example:

<table>
<thead>
<tr>
<th>(i)</th>
<th>1</th>
<th>2</th>
<th>(e_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ \Pr\{1\}|\text{obs} = \]
\[ \Pr\{2\}|\text{obs} = \]
Fuzzinel Approach

Work in progress.

Introduces the concept of assertion confidence:
- No longer assuming that all assertions are equally trustworthy.
- Fuzzy memberships of coincidentally correct tests can represent confidence.

\[
Pr(o_{bs_i}, c_i|d) = (1 - c_i) + (c_i \cdot Pr(o_{bs_i}|d))
\]

Example:

| i | e_i | c_i | Pr\{1\}|obs | Pr\{2\}|obs |
|---|-----|-----|-----------|--------|-----------|
| 1 | 1   | 1   | (1 - c_i) + (c_i \cdot Pr(o_{bs_i}|d)) | 5.0 \times 10^{-4} |
| 2 | 0   | 0.5 | Pr\{1\}|obs | 2.5 \times 10^{-4} |
| 3 | 0   | 1   | Pr\{2\}|obs |            |
1 Spectrum-based Reasoning

2 Redundancy at the Test Level: Impacts?

3 Minimising Coincidental Correctness

4 Conclusions
Exploiting Redundant Test Cases in Fault Localisation: Good or Bad?

At the hit-spectra level of abstraction:

- Coincidental correctness from redundant test cases has a potential negative effect on accuracy.
- The fault is exercised without triggering the failure, exonerating potentially faulty components.
- Negative effects on fault localisation can however be minimised.
Exploiting Redundant Test Cases in Fault Localisation: Good or Bad?

At the hit-spectra level of abstraction:

- Coincidental correctness from redundant test cases has a potential negative effect on accuracy.
- The fault is exercised without triggering the failure, exonerating potentially faulty components.
- Negative effects on fault localisation can however be minimised.

Introduced Fuzzinel:

- Does not remove nor relabel the input.
- Changes the confidence we have in certain tests.

Future challenges:

- How to better estimate the number of centroids in our fuzzy clustering step?
Exploiting Redundant Test Cases in Fault Localisation: Good or Bad?

Alexandre Perez  
alexandre.perez@fe.up.pt

Nuno Cardoso, José Campos, Rui Abreu  
nunopcardoso@gmail.com, jose.carlos.campos@fe.up.pt, rui@computer.com

Department of Informatics Engineering  
Faculty of Engineering, University of Porto

November 19, 2013