Group Testing Algorithms: Bounds and Simulations

Matthew Aldridge, Leonardo Baldassini, Oliver Johnson

Nemo Fournier

January 17, 2020

Outline

Setting and Framework

Some algorithms

Analysis of DD algorithm

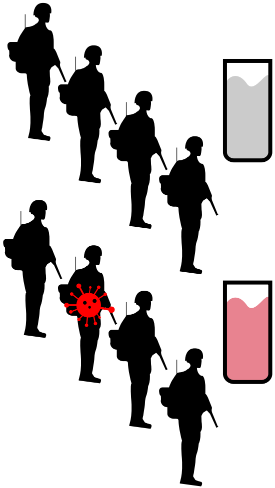
More bounds

Simulations

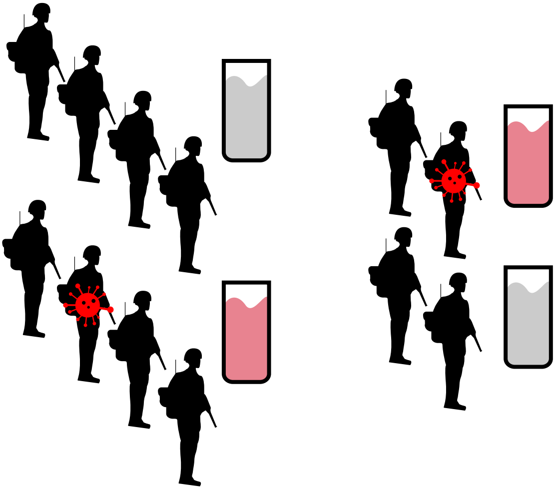
Perspectives

A natural problem [Dorfman, 1943]

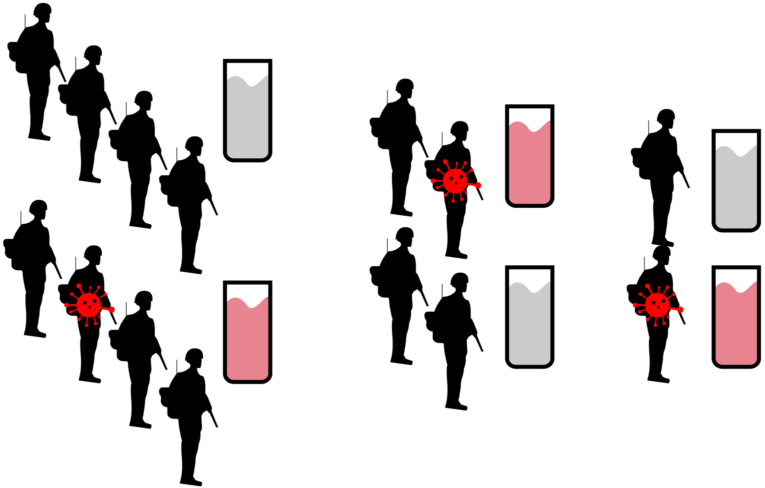
A natural problem [Dorfman, 1943]



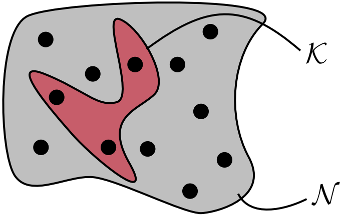
A natural problem [Dorfman, 1943]



A natural problem [Dorfman, 1943]



The Framework



$$\left.\begin{bmatrix}0\\1\\1\\0\\\vdots\\x_{N,t}\end{bmatrix}\right\} N = |\mathcal{N}|$$

test t

$$y_t = \begin{cases} 1 \text{ if } \left|\{i \in \mathcal{K} \mid x_{i,t} = 1\}\right| \geq 1 \\ 0 \text{ if } \left|\{i \in \mathcal{K} \mid x_{i,t} = 1\}\right| = 0 \end{cases}$$

Non adaptative testing

$$\mathsf{X} = \begin{bmatrix} x_{1,1} & & & & \\ x_{2,1} & & & & \\ \vdots & T_2 & T_3 & \cdots & T_T \\ x_{N-1,1} & & & & \\ x_{N,1} & & & & \end{bmatrix}$$

Non adaptative testing

$$\mathbf{X} = \begin{bmatrix} \begin{matrix} x_{1,1} \\ x_{2,1} \\ \vdots \\ x_{N-1,1} \\ x_{N,1} \end{matrix} & T_2 & T_3 & \cdots & T_T \end{bmatrix} \mapsto \begin{bmatrix} y_1 & y_2 & \cdots & y_{T-1} & y_T \end{bmatrix} = \mathbf{y}$$

Non adaptative testing

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & & & & \\ x_{2,1} & & & & \\ \vdots & T_2 & T_3 & \cdots & T_T \\ x_{N-1,1} & & & & \\ x_{N,1} & & & & \end{bmatrix} \mapsto \begin{bmatrix} y_1 & y_2 & \cdots & y_{T-1} & y_T \end{bmatrix} = \mathbf{y}$$

Design stage: $\mathbf{X}$

Non adaptative testing

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & & & & \\ x_{2,1} & & & & \\ \vdots & T_2 & T_3 & \cdots & T_T \\ x_{N-1,1} & & & & \\ x_{N,1} & & & & \end{bmatrix} \mapsto \begin{bmatrix} y_1 & y_2 & \cdots & y_{T-1} & y_T \end{bmatrix} = \mathbf{y}$$

Design stage: $\mathbf{X}$

Detection stage: $A(\mathbf{X}, \mathbf{y}) \mapsto \hat{\mathcal{K}}$

Non adaptative testing

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & & & & \\ x_{2,1} & & & & \\ \vdots & T_2 & T_3 & \cdots & T_T \\ x_{N-1,1} & & & & \\ x_{N,1} & & & & \end{bmatrix} \mapsto \begin{bmatrix} y_1 & y_2 & \cdots & y_{T-1} & y_T \end{bmatrix} = \mathbf{y}$$

Design stage: $\mathbf{X}$
Detection stage: $A(\mathbf{X},\mathbf{y}) \mapsto \hat{\mathcal{K}}$

$$\epsilon = \mathbf{P}_{\mathbf{X},\mathcal{K}}\left(\hat{\mathcal{K}} \neq \mathcal{K}\right) \qquad r = \frac{\log_2 \binom{N}{K}}{T}$$

Some assumptions

Random test matrix: $x_{i,t} \sim \mathcal{B}(p)$

Density regime: $K \approx N^{1-\beta}$

COMP algorithm

If an item appears in a negative test, then it cannot be defective.

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \color{blue}{0} \end{bmatrix}$$

COMP algorithm

If an item appears in a negative test, then it cannot be defective.

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \begin{matrix} - \\ 2 \\ 3 \\ \vdots \\ - \\ N \end{matrix}$$

$$\begin{bmatrix} 0 \end{bmatrix}$$

COMP algorithm

If an item appears in a negative test, then it cannot be defective.

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \begin{matrix} - \\ \color{blue}{2} \\ \color{blue}{3} \\ \vdots \\ - \\ \color{blue}{N} \end{matrix}$$

$$\begin{bmatrix} \color{blue}{0} \end{bmatrix} \color{blue}{\hat{\mathcal{K}}^c}$$

COMP algorithm

If an item appears in a negative test, then it cannot be defective.

$$
\begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix}
\begin{matrix} - \\ 2 \\ 3 \\ \vdots \\ - \\ N \end{matrix}
\qquad \longrightarrow \text{only false positives}
$$

$$
\begin{bmatrix} 0 \end{bmatrix} \hat{\mathcal{K}}^{c}
$$

DD algorithm

If a positive test contains only one possibly defective item, then this item is ***definitely defective***

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \color{blue}{0} \end{bmatrix}$$

DD algorithm

If a positive test contains only one possibly defective item, then this item is ***definitely defective***

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \begin{matrix} \\ 2 \\ 3 \\ \vdots \\ \\ N \end{matrix}$$

$$\begin{bmatrix} 0 \end{bmatrix}$$

DD algorithm

If a positive test contains only one possibly defective item, then this item is ***definitely defective***

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{matrix} \\ 2 \\ 3 \\ \vdots \\ \\ N \end{matrix}$$

$$\begin{bmatrix} 0 & 1 \end{bmatrix}$$

DD algorithm

If a positive test contains only one possibly defective item, then this item is ***definitely defective***

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{matrix} \color{red}{1} \\ \color{blue}{2} \\ \color{blue}{3} \\ \vdots \\ \\ \color{blue}{N} \end{matrix}$$

$$\begin{bmatrix} \color{blue}{0} & \color{red}{1} \end{bmatrix}$$

DD algorithm

If a positive test contains only one possibly defective item, then this item is ***definitely defective***

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ \\ N \end{matrix}$$

$$\begin{bmatrix} 0 & 1 \end{bmatrix} \hat{\mathcal{K}}$$

DD algorithm

*If a positive test contains only one possibly defective item, then this item is **definitely defective***

$$\begin{bmatrix} & 0 & 1 & \\ & 1 & 1 & \\ & 1 & 1 & \\ & \vdots & \vdots & \\ & 0 & 0 & \\ & 1 & 1 & \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ \\ N \end{matrix}$$

$$\begin{bmatrix} & 0 & 1 & \end{bmatrix} \hat{\mathcal{K}}$$

$\rightarrow$ only false negatives

SCOMP and SSS algorithms

SCOMP: iterative DD algorithm

SSS: an ILP formulation

Analysis of DD algorithm (1)

- non defective $\mathcal{ND}$
- possibly defective $\mathcal{PD} = \mathcal{ND}^c = \mathcal{K} \cup \mathcal{G}$
- say that $i \in \mathcal{PD}$ is **definitely defective** if there is a positive test where i is the only $\mathcal{PD}$

Analysis of DD algorithm (1)

- non defective $\mathcal{ND}$
- possibly defective $\mathcal{PD} = \mathcal{ND}^c = \mathcal{K} \cup \mathcal{G}$
- say that $i \in \mathcal{PD}$ is **definitely defective** if there is a positive test where i is the only $\mathcal{PD}$

Define, given $\mathbf{X}$ and $\mathcal{K}$:

$$L_0 = \#\text{ test with no defective items in it}$$

$$L_i = \#\text{ test containing } i \text{ and no other element of } \mathcal{PD}$$

$$L_+ = \#\text{ other tests}$$

$$\mathbf{P}\{\text{success}\} = \mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\}$$

Analysis of DD algorithm (2)

$\mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\}$: hard to compute

Analysis of DD algorithm (2)

$\mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\}$: hard to compute

Idea: condition on l_0 and G:

$$\mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\} = \sum_{l_0=0}^{T} \sum_{g=0}^{N-K} \mathbf{P}\{L_0 = l_0\} \mathbf{P}\{G = g \mid L_0 = l_0\} \times \mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0 \mid L_0 = l_0, G = g\}$$

Analysis of DD algorithm (2)

$\mathbf{P}\{L_1 \neq 0, \dots, L_K \neq 0\}$: hard to compute
Idea: condition on l_0 and G:

$$\mathbf{P}\{L_1 \neq 0, \dots, L_K \neq 0\} = \sum_{l_0=0}^{T} \sum_{g=0}^{N-K} \mathbf{P}\{L_0 = l_0\} \mathbf{P}\{G = g \mid L_0 = l_0\} \times \mathbf{P}\{L_1 \neq 0, \dots, L_K \neq 0 \mid L_0 = l_0, G = g\}$$

- $L_0 \sim Bin(T, (1-p)^K)$

Analysis of DD algorithm (2)

$\mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\}$: hard to compute
Idea: condition on l_0 and G:

$$\mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\} = \sum_{l_0=0}^{T} \sum_{g=0}^{N-K} \mathbf{P}\{L_0 = l_0\} \mathbf{P}\{G = g \mid L_0 = l_0\} \times \mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0 \mid L_0 = l_0, G = g\}$$

- $L_0 \sim Bin(T, (1-p)^K)$
- $G|L_0 \sim Bin(N-K, (1-p)^{L_0})$

Analysis of DD algorithm (2)

$\mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\}$: hard to compute

Idea: condition on l_0 and G:

$$\mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\} = \sum_{l_0=0}^{T} \sum_{g=0}^{N-K} \mathbf{P}\{L_0 = l_0\} \mathbf{P}\{G = g \mid L_0 = l_0\} \times \mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0 \mid L_0 = l_0, G = g\}$$

- $L_0 \sim Bin(T, (1-p)^K)$
- $G | L_0 \sim Bin(N-K, (1-p)^{L_0})$
- $(L_i)_{1 \leq i \leq K} | L_0, G$: harder, but essentially multinomial

Analysis of DD algorithm (2)

$\mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\}$: hard to compute

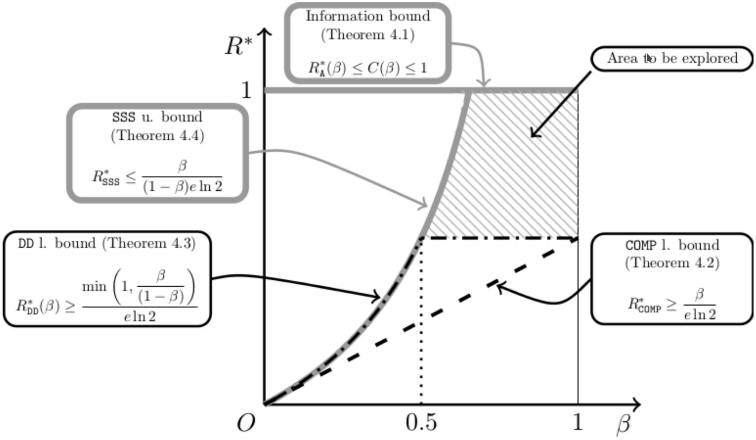
Idea: condition on l_0 and G:

$$\mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0\} = \sum_{l_0=0}^{T} \sum_{g=0}^{N-K} \mathbf{P}\{L_0 = l_0\} \mathbf{P}\{G = g \mid L_0 = l_0\} \times \mathbf{P}\{L_1 \neq 0, \ldots, L_K \neq 0 \mid L_0 = l_0, G = g\}$$

- $L_0 \sim Bin(T, (1-p)^K)$
- $G|L_0 \sim Bin(N-K, (1-p)^{L_0})$
- $(L_i)_{1 \leq i \leq K}|L_0, G$: harder, but essentially multinomial

$$\mathbf{P}\{\text{success}\} = \sum_{l_0=0}^{T} \sum_{g=0}^{N-K} b(l_0, T, (1-p)^K) b(g, N-K, (1-p)^{l_0}) \Phi_K(g, l_0)$$

Rate bounds


Information bound
(Theorem 4.1)
$R^*_A(\beta) \leq C(\beta) \leq 1$
Area to be explored
SSS u. bound
(Theorem 4.4)
$R^*_{SSS} \leq \frac{\beta}{(1-\beta)e \ln 2}$
DD l. bound (Theorem 4.3)
$R^*_{DD}(\beta) \geq \frac{\min\left(1, \frac{\beta}{(1-\beta)}\right)}{e \ln 2}$
COMP l. bound
(Theorem 4.2)
$R^*_{COMP} \geq \frac{\beta}{e \ln 2}$
R^*
1
O
0.5
1
β

Comparisons of the Algorithms

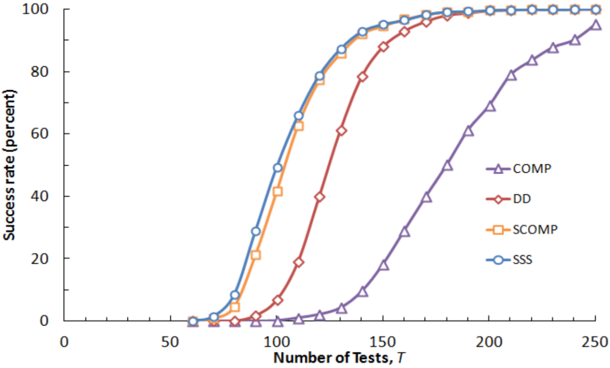


Figure: $N = 500, K = 10, p = 1/10$

Simulation vs Bounds

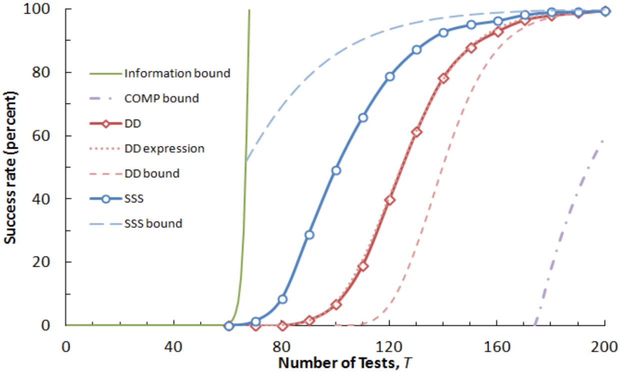


Figure: $N = 500, K = 10, p = 1/10$

Sparsity and Density

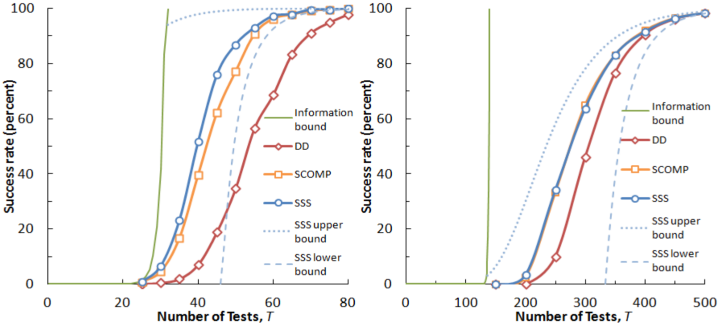


Figure: $N = 500$, left:$K = 4$, $p = 1/4$, right:$K = 25$, $p = 1/25$

Why do we care?

- Many problems can be seen as group testing (Biology (DNA, diseases), Communication (Anomaly discovery in networks, MAC channels, cognitive radios), Information Technology (data compression, cybersecurity), Data science in general (from counterfeit coins to graph problems), Theoretical Computer Science (graph problems, complexity theory)
- This paper proposes a precise framework and works out a part of the capacity spectrum
- Still a limited case: noiseless, perfect recovery, non-adaptative

$$R^*_{COMP} \geq \frac{\beta}{e \ln 2} \approx 0.53\beta$$
$$R^*_{DD} \geq \frac{1}{e \ln 2} \min\left\{1, \frac{\beta}{1-\beta}\right\} \approx 0.53 \min\left\{1, \frac{\beta}{1-\beta}\right\}$$
$$R^*_{SSS} \leq \frac{1}{e \ln 2} \frac{\beta}{1-\beta}$$

Conjecture $R^*_{SCOMP} \begin{cases} = \frac{1}{e \ln 2} \frac{\beta}{1-\beta} & \text{for } \beta \leq 1/2 \\ \geq \frac{1}{e \ln 2} & \text{for } \beta > 1/2 \end{cases}$

SCOMP algorithm

- use DD algorithm $\rightarrow \hat{\mathcal{K}}$
- while $\hat{\mathcal{K}}$ is not satisfying: find i in $\mathcal{PD}$ which appears in the largest number of tests unexplained by $\hat{\mathcal{K}}$ and do $\hat{\mathcal{K}} \leftarrow \hat{\mathcal{K}} \cup \{i\}$

SSS algorithm

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{1}^\top \mathbf{z} \\
\text{subject to} \quad & \mathrm{x}_t = 0 \cdot \mathbf{z} \text{ for } t \text{ with } y_t = 0 \\
& \mathrm{x}_t \cdot \mathbf{z} \leq 1 \text{ for } t \text{ with } y_t = 1 \\
& \mathbf{z} \in \{0,1\}^N
\end{aligned}
$$