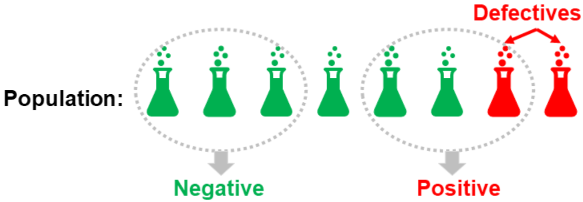
An Analysis of the DD Algorithm for Group Testing with Size-Constrained Tests

Nelvin Tan and Jonathan Scarlett

National University of Singapore (NUS)

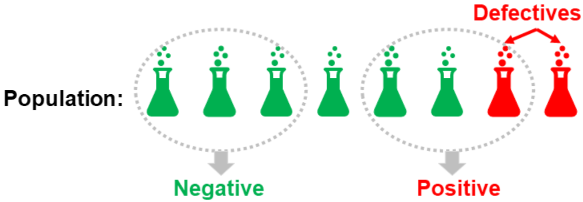
May 2021

Introduction



- **Goal:** Identify a subset of defective items within a larger set of items based on pooled tests.
- Can help to reduce the #tests, which is ideal when tests are costly.
- **Some applications:**
  - Medical testing (e.g., COVID-19)
  - Data science
  - Communication protocols

Introduction



- **Goal:** Identify a subset of defective items within a larger set of items based on pooled tests.
- Can help to reduce the #tests, which is ideal when tests are costly.
- **Some applications:**
  - Medical testing (e.g., COVID-19)
  - Data science
  - Communication protocols

Setup

- Population of n items labelled $\{1, \dots, n\}$.
- Defective set $\mathcal{S} \subset \{1, \dots, n\}$, where $k = |\mathcal{S}| = \Theta\big(n^{\theta}\big)$, for $\theta \in [0, 1)$.
- We consider the following settings:
  - **Combinatorial** prior: Defective set $\mathcal{S}$ chosen uniformly among all sets of size k.
  - **Non-adaptive:** Test pools are designed in advance.
  - **Noiseless:** Get a +ve test outcome if there is least one defective item, and a -ve outcome if there is no defective item.
  - **Small error probability recovery:** Produce $\widehat{\mathcal{S}}$ such that

$$\mathbb{P}[\widehat{\mathcal{S}} \neq \mathcal{S}] \to 0 \text{ as } n \to \infty,$$

where the probability is taken over the randomness of $\mathcal{S}$ and the test design.

Setup

- Population of n items labelled $\{1, \ldots, n\}$.
- Defective set $\mathcal{S} \subset \{1, \ldots, n\}$, where $k = |\mathcal{S}| = \Theta\big(n^{\theta}\big)$, for $\theta \in [0, 1)$.
- We consider the following settings:
  - **Combinatorial** prior: Defective set $\mathcal{S}$ chosen uniformly among all sets of size k.
  - **Non-adaptive:** Test pools are designed in advance.
  - **Noiseless:** Get a $+$ve test outcome if there is least one defective item, and a -ve outcome if there is no defective item.
  - **Small error probability recovery:** Produce $\widehat{\mathcal{S}}$ such that

$$\mathbb{P}\big[\widehat{\mathcal{S}} \neq \mathcal{S}\big] \to 0 \text{ as } n \to \infty,$$

where the probability is taken over the randomness of $\mathcal{S}$ and the test design.

Sparsity-Constrained Group Testing

- Tests are size-constrained and thus contain no more than ρ items per test.
- **Applications:** Testing equipment may be limited in the volume of samples it receives, or large pools may be unsuitable due to dilution effects,

- Previous work in literature shows that $\rho \in \Theta\left(\frac{n}{k}\right)$ is required to attain optimal scaling laws for the unconstrained setting.
- Hence, we are interested in the regime $\rho \in o\left(\frac{n}{k}\right)$, and more specifically $\rho = \Theta\left(\left(\frac{n}{k}\right)^{\beta}\right)$, for $\beta \in [0, 1)$.

Sparsity-Constrained Group Testing

- Tests are size-constrained and thus contain no more than ρ items per test.
- **Applications:** Testing equipment may be limited in the volume of samples it receives, or large pools may be unsuitable due to dilution effects,

- Previous work in literature shows that $\rho \in \Theta\left(\frac{n}{k}\right)$ is required to attain optimal scaling laws for the unconstrained setting.
- Hence, we are interested in the regime $\rho \in o\left(\frac{n}{k}\right)$, and more specifically $\rho = \Theta\left(\left(\frac{n}{k}\right)^{\beta}\right)$, for $\beta \in [0, 1)$.

Previous Result

Under our setup (recall $k = \Theta(n^\theta)$ and $\rho = \Theta\big(\big(\frac{n}{k}\big)^\beta\big)$), we have:

- **Converse:** $\frac{1-6\epsilon}{1-\beta} \cdot \frac{n}{\rho}$.
- **Achievability:** $\left\lceil \frac{1+\epsilon}{(1-\theta)(1-\beta)} \right\rceil \cdot \left\lceil \frac{n}{\rho} \right\rceil$.
- **Improved converse (when $\beta = 0$):** $\max\left\{\left(1 + \left\lfloor \frac{\theta}{1-\theta} \right\rfloor\right)\frac{n}{\rho}, \frac{2n}{\rho+1}\right\}$.
- **Improved achievability (when $\beta = 0$):** $\max\left\{\left(1 + \left\lfloor \frac{\theta}{1-\theta} \right\rfloor\right)\frac{n}{\rho}, \frac{2n}{\rho+1}\right\}$.

Note that ϵ arbitrarily small positive constant.

- **Goal:** Get similar improvements (in the constant factor) for large ρ (i.e., when $\beta > 0$).

Previous Result

Under our setup (recall $k = \Theta(n^\theta)$ and $\rho = \Theta\big(\big(\frac{n}{k}\big)^\beta\big)$), we have:

- **Converse:** $\frac{1-6\epsilon}{1-\beta} \cdot \frac{n}{\rho}$.
- **Achievability:** $\left\lceil \frac{1+\epsilon}{(1-\theta)(1-\beta)} \right\rceil \cdot \left\lceil \frac{n}{\rho} \right\rceil$.
- **Improved converse (when $\beta = 0$):** $\max\left\{\left(1 + \left\lfloor \frac{\theta}{1-\theta} \right\rfloor\right)\frac{n}{\rho}, \frac{2n}{\rho+1}\right\}$.
- **Improved achievability (when $\beta = 0$):** $\max\left\{\left(1 + \left\lfloor \frac{\theta}{1-\theta} \right\rfloor\right)\frac{n}{\rho}, \frac{2n}{\rho+1}\right\}$.

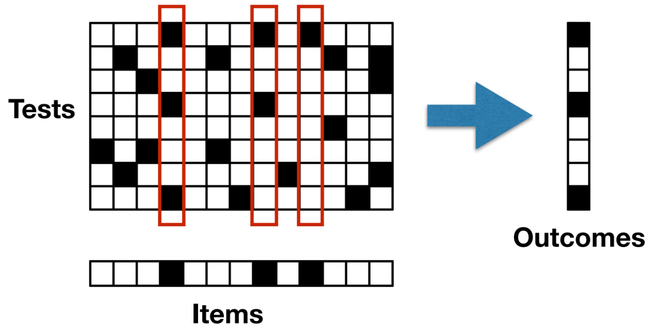
Note that ϵ arbitrarily small positive constant.

- **Goal:** Get similar improvements (in the constant factor) for large ρ (i.e., when $\beta > 0$).

Outline

Randomized Test Design

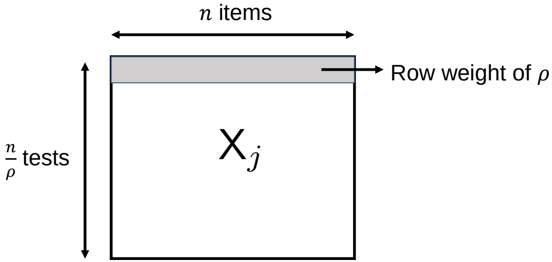
Example of a random test matrix:



- The i-th column represents the tests that the i-th item participates in.
- The i-th row determines the items in the i-th test pool.

Randomized Test Design

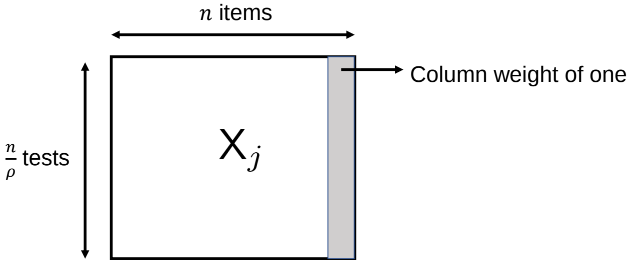
Uniformly sample the sub-matrix X_j:



- Sample c sub-matrices $(\mathsf{X}_1, \dots, \mathsf{X}_c)$ independently.
- Form test matrix by concatenating $\mathsf{X}_1, \dots, \mathsf{X}_c$ vertically, giving us a doubly-constrained matrix.
- Column weight restriction is not strictly imposed by the testing constraints but helps in avoiding "bad" events where some items are not being tested.

Randomized Test Design

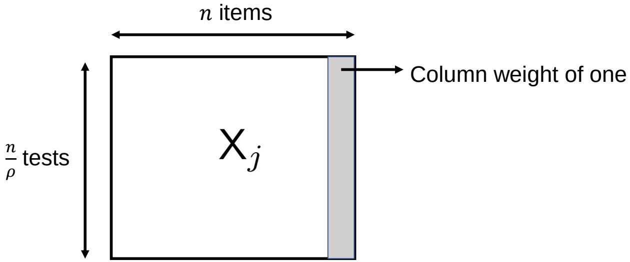
Uniformly sample the sub-matrix X_j:



- Sample c sub-matrices $(\mathsf{X}_1, \ldots, \mathsf{X}_c)$ independently.
- Form test matrix by concatenating $\mathsf{X}_1, \ldots, \mathsf{X}_c$ vertically, giving us a doubly-constrained matrix.
- Column weight restriction is not strictly imposed by the testing constraints but helps in avoiding "bad" events where some items are not being tested.

Randomized Test Design

Uniformly sample the sub-matrix X_j:



- Sample c sub-matrices $(\mathsf{X}_1, \ldots, \mathsf{X}_c)$ independently.
- Form test matrix by concatenating $\mathsf{X}_1, \ldots, \mathsf{X}_c$ vertically, giving us a doubly-constrained matrix.
- Column weight restriction is not strictly imposed by the testing constraints but helps in avoiding "bad" events where some items are not being tested.

Definite Defectives (DD) Algorithm

Some useful observations:

- Any item in a negative test is definitely non-defective (DND).
- All other items can (initally) be considered possibly defective (PD).
- If a test contains only one PD item, then that item is definitely defective (DD).

Algorithms:

- **COMP (previous):** Declare DND items to be non-def. and the rest def.
- **DD (this talk):** Declare DD items to be def. and the rest non-def.

Main Result

Theorem

For $k = \Theta\big(n^{\theta}\big)$*, with* $\theta \in [0,1)$*, and* $\rho = \Theta\Big(\big(\frac{n}{k}\big)^{\beta}\Big)$*, with* $\beta \in [0,1)$*, for any integer* c *satisfying:*

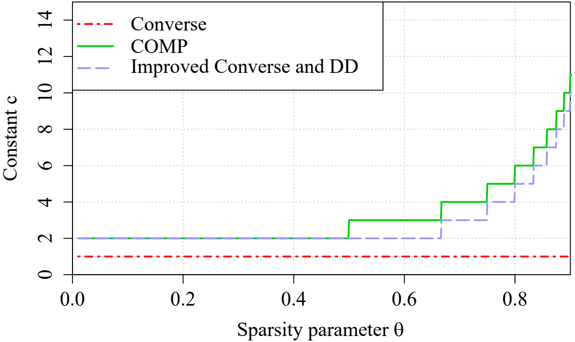
- *If* $\theta \geq \frac{1}{2}$: $c > \frac{\theta}{(1-\theta)(1-\beta)}$ *and* $c \geq \frac{2-\beta}{1-\beta}$*;*
- *If* $\theta < \frac{1}{2}$*:* $c \geq \frac{1-\theta\beta}{(1-\theta)(1-\beta)}$*;*

the DD algorithm with $\frac{cn}{\rho}$ *tests recovers the defective set* $\mathcal{S}$ *with asymptotically vanishing error probability.*

- This is an improvement over previous acheivability results.

Main Result

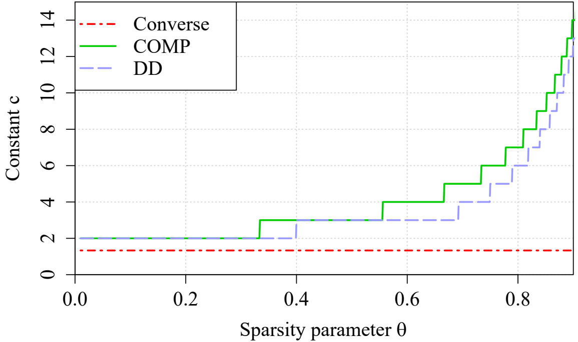
Plot for $\beta = 0$:



- For $\beta = 0$, our result coincides with the improved achievability result.
- For each β shown (0, 0.25, and 0.75), there are strict improvements over COMP (previous work), and the gap widens for $\beta = 0.75$.

Main Result

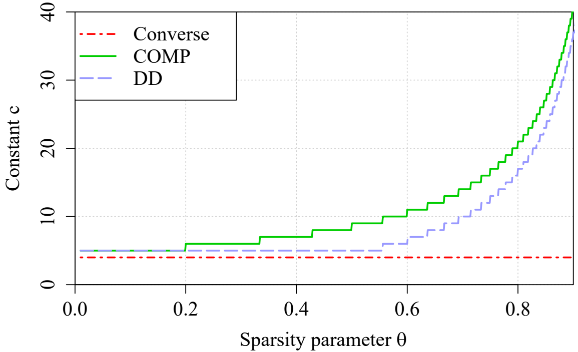
Plot for $\beta = 0.25$:



- For $\beta = 0$, our result coincides with the improved achievability result.
- For each β shown (0, 0.25, and 0.75), there are strict improvements over COMP (previous work), and the gap widens for $\beta = 0.75$.

Main Result

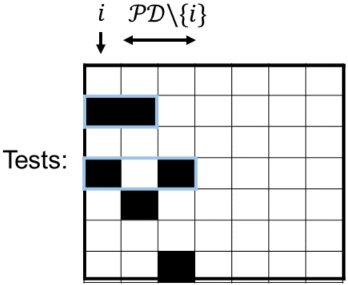
Plot for $\beta = 0.75$:



- For $\beta = 0$, our result coincides with the improved achievability result.
- For each β shown (0, 0.25, and 0.75), there are strict improvements over COMP (previous work), and the gap widens for $\beta = 0.75$.

Analysis Outline of the DD Algorithm

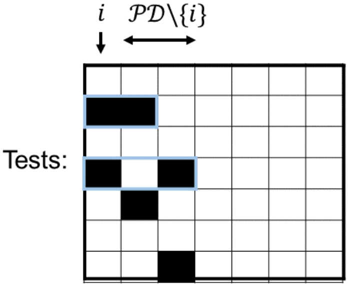
- **DD algorithm:**
  - Declare all DD items as defective and the rest non-defective.
  - DD algorithm makes a mistake if any defective item i is masked by $\mathcal{PD} \setminus \{i\}$ in the test matrix.



- We used independent sub-matrices X_j's so that we can study the above error event for X_j and build our way up.

Analysis Outline of the DD Algorithm

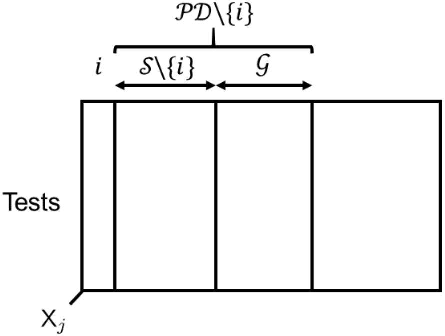
- **DD algorithm:**
  - Declare all DD items as defective and the rest non-defective.
  - DD algorithm makes a mistake if any defective item i is masked by $\mathcal{PD} \setminus \{i\}$ in the test matrix.



- We used independent sub-matrices X_j's so that we can study the above error event for X_j and build our way up.

Analysis Outline of the DD Algorithm

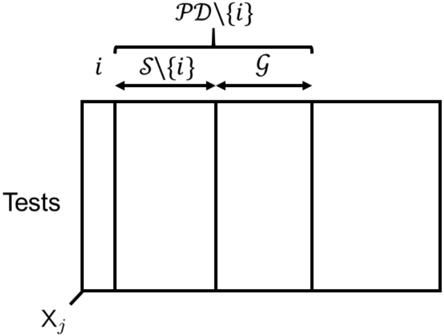
- For a given X_j and a defective item i, the error event can be simplified in the following manner:



- Hence, the error probability can be upper bounded by $\mathbb{P}[i \text{ masked by } \mathcal{S} \setminus \{i\}] + \mathbb{P}[i \text{ masked by } \mathcal{G} | i \text{ not masked by } \mathcal{S} \setminus \{i\}]$.

Analysis Outline of the DD Algorithm

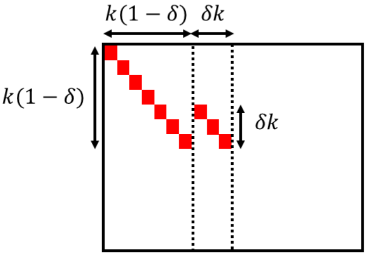
- For a given X_j and a defective item i, the error event can be simplified in the following manner:



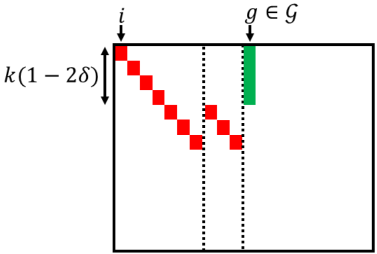
- Hence, the error probability can be upper bounded by
$\mathbb{P}[i \text{ masked by } \mathcal{S} \setminus \{i\}] + \mathbb{P}[i \text{ masked by } \mathcal{G} | i \text{ not masked by } \mathcal{S} \setminus \{i\}]$.

Analysis Outline of the DD Algorithm

- Conditioned on the #positive tests $k(1-\delta)$ and the size of $\mathcal{G}$, we can upper bound $\mathbb{P}\big[i \text{ masked by } \mathcal{S}\setminus\{i\}\big] + \mathbb{P}\big[i \text{ masked by } \mathcal{G} | i \text{ not masked by } \mathcal{S}\setminus\{i\}\big]$.
- This gives us the upper bound $\frac{2\delta k}{k} + \frac{|\mathcal{G}|}{k(1-2\delta)}$.



$\mathbb{P}\big[i \text{ masked by } \mathcal{S}\setminus\{i\}\big]$



$\mathbb{P}\big[i \text{ masked by } \mathcal{G} | i \text{ not masked by } \mathcal{S}\setminus\{i\}\big]$

Analysis Outline of the DD Algorithm

- We have an upper bound on the error probability for a given sub-matrix X_j and defective item i.
- Recall that we need the error probability to hold for the entire test matrix and for any defective item.
  - **Extend to entire test matrix:** Repeatedly multiply c times (by independence of sub-matrices).
  - **Extend to any defective item:** Multiply by k (by union bound).
- Finally, we choose an appropriate c such that the final upper bound is vanishing (as $n \to \infty$).

Analysis Outline of the DD Algorithm

- We have an upper bound on the error probability for a given sub-matrix X_j and defective item i.
- Recall that we need the error probability to hold for the entire test matrix and for any defective item.
  - **Extend to entire test matrix:** Repeatedly multiply c times (by independence of sub-matrices).
  - **Extend to any defective item:** Multiply by k (by union bound).
- Finally, we choose an appropriate c such that the final upper bound is vanishing (as $n \to \infty$).

Analysis Outline of the DD Algorithm

- We have an upper bound on the error probability for a given sub-matrix X_j and defective item i.
- Recall that we need the error probability to hold for the entire test matrix and for any defective item.
  - **Extend to entire test matrix:** Repeatedly multiply c times (by independence of sub-matrices).
  - **Extend to any defective item:** Multiply by k (by union bound).
- Finally, we choose an appropriate c such that the final upper bound is vanishing (as $n \to \infty$).

Summary

- We used
  - independent doubly-constrained sub-matrices;
  - the DD algorithm.
- We showed that this improves the constant term in the achievability result.