

Myths of Pairwise Testing

Recognizing Use, Misuse, and Overuse



RBCS

TIME TESTED.
TESTING IMPROVED.

www.RBCS-US.com



Introduction

- ⊕ Ah, myths...
 - ⊕ The Fountain of Youth
 - ⊕ The Philosopher's Stone
 - ⊕ El Dorado
- ⊕ Software testing has myths, too
- ⊕ Pairwise testing techniques have attracted new myths in the last ten years
 - ⊕ What is the proper use of pairwise testing?
 - ⊕ How do we avoid overuse?
 - ⊕ In what situations are other techniques better (and often cheaper)?
- ⊕ Let's find some answers...



Pairwise Testing Is...

- ❖ Testing all possible pairs of options for all possible pairs of factors
 - ❖ Some use orthogonal arrays
 - ❖ Alternatively, find free tools at pairwise.org
- ❖ Options are usually independent (any option can coexist with any option)
- ❖ Classification tree tools allow for constraints, as well as for weighting
- ❖ Often seen as a cure for the common combinatorial explosion



Combinatorial Explosion? What's That?

Enter transaction: Buy - Shares Bought

Enter the purchase of shares of a security (subtract the cost from my cash balance).

Buy - Shares Bought:

Transaction date: 5/5/2013

Account: PW Invest Acct

Memo:

Security name:

Number of shares:

Price paid: per share

Commission: 0.00

Total cost: 0.00

Use cash for this transaction

From this account's cash balance

From

Buttons: Cancel, Clear, Enter/New, Enter/Done

$5 * 5 * 5 * 5 * 5 * 5 * 5 * 5 * 5 = 5^9 = 1,953,125!$ In reality, there are more than five interesting types of inputs for each field, so the number is larger.

How many possible tests for this simple input screen? Assume just five interesting equivalence partitions for each field, and no need to test “behind” the UI.



Police System Decision Table

<i>Condition</i>	1	2	3	4	5	6	7	8
License OK	No	-	-	-	-	-	-	-
Warrant	-	Yes	-	-	-	-	-	-
Registration OK	-	-	No	-	-	-	-	-
Vehicle OK	-	-	-	No	-	-	-	-
Excess Speed	-	-	-	-	1-10	11-20	21-25	>25
<i>Action</i>								
Arrest	Yes	Yes	-	-	-	-	-	Yes
Fix-It Ticket	-	-	Yes	Yes	-	-	-	-
Warning	-	-	-	-	Yes	-	-	-
Fine	+250	+250	+25	+25	+0	+75	+150	+250



ACTS Pairwise Tests

ACTS - ACTS Main Window

System Edit Operations Help

Algorithm: IPOG Strength: 2

System View

- [Root Node]
 - [SYSTEM-Police]
 - License_OK
 - true
 - false
 - Warrant
 - Registration_OK
 - Vehicle_OK
 - Excess_Speed
 - Relations

Test Result

	LICENSE_OK	WARRANT	REGISTRATION_OK	VEHICLE_OK	EXCESS_SPEED
1	true	false	false	false	A
2	false	true	true	true	A
3	true	true	false	true	B
4	false	false	true	false	B
5	true	true	true	false	C
6	false	false	false	true	C
7	true	true	true	false	D
8	false	false	false	true	D
9	true	true	true	false	E
10	false	false	false	true	E

Assume A: XS=0; B: XS=1-10; C: XS: 11-20; D: XS: 21-25; E: XS: 26-???

What's wrong with these tests?

1. Never tests each rule by itself (which decision table testing would do).
2. Tests boundary values of Excess Speed, but not of the total fine (which domain analysis would do).

Additional tests could be added to cover these missing conditions but it would not result in any significant savings over generating the tests using DT and DA.



Why Does PW Miss Obvious Tests?

- ❖ Pairwise techniques assume no deliberate interaction between factors
- ❖ In the police example, interactions are both implicit and explicit
- ❖ In techniques like domain analysis and decision tables, the analysis focuses on the interactions
- ❖ Pairwise techniques can accidentally discover unintended interactions but cannot explore deliberate interactions



So When Might We Use Pairwise?

- ⊕ Often people suggest that we should use pairwise techniques for compatibility testing
- ⊕ Consider testing compatibility of www.rbc-us.com
- ⊕ Four factors: two, three, or four options
 - ⊞ Connection speed: low speed and high speed
 - ⊞ Operating system: Mac, Linux, Windows 7, Windows 8
 - ⊞ Security: OS only, Symantec, Trend Micro, McAfee
 - ⊞ Browser: Firefox, Internet Explorer, Safari
- ⊕ What does the ACTS tool tell us about this?



Pairwise Testing of Website

ACTS - ACTS Main Window

System Edit Operations Help

Algorithm: IPOG Strength: 2

System View

- [Root Node]
 - [SYSTEM-Police]
 - License_OK
 - true
 - false
 - Warrant
 - Registration_OK
 - Vehicle_OK
 - Excess_Speed
 - Relations
 - [SYSTEM-RBCS Website]
 - C_S
 - OS
 - Sec
 - Br
 - Relations

Test Result

	C_S	OS	SEC	BR
1	high	Mac	OS	IE
2	low	Mac	Sym	Saf
3	high	Mac	TM	FF
4	low	Mac	McA	IE
5	high	Linux	OS	Saf
6	low	Linux	Sym	FF
7	low	Linux	TM	IE
8	high	Linux	McA	Saf
9	low	W7	OS	FF
10	high	W7	Sym	IE
11	high	W7	TM	Saf
12	high	W7	McA	FF
13	low	W8	OS	IE
14	high	W8	Sym	Saf
15	low	W8	TM	FF
16	high	W8	McA	Saf



So What Does That Tell Us?

- ✚ Every combination of options for each pair of factors has occurred
- ✚ Any unanticipated interactions between pairs of factors may well result in unexpected outputs
- ✚ Only 16 different test configurations are required
- ✚ But are 16 test configurations worth doing?



Is PW Better than EP?

- ✦ We can cover the equivalence partitions in four tests
- ✦ Every option is tested at least once
- ✦ If we don't think there's a mechanism for interaction...hmmm...
- ✦ You can use some tools (such as ACTS) to force pairs of only those factors that you think might interact
- ✦ Such an approach is probably a better risk-aware approach to testing...

The screenshot shows a software interface with a 'Test Result' table and a 'Statistics' button. The table has five columns: an empty column, 'C_S', 'OS', 'SEC', and 'BR'. It contains four rows of data.

	C_S	OS	SEC	BR
1	high	Mac	Sym	IE
2	low	Linux	TM	Saf
3	high	W7	McA	FF
4	low	W8	OS	IE



Comments on Pairwise Techniques 1

Cem Kaner: “My key issues are the lack of a theory of error and the lack of an oracle. Just because you test two things together doesn't mean you are testing them together in a way that can expose a problem. [E.g., if you test] Calendar Creator Windows 1.0 [with] high-res video and high-res print output, [it will] crash [but only if] you do PRINT PREVIEW while testing these together. All-pairs leads you to combining the devices/settings, but it can't [tell you what to test].”



Comments on Pairwise Techniques 2

- ❖ From two users of PW techniques on DoD projects
 - ❖ Robin Juhl: “It can be very useful [as part of a larger mix]. Even if you can’t do **all** of the pairwise table’s rows, you can...select samples [based on how] many rows [you can do].”
 - ❖ James Nazar: “I have found that just methodically applying EP, BVA, and then combinatorial (pairwise or classification trees) beneficial in identifying interesting tests.... It often results in thoroughly thinking through the constraints between the input factors, platform configurations, and/or environment configurations, [forcing] consideration of ways the system may be used in real operations, but not previously tested. [You may ultimately] apply a different test method (i.e. decision table or domain analysis), but you often have to consider all [to] determine...the best.”



Conclusions

- ❖ Pairwise techniques, while overhyped in some ways, have their uses
- ❖ Careful consideration of options rather than blind application of the technique is required
- ❖ Decision tables and domain analysis might be better techniques when interactions are clear
- ❖ Additional thinking is required to design the tests themselves (actions, inputs and oracle)
- ❖ A broad test using equivalence partitioning and boundary values can help decide whether to test pairwise



To Contact RBCS

For almost 20 years, RBCS has delivered consulting, outsourcing and training services to clients, helping them with software and hardware testing. Employing the industry's most experienced and recognized consultants, RBCS advises its clients, trains their employees, conducts product testing, builds and improves testing groups, and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation and more. To learn more about RBCS, visit www.rbc-us.com.

Address: RBCS, Inc.
31520 Beck Road
Bulverde, TX 78163-3911
USA

Phone: +1 (830) 438-4830
Fax: +1 (830) 438-4831
E-mail: info@rbc-us.com
Web: www.rbc-us.com