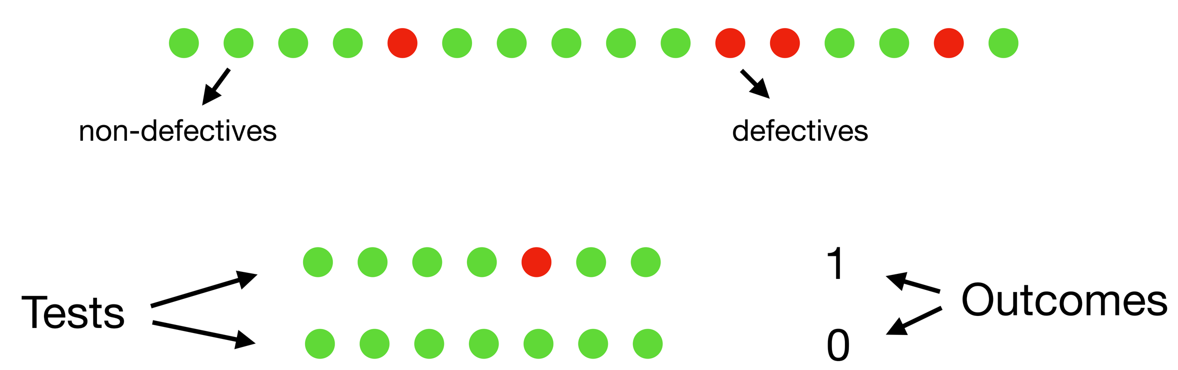
Deletion Resilient Group Testing

Venkata Gandikota, EECS, Syracuse University

joint work with Nikita Polyanskii and Haodong Yang

Introduction

Group Testing: Pooling technique to identify ***k*** defective elements from a population of ***n*** items.



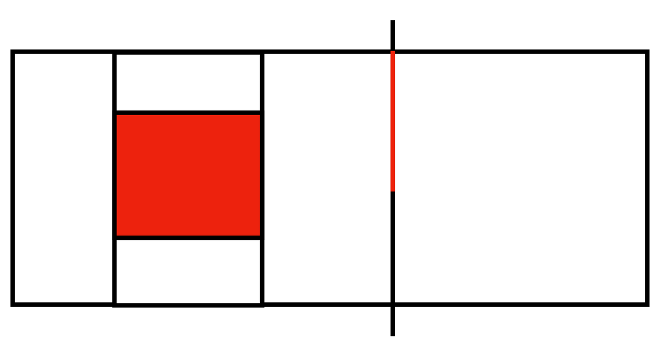
Noiseless case

O(k log n) tests are necessary — counting

O(k² log n) tests are sufficient — disjunct matrices

k-Disjunct Matrix: Binary matrix ***A*** such that:

Union of any ***k*** columns does not contain any other column.



Efficient Decoding algorithms: O(n) time.

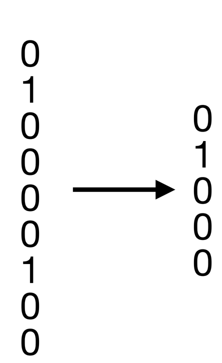
1) Select all columns that are contained in the outcome vector
2) Remove from [n] all items that appear in a 0 test outcome

Setup

Deletion Noise:

Some test outcomes get deleted.

Position of deletions are unknown.



A: Testing Matrix with m rows, n columns

$x^* \in \{0,1\}^n$ (unknown),

$y^* = \text{Test}(A, x^*)$,

$y = \text{del}(y^*, \Delta)$ — deletion of Δ arbitrary entries from y^*

Problem

- Can we identify all k defectives after deletions of Δ test outcomes?
- How many tests are necessary and sufficient for accurate recovery?
- Design efficient recovery algorithms.

Lower bound

Necessary Condition:

Binary matrix A such that for any two disjoint sets of k columns S, T $\subset$ [n], |S|, |T| $\leq$ k,

lcs($\cup_{i \in S} A_i$, $\cup_{j \in T} A_j$) < m - Δ

Any two unions of distinct set of columns should have all distinct m-Δ length subsequences.

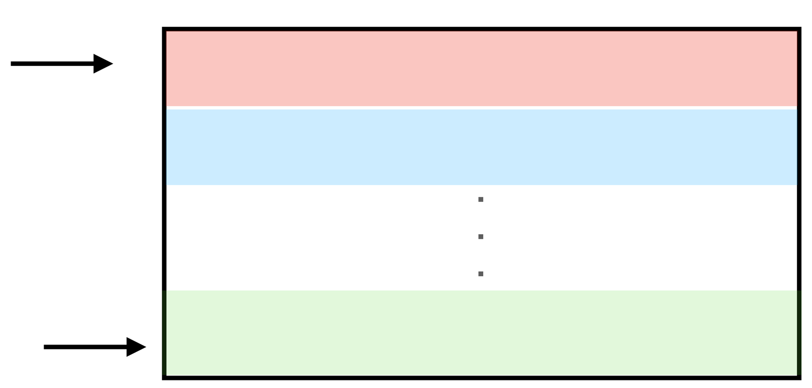
Simple counting argument gives us m > Ω((k + Δ) log n).

Upper bounds

Deletion disjunct matrix: Binary matrix A $\in \{0, 1\}^{m \times n}$ such that for any set of k columns S $\subset$ [n], |S| $\leq$ k, and any column j $\notin$ S, lcs($\cup_{i \in S} A_i$, A_j) < m - Δ

Trivial Construction: Let **B** be a k-disjunct matrix with O(k² log n) rows.

Construct **A** by repeating each row of B $(\Delta + 1)$ times.



#tests = O(Δ k² log n)

Decoding algorithm:

A: Deletion disjunct matrix

$x^* \in \{0,1\}^n$ (unknown), $y^* = \text{Test}(A, x^*)$, $y = \text{del}(y^*, \Delta)$

- Recover original outcome vector y* as follows:
  - greedily complete each run of 0's or 1's to the next multiple of $(\Delta + 1)$
- Recover the test outcomes for B, z
- Use regular GT recovery algorithms with z = Test(B, x*).

Correctness:

- Any adversary can delete at most Δ test outcomes.
- Therefore, they cannot delete all outcomes corresponding to the repetitions of any single row of B.
- e.g. for a run of s$(\Delta + 1)$ 0's, after at most Δ deletions, we will be left with a run of least (s-1)(Δ + 1) + t consecutive 0's (for t < Δ +1)

Alternate Approach

Asymmetric deletion distance $d_a(x,y)$:

Let x, y be n-length binary strings.

$d_a(x,y)$ is the maximum number of deletions Δ such that for any m-Δ length subsequences, x' of x and y' of y

there exists an index i $\in$ [m-Δ] such that $x'_i = 1$ and $y'_i = 0$

Asymmetric deletion disjunct matrix: Binary matrix A $\in \{0, 1\}^{m \times n}$ such that for any set of k columns S $\subset$ [n], |S| $\leq$ k, and an column j $\notin$ S, d_a($\cup_{i \in S} A_i$, A_j) > Δ

Decoding Algorithm

A: Asymmetric deletion disjunct matrix

$x^* \in \{0,1\}^n$ (unknown), $y^* = \text{Test}(A, x^*)$, $y = \text{del}(y^*, \Delta)$

- Let S = $\varnothing$
- For each column v of A:
  - for each m-Δ length subseq $v^{(i)}$ of v:
    - if there is no p $\in$ [m-Δ], s.t. y(p) = 0 and $v^{(i)}(p) = 1$,
      - then add v to S
- Return S

Correctness: Follows from the properties of asymmetric deletion disjunct matrix A.

Runtime: O(n $m^{2\Delta}$) — inefficient for large Δ

Construction: random Bernoulli matrices with p = O(1/k) are asymmetric deletion disjunct matrices with high probability if

$$m = \Omega(k^2 \log n + k\Delta)$$

Work in Progress

Q: How to construct asymmetric deletion disjunct matrices?

Q: Can we tighten the LB?

Q: How to decode efficiently?