Group testing: Finding needles in haystacks

Oliver Johnson
bristoliver.substack.com

University of Bristol

IMO reception London 22nd September 2025

Section 1:
Toxic talk treat teaser

- Professor J is talking to students at the IMO reception.
- 7 plates of delicious post-lecture snacks.
- Professor J's evil nemesis, Dr X, has poisoned one of them.
- Whoever eats that snack will fall asleep for 24 hours.
- How to find the poisoned food, as efficiently as possible?
- Can pay any IMO helper £10 to eat what we tell them.

How to solve the mystery?

- One idea: pay 7 helpers to eat one snack each ('individual testing')
- One will fall asleep.
- Will cost us £70.
- Better idea: use the following strategy (only costs £30).

	Olives	Nuts	Bread sticks	Crisps	Dip	Cheese straws	Jelly
Helper 1	✓	×	✓	×	✓	×	✓
Helper 2	✓	✓	×	×	✓	✓	×
Helper 3	✓	✓	✓	✓	×	×	×

Outcome

	Olives	Nuts	Bread sticks	Crisps	Dip	Cheese straws	Jelly	
Helper 1	✓	×	✓	×	✓	×	✓	😴
Helper 2	✓	✓	×	×	✓	✓	×	😮
Helper 3	✓	✓	✓	✓	×	×	×	😴

- Solution: breadsticks were poisoned.
- Strategy based on binary representation?
- This strategy would always work.
- But what if more than one snack poisoned?

More interesting problems

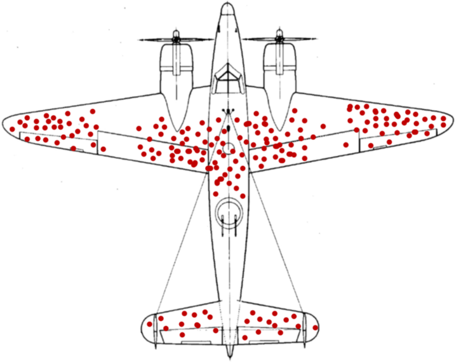
- What if you had 500 snacks, with 10 poisoned?
- How many helpers would we need?
- What should we get them to eat?
- How would we find the poisoned snacks?
- What if some helpers are immune to poison . . . or fall asleep anyway?
- This is **group testing**.

Section 2:
The IMO and me

A bit about me

- 1991, 1992 UK IMO team Sweden and Russia.
- 1995 Cambridge Maths degree.
- 1999 Cambridge PhD.
-
- 2024 Head of School of Mathematics, University of Bristol.

But actually how it looked

- 1991, 1992 UK IMO team Sweden and Russia.
- 1995 Cambridge Maths degree.
- 1999 Cambridge PhD.
- 
- 2024 Head of School of Mathematics, University of Bristol.

Theorem

IMO success is neither a necessary or sufficient condition for an academic career.

- Very rarely have to find functions such that $f(f(2025)+1) = 2025$.
- Hit my talent wall with geometry, combinatorics and other things.
- Found interesting problems via probability (and information theory).
- Write theoretical papers but also work with engineers, CS, biologists.
- Had fun using simple maths ideas for COVID public engagement.

Section 3:
Adaptive group testing algorithms

What is group testing?

- A way of efficiently testing a large population for a rare disease.
- A toy model that throws up some interesting combinatorics problems.
- A framework used in many applied fields.
- See our survey for more detail (joint with Matt Aldridge, Jon Scarlett)
- Free version at `arxiv:1902.06002`.

Start of the group testing problem

- 1942/3, US wanted to test all men joining army for syphilis.
- Tests not cheap, so overall potentially very expensive.
- Condition rare, so test outcomes known with high probability.
- Idea: pool blood from a group of people, **test it together**:
  - If syphilis present in any blood sample, test outcome is positive.
  - If no syphilis present, test outcome is negative.
- Obviously an idealization.
- Call this standard noiseless group testing.

Notation

- Refer to population as 'items' (say N of them).
- Refer to infected people as 'defective' (say $K \ll N$ of them).
- Sometimes refer to defective set $\mathcal{K}$ (collection of infected people).
- Write $\widehat{\mathcal{K}}$ for estimate of defective set.
- Define success probability $\mathbb{P}(\mathrm{suc}) = \mathbb{P}(\widehat{\mathcal{K}} = \mathcal{K})$.
- Do T tests, hope to have high $\mathbb{P}(\mathrm{suc})$.

Olympiad maths . . . the magic number

- Argument: if we want $\mathbb{P}(\text{suc}) = 1$, the pigeonhole principle means we need $T \geq T^* = \log_2 \binom{N}{K}$ tests.
- **Proof**: There are 2^T combinations of test results, but there are $\binom{N}{K}$ possible defective sets that each must give a different set of results.
- Call T^* the magic number.

Theorem (BJA, *ISIT13*)

Choose defective set uniformly from $\binom{N}{K}$ sets of size K. For standard noiseless group testing, any test design and any algorithm:

$$\mathbb{P}(\text{suc}) \leq 2^{-(T^* - T)} = \frac{2^T}{\binom{N}{K}}.$$

- Success probability decays exponentially below magic number of tests.

Dorfman testing

- Simplest strategies are adaptive (choice of tests depends on results of previous tests).

Algorithm (Dorfman testing)

Dorfman suggested two stage strategy:

1. *Split large group of people into pools, test each pool together.*
2. *If negative, know everyone is disease-free.*
3. *If positive, retest individually.*

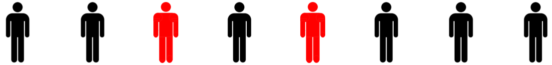
- Basic strategy for COVID-19 pooled testing.
- At low prevalence can save many tests.
- With large initial pools can be inefficient to retest individually.
- Binary search strategy is better.
- Hwang's algorithm (*JASA 1972*) essentially achieves magic number.

Drawback of adaptive testing

- Binary search can take many rounds of tests.
- For COVID this could be close to generation time.
- Trade-off between tests saved and value of each one.
- PCR testing is parallel: need to declare whole test strategy in advance.
- How well can we do non-adaptively?

Section 4:
Non-adaptive group testing algorithms

Standard non-adaptive noiseless group testing



								Outcome **Y**
1	1	1	1	0	0	0	0	Positive
0	0	0	0	1	1	1	1	Positive
1	1	0	0	0	0	0	0	Negative
0	0	1	0	0	0	0	0	Positive
0	0	0	0	1	1	0	0	Positive
0	0	0	0	1	0	0	0	Positive

- Represent pooling strategy via binary test matrix.
- Rows are tests, columns are people or 'items'.
- Put a 1 if item is in test.
- Red denotes being defective

In practice

?	?	?	?	?	?	?	?	Outcome
1	1	1	1	0	0	0	0	Positive
0	0	0	0	1	1	1	1	Positive
1	1	0	0	0	0	0	0	Negative
0	0	1	0	0	0	0	0	Positive
0	0	0	0	1	1	0	0	Positive
0	0	0	0	1	0	0	0	Positive

- Want to infer defective set $\mathcal{K}$.
- Want as few tests as possible.
- Can separate choice of algorithm and design of matrix.

COMP algorithm (Chan, Che, Jaggi, Saligrama 2011)

Algorithm (COMP)

- *Algorithm has two stages:*
  1. *All items in a negative test are non-defective ...*
  2. *Mark all remaining items (Possible Defectives) as defective.*

COMP example

?	?	?	?	?	?	?	
1	0	1	0	0	1	0	Negative
1	1	0	1	0	0	1	Positive
1	0	0	0	1	0	0	Negative
0	1	1	0	1	1	0	Positive
1	0	1	1	0	1	0	Positive

- First, look at negative tests.
- Test 1 is negative, so items 1,3,6 are non-defective.
- Test 3 is negative, so items 1,5 are non-defective.
- Hence items 2,4,7 are possible defectives (PDs).
- COMP algorithm: declare 2,4,7 to be defective.

COMP performance

- COMP makes no errors in the first stage.
- Item labelled as non-defective must indeed be so.
- COMP estimate $\widehat{\mathcal{K}}$ contains no false negatives.
- Works OK, but not really optimal.

DD algorithm (ABJ 2014)

Algorithm (DD)

- *Algorithm has three stages (first stage is same as COMP):*
  1. *All items in a negative test are non-defective . . . leaving smaller set of possible defectives (PDs).*
  2. *Look for positive tests with* **exactly one** *PD item in . . . that item must be defective.*
  3. *Deal with others arbitrarily e.g. mark as non-defective.*

DD example: Stage 2

	0		0			0	
	1		1			1	Positive
	0		0			0	
	1		0			0	Positive
	0		1			0	Positive

- Restrict to submatrix corresponding to the PD set
- Test 4 is positive with one PD item in, so item 2 is defective.
- Test 5 is positive with one PD item in, so item 4 is defective.
- Up to this point, inference is definitely correct.

DD example: Stage 3

	0		0			0	
	1		1			1	Positive
	0		0			0	
	1		0			0	Positive
	0		1			0	Positive

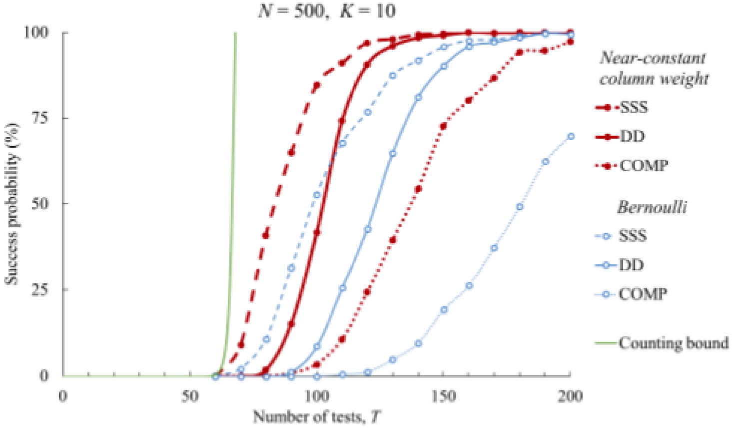
- Don't know about item 7.
- Arbitrarily, make it non-defective (sparsity grounds).
- Probably the obvious algorithm.
- However can prove performance bounds.

Section 5:
Algorithm performance

Matrix designs

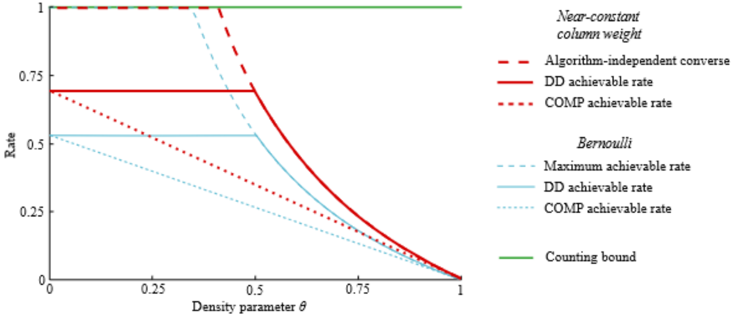
- How to choose test strategy (design the text matrix)?
- COMP and DD should work for any design given enough tests.
- Hope a sensible random matrix performs well on average.
- Common strategy in combinatorics – see ‘probabilistic method’.
- Most basic design is Bernoulli(p) – every item in every test independently with probability p.
- This design is relatively easy to analyse.
- Constant column-weight matrix can perform better (though harder to analyse).

Can give performance bounds by simulation



Can give performance bounds theoretically

- Study regime where $K = N^\theta$, $N \to \infty$ for fixed $0 < \theta < 1$.
- Rate is 'what proportion of magic number of tests do we need?'



Variations on the model

- Can make assumptions more realistic:
  - Binary vs non-binary outcomes
  - Noiseless vs noisy
  - Number of defectives known or unknown
  - Fixed number of defectives vs defective with fixed probability (or community infection structure)

```
|------------|
| THANK      |
| YOU        |
| FOR        |
| LISTENING! |
|------------|
(\__/) ||
(•ㅅ•) ||
/ 　 づ
```