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

Table of Contents

A hit spectra is a pair (obs, e):

- obs_i Activity of components in transaction i.
- e_i Outcome of transaction i (pass or fail).

i	obs_i			e_i
	1	2	3	
1	1	1	0	1
2	1	0	1	1

A hit spectra is a pair (obs, e):

obs_i Activity of components in transaction i.

e_i Outcome of transaction i (pass or fail).

i	obs_i			e_i
	1	2	3	
1	1	1	0	1
2	1	0	1	1

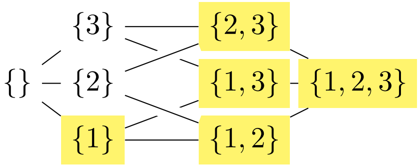
Spectrum-based Reasoning:

- 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.

Diagnostic Candidate Generation

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

i	obs_i			e_i
	1	2	3	
1	1	1	0	1
2	1	0	1	1



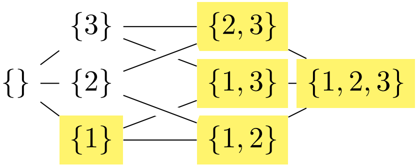
[1] Rui Abreu and Arjan J. C. van Gemund. "A Low-Cost Approximate Minimal Hitting Set Algorithm and its Application to Model-Based Diagnosis". In: *SARA*. 2009.

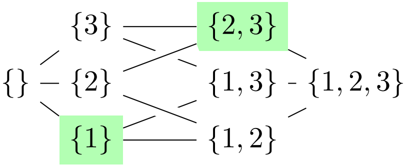
[2] Nuno Cardoso and Rui Abreu. "MHS2: A Map-Reduce Heuristic-Driven Minimal Hitting Set Search Algorithm". In: *MUSEPAT*. 2013, pp. 25–36.

Diagnostic Candidate Generation

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

i	obs_i			e_i
	1	2	3	
1	1	1	0	1
2	1	0	1	1





- 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.

[2] Nuno Cardoso and Rui Abreu. "MHS2: A Map-Reduce Heuristic-Driven Minimal Hitting Set Search Algorithm". In: *MUSEPAT*. 2013, pp. 25–36.

BARINEL[3] approach:

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

$$\mathsf{Pr}(d \mid obs) = \mathsf{Pr}(d) \times \frac{\mathsf{Pr}(obs \mid d)}{\mathsf{Pr}(obs)}$$

[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].

- $\text{Pr}(d)$ is used to make larger candidates less probable.

$$\text{Pr}(d \mid obs) = \boxed{\text{Pr}(d)} \times \frac{\text{Pr}(obs \mid d)}{\text{Pr}(obs)}$$

[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.

Diagnostic Candidate Ranking

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.

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

- $\Pr(obs)$ 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].
- $\text{Pr}(d)$ is used to make larger candidates less probable.

$$\text{Pr}(d \mid obs) = \text{Pr}(d) \times \frac{\boxed{\text{Pr}(obs \mid d)}}{\text{Pr}(obs)}$$

- $\text{Pr}(obs)$ is not considered for ranking purposes (does not depend on d).
- $\text{Pr}(obs \mid d)$ 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.

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

$G(obs_i, d)$ 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.

[5] Rui Abreu, Peter Zoeteweij, and Arjan J. C. van Gemund. "Spectrum-Based Multiple Fault Localization". In: *ASE*. 2009, pp. 88–99.

[6] Nuno Cardoso and Rui Abreu. "A Kernel Density Estimate-Based Approach to Component Goodness Modeling". In: *AAAI*. 2013, pp. 152–158.

Table of Contents

At the spectra level of abstraction:

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

[7] Wes Masri and Rawad Abou Assi. "Cleansing Test Suites from Coincidental Correctness to Enhance Fault-Localization". In: *ICST*. 2010, pp. 165–174.

[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.

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.

[7] Wes Masri and Rawad Abou Assi. "Cleansing Test Suites from Coincidental Correctness to Enhance Fault-Localization". In: *ICST*. 2010, pp. 165–174.

[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 – Example

Consider the following hit-spectra matrix:

i	obs_i			e_i
	1	2	3	
1	1	1	0	1
2	0	1	1	1
3	1	0	1	1
4	1	0	0	0

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

Redundant Test Cases – Example

After a redundant test case:

i	obs_i 1	2	3	e_i
1	1	1	0	1
2	0	1	1	1
3	1	0	1	1
4	1	0	0	0

- $\mathsf{Pr}(\{2,3\}|obs) = 0.66$
- $\mathsf{Pr}(\{1,2\}|obs) = 0.17$
- $\mathsf{Pr}(\{1,3\}|obs) = 0.17$

After a redundant test case:

i	obs_i			e_i
	1	2	3	
1	1	1	0	1
2	0	1	1	1
3	1	0	1	1
4	1	0	0	0

- $\text{Pr}(\{2,3\}|obs) = 0.66$
- $\text{Pr}(\{1,2\}|obs) = 0.17$
- $\text{Pr}(\{1,3\}|obs) = 0.17$

i	obs_i			e_i
	1	2	3	
1	1	1	0	1
2	0	1	1	1
3	1	0	1	1
4	1	0	0	0
5	1	0	1	0

- $\text{Pr}(\{2,3\}|obs) = 0.59$
- $\text{Pr}(\{1,2\}|obs) = 0.35$
- $\text{Pr}(\{1,3\}|obs) = 0.06$

After a redundant test case:

i	obs_i			e_i
	1	2	3	
1	1	1	0	1
2	0	1	1	1
3	1	0	1	1
4	1	0	0	0

- $\mathsf{Pr}(\{2,3\}|obs) = 0.66$
- $\mathsf{Pr}(\{1,2\}|obs) = 0.17$
- $\mathsf{Pr}(\{1,3\}|obs) = 0.17$

After a redundant test case:

i	obs_i 1	2	3	e_i
1	1	1	0	1
2	0	1	1	1
3	1	0	1	1
4	1	0	0	0

i	obs_i 1	2	3	e_i
1	1	1	0	1
2	0	1	1	1
3	1	0	1	1
4	1	0	0	0
5	0	1	1	0

- Pr$(\{2,3\}|obs) = 0.66$
- Pr$(\{1,2\}|obs) = 0.17$
- Pr$(\{1,3\}|obs) = 0.17$

- Pr$(\{1,2\}|obs) = 0.33$
- Pr$(\{1,3\}|obs) = 0.33$
- Pr$(\{2,3\}|obs) = 0.33$

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.

[9] Wes Masri and Rawad Abou Assi. "Cleansing Test Suites from Coincidental Correctness to Enhance Fault-Localization". In: *ICST*. 2010, pp. 165–174.

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.

[10] Yi Miao et al. "Identifying Coincidental Correctness for Fault Localization by Clustering Test Cases". In: *SEKE*. 2012, pp. 267–272.

Minimising Coincidental Correctness – Related work

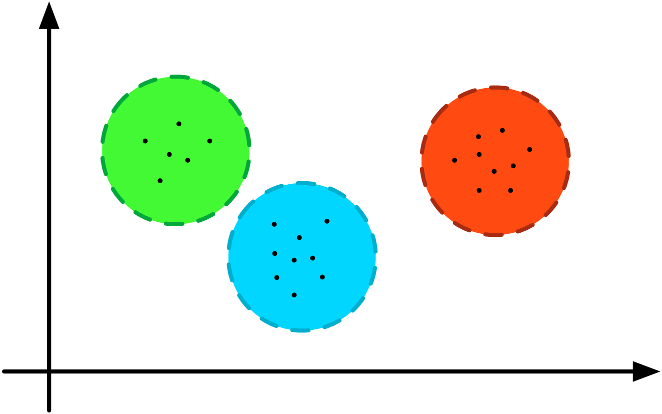
Miao et al.[10] use a similar clustering approach:

- 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.

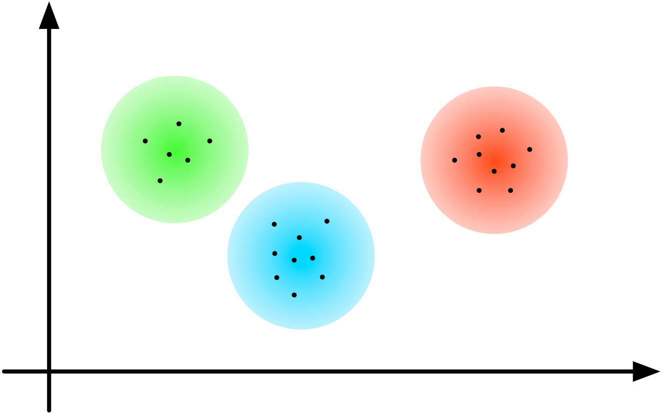
Two strategies:

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

[10] Yi Miao et al. "Identifying Coincidental Correctness for Fault Localization by Clustering Test Cases". In: *SEKE*. 2012, pp. 267–272.



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.

$$\mathsf{Pr}(obs_i, c_i|d) = (1 - c_i) + (c_i \cdot \mathsf{Pr}(obs_i|d))$$

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.

$$\mathsf{Pr}(obs_i, c_i|d) = (1 - c_i) + (c_i \cdot \mathsf{Pr}(obs_i|d))$$

Example:

i	1	2	e_i
1	1	1	1
2	1	0	0
3	0	1	0

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

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.

$$\Pr(obs_i, c_i|d) = (1 - c_i) + (c_i \cdot \Pr(obs_i|d))$$

Example:

i	1	2	e_i	c_i
1	1	1	1	1
2	1	0	0	0.5
3	0	1	0	1

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

$\Pr(\{1\}|obs, c) = 5.0 \times 10^{-4}$

$\Pr(\{2\}|obs, c) = 2.5 \times 10^{-4}$

Table of Contents

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