LDPC Codes for Quantitative Group Testing with a Non-Binary Alphabet

Michael Lentmaier[†]

Joint work with Mgeni Makambi Mashauri[†] and Alexandre Graell i Amat[‡]

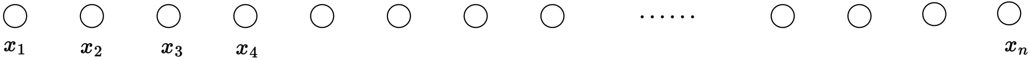
†Department of Electrical and Information Technology, Lund University, Sweden
‡Department of Electrical Engineering, Chalmers University of Technology, Sweden

Munich Workshop on Shannon Coding Techniques

TUM, April 5, 2024

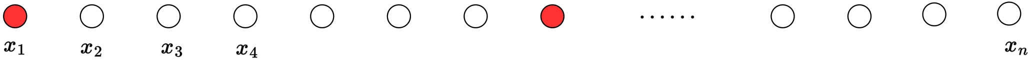
Background: Group Testing

- We have a large population of items



Background: Group Testing

- We have a large population of items
- Very few of them are "defective" (probability of being defective, γ is very small)

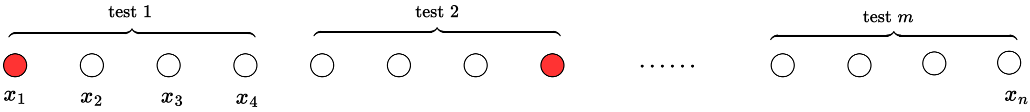


- **Goal:** Identify $\boldsymbol{x}$: defective $(x_i = 1)$, non-defective $(x_i = 0)$

[Dorfman1943] Robert Dorfman, “The Detection of Defective Members of Large Populations,” *The Annals of Mathematical Statistics,*, vol. 14, no. 4, pp. 436–440, 1943.

Background: Group Testing

- We have a large population of items
- Very few of them are "defective" (probability of being defective, γ is very small)

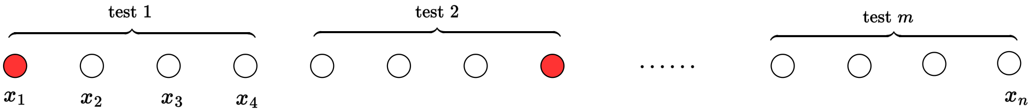


- **Goal:** Identify $\boldsymbol{x}$: defective $(x_i = 1)$, non-defective $(x_i = 0)$
- To reduce the number of tests: **test the items in groups** (pooling) [Dorfman1943]
- Rate, $\Omega = \frac{m}{n}$ (smaller is better)

[Dorfman1943] Robert Dorfman, “The Detection of Defective Members of Large Populations,” *The Annals of Mathematical Statistics,*, vol. 14, no. 4, pp. 436–440, 1943.

Background: Group Testing

- We have a large population of items
- Very few of them are "defective" (probability of being defective, γ is very small)

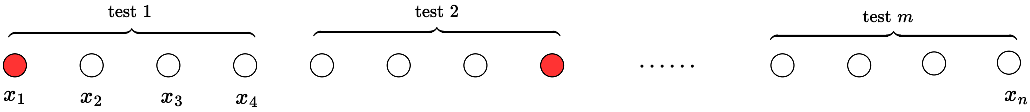


- **Goal:** Identify $\boldsymbol{x}$: defective $(x_i = 1)$, non-defective $(x_i = 0)$
- To reduce the number of tests: **test the items in groups** (pooling) [Dorfman1943]
- Rate, $\Omega = \frac{m}{n}$ (smaller is better)
- Adaptive vs non-adaptive test design

[Dorfman1943] Robert Dorfman, “The Detection of Defective Members of Large Populations,” *The Annals of Mathematical Statistics*,, vol. 14, no. 4, pp. 436–440, 1943.

Background: Group Testing

- We have a large population of items
- Very few of them are "defective" (probability of being defective, γ is very small)



- **Goal:** Identify $\boldsymbol{x}$: defective $(x_i = 1)$, non-defective $(x_i = 0)$
- To reduce the number of tests: **test the items in groups** (pooling) [Dorfman1943]
- Rate, $\Omega = \frac{m}{n}$ (smaller is better)
- Adaptive vs non-adaptive test design
- We consider the asymptotic regime: $n \to \infty$

[Dorfman1943] Robert Dorfman, “The Detection of Defective Members of Large Populations,” *The Annals of Mathematical Statistics,*, vol. 14, no. 4, pp. 436–440, 1943.

Background: Graphical Representation

- For non-adaptive group testing the pooling can be represented by a test matrix $\boldsymbol{A}$

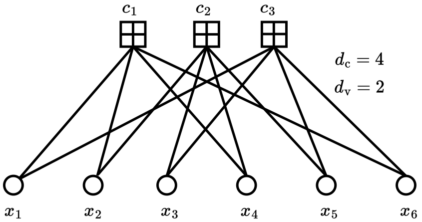
$$\boldsymbol{A} = \begin{array}{c} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \\ \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \end{array}$$

Background: Graphical Representation

- For non-adaptive group testing the pooling can be represented by a test matrix $\boldsymbol{A}$

$$\boldsymbol{A} = \begin{array}{c} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \\ \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \end{array}$$

- The matrix can be represented by a bipartite graph G



- We consider the scenario where the graph is sparse

Non-quantitative vs Quantitative

- **Non-quantitative:** test result, $s_i = 1$ if at least one item is defective otherwise $s_i = 0$ (logical OR)

$$\begin{array}{ccc} \begin{array}{cccccc} \bullet & \bullet & \circ & \circ & \circ & \bullet \end{array} & & \text{Non-Quantitative} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} & \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \end{array} & \begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \end{array} \end{array}$$

Non-quantitative vs Quantitative

- **Non-quantitative:** test result, $s_i = 1$ if at least one item is defective otherwise $s_i = 0$ (logical OR)

$$\begin{array}{cccccc|ccc}
\bullet & \bullet & \circ & \circ & \circ & \bullet & & \text{Non-Quantitative} & \text{Quantitative} \\
0 & 0 & 1 & 1 & 1 & 0 & s_1 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & s_2 & 1 & 2 \\
0 & 1 & 1 & 0 & 1 & 0 & s_3 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & s_4 & 1 & 3
\end{array}$$

- For **quantitative** group testing, a test result shows the number of defective items

$$s_i = \sum_{j=1}^{n} x_j a_{ij} \quad \rightarrow \quad \boldsymbol{s} = Ax$$

Quantitative Group Testing with Sparse Graphs: Prior work

- The test results show the number of defectives
- Best known scheme with sparse graph uses GLDPC [KAR2019]

$$\boldsymbol{U} = \begin{pmatrix} \overbrace{\begin{matrix}1 & 1 & 1 & 1 & 1 & 1 & 1\end{matrix}}^{d_{\mathsf{c}}} \\ \begin{matrix}0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1\end{matrix} \end{pmatrix} \quad t = 1, m_{\mathsf{u}} = 3$$

- A t-error-correcting BCH code is used as a component code
- An additional row of ones to identify # of defective items

[KAR2019] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, “Sparse graph codes for non-adaptive quantitative group testing,” in*Proc. IEEE Inf. Theory Workshop (ITW)*, 2019.

Quantitative Group Testing with Sparse Graphs: Prior work

- The test results show the number of defectives
- Best known scheme with sparse graph uses GLDPC [KAR2019]

$$\boldsymbol{U} = \begin{pmatrix} \multicolumn{7}{c}{\overbrace{\hphantom{1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1}}^{d_{\mathsf{c}}}} \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \quad t = 1, m_{\mathsf{u}} = 3$$

- $d_{\mathsf{c}} = 2^{m_{\mathsf{u}}} - 1 \rightarrow m_{\mathsf{u}} = \log_2(d_{\mathsf{c}} + 1)$
- Tests per subcode $= t \log_2(d_{\mathsf{c}} + 1) + 1$

- A t-error-correcting BCH code is used as a component code
- An additional row of ones to identify # of defective items

[KAR2019] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, “Sparse graph codes for non-adaptive quantitative group testing,” in*Proc. IEEE Inf. Theory Workshop (ITW)*, 2019.

Quantitative Group Testing with Sparse Graphs: Prior work

- The test results show the number of defectives
- Best known scheme with sparse graph uses GLDPC [KAR2019]

$$\boldsymbol{U} = \begin{pmatrix} \multicolumn{7}{c}{\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad}^{d_{\mathsf{c}}}} \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \quad t = 1, m_{\mathsf{u}} = 3$$

- $d_{\mathsf{c}} = 2^{m_{\mathsf{u}}} - 1 \rightarrow m_{\mathsf{u}} = \log_2(d_{\mathsf{c}} + 1)$
- Tests per subcode $= t \log_2(d_{\mathsf{c}} + 1) + 1$
- Rate, $\Omega = \frac{m}{n} = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}} \left(t \lceil \log_2(d_{\mathsf{c}} + 1) \rceil + 1 \right)$

- A t-error-correcting BCH code is used as a component code
- An additional row of ones to identify # of defective items

[KAR2019] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, “Sparse graph codes for non-adaptive quantitative group testing,” in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2019.

Prior Work

- **Density Evolution**

  For each iteration ℓ

  $q^{(\ell)}$: probability a test sends resolved to item
  $p^{(\ell)}$: probability a defective item is unresolved

  Test to item:

  $$q^{(\ell)} = \sum_{i=0}^{t-1} \binom{d_{\mathsf{c}}-1}{i} \left(p^{(\ell-1)}\right)^{i} \left(1-p^{(\ell-1)}\right)^{d_{\mathsf{c}}-1-i}$$

  Item to test:

  $$p^{(\ell)} = \gamma(1-q^{(\ell-1)})^{d_{\mathsf{v}}-1}$$

Prior Work

- **Density Evolution**

  For each iteration ℓ

  $q^{(\ell)}$: probability a test sends resolved to item
  $p^{(\ell)}$: probability a defective item is unresolved

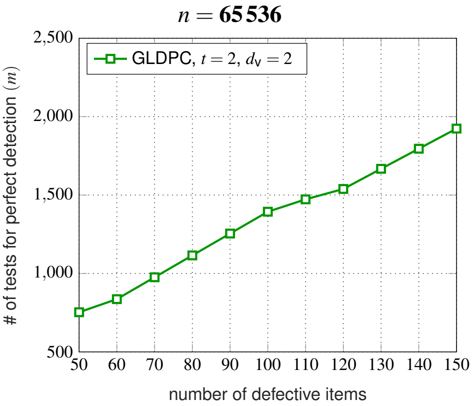
  Test to item:

  $$q^{(\ell)} = \sum_{i=0}^{t-1} \binom{d_\mathsf{c}-1}{i} \left(p^{(\ell-1)}\right)^{i} \left(1-p^{(\ell-1)}\right)^{d_\mathsf{c}-1-i}$$

  Item to test:

  $$p^{(\ell)} = \gamma(1-q^{(\ell-1)})^{d_\mathsf{v}-1}$$

- Small number of tests for a large population size



Prior Work

- **Density Evolution**

  For each iteration ℓ

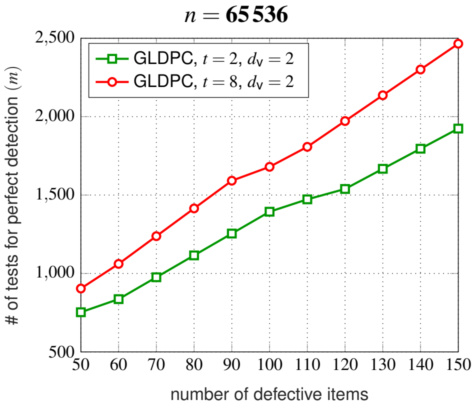
  $q^{(\ell)}$: probability a test sends resolved to item
  $p^{(\ell)}$: probability a defective item is unresolved

  Test to item:

  $$q^{(\ell)} = \sum_{i=0}^{t-1} \binom{d_{\mathsf{c}}-1}{i} \left(p^{(\ell-1)}\right)^{i} \left(1-p^{(\ell-1)}\right)^{d_{\mathsf{c}}-1-i}$$

  Item to test:

  $$p^{(\ell)} = \gamma(1-q^{(\ell-1)})^{d_{\mathsf{v}}-1}$$



- Small number of tests for a large population size
- Increasing t improves error correction
- Penalized by increasing number of tests
  $m = n\frac{d_{\mathsf{v}}}{d_{\mathsf{c}}}\left(t\lceil\log_2(d_{\mathsf{c}}+1)\rceil+1\right)$

Prior Work

- **Density Evolution**

  For each iteration ℓ

  $q^{(\ell)}$: probability a test sends resolved to item
  $p^{(\ell)}$: probability a defective item is unresolved

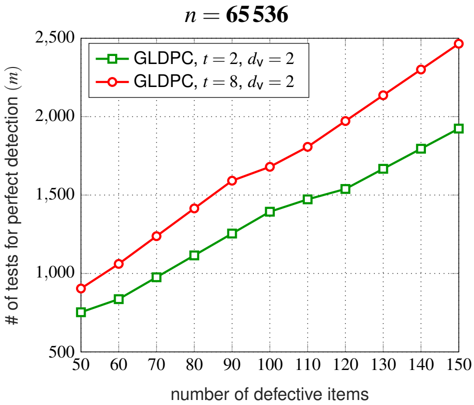
  Test to item:

  $$q^{(\ell)} = \sum_{i=0}^{t-1} \binom{d_\mathsf{c}-1}{i} \left(p^{(\ell-1)}\right)^i \left(1-p^{(\ell-1)}\right)^{d_\mathsf{c}-1-i}$$

  Item to test:

  $$p^{(\ell)} = \gamma(1-q^{(\ell-1)})^{d_\mathsf{v}-1}$$

- Small number of tests for a large population size
- Increasing t improves error correction
- Penalized by increasing number of tests
  $m = n\frac{d_\mathsf{v}}{d_\mathsf{c}}\left(t\lceil\log_2(d_\mathsf{c}+1)\rceil+1\right)$



- What about using $t=0$?

Proposed scheme: Group Testing with LDPC

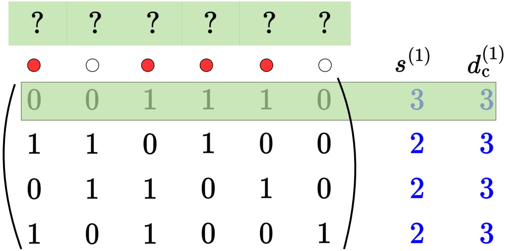
- With $t = 0$ we loose local error correcting capability

Proposed scheme: Group Testing with LDPC

- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_{\mathsf{c}}^{(\ell)}$
    Infer all items as 1 (defective)

Proposed scheme: Group Testing with LDPC

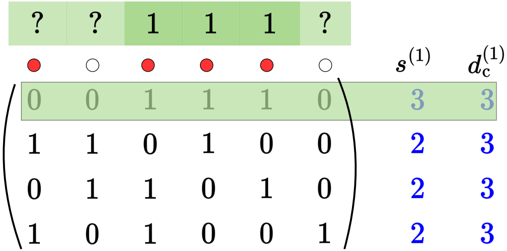
- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_{\mathrm{c}}^{(\ell)}$
    Infer all items as 1 (defective)



$$\begin{array}{cccccccc}
? & ? & ? & ? & ? & ? & & \\
\bullet & \circ & \bullet & \bullet & \bullet & \circ & \boldsymbol{s}^{(1)} & \boldsymbol{d}_{\mathrm{c}}^{(1)} \\
0 & 0 & 1 & 1 & 1 & 0 & 3 & 3 \\
1 & 1 & 0 & 1 & 0 & 0 & 2 & 3 \\
0 & 1 & 1 & 0 & 1 & 0 & 2 & 3 \\
1 & 0 & 1 & 0 & 0 & 1 & 2 & 3
\end{array}$$

Proposed scheme: Group Testing with LDPC

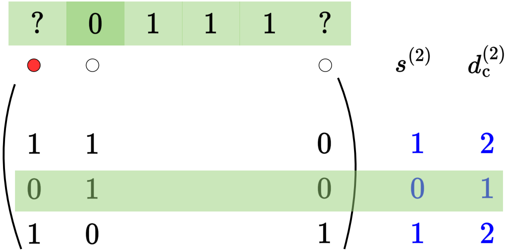
- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_{\mathrm{c}}^{(\ell)}$
    Infer all items as 1 (defective)
- We then peel off *resolved* items (reducing the syndrome accordingly)



$$\begin{array}{cccccccc} ? & ? & 1 & 1 & 1 & ? & & \\ \bullet & \circ & \bullet & \bullet & \bullet & \circ & s^{(1)} & d_{\mathrm{c}}^{(1)} \\ 0 & 0 & 1 & 1 & 1 & 0 & 3 & 3 \\ 1 & 1 & 0 & 1 & 0 & 0 & 2 & 3 \\ 0 & 1 & 1 & 0 & 1 & 0 & 2 & 3 \\ 1 & 0 & 1 & 0 & 0 & 1 & 2 & 3 \end{array}$$

Proposed scheme: Group Testing with LDPC

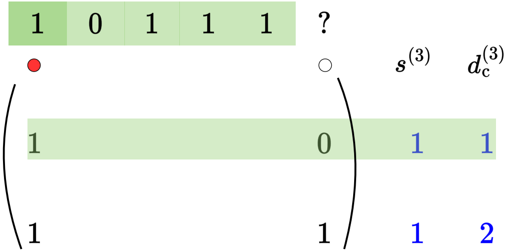
- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_\mathrm{c}^{(\ell)}$
    Infer all items as 1 (defective)
- We then peel off *resolved* items (reducing the syndrome accordingly)
- This is repeated until no item to peel



$$\begin{array}{cccccccc} ? & 0 & 1 & 1 & 1 & ? & & \\ \bullet & \circ & & & & \circ & s^{(2)} & d_\mathrm{c}^{(2)} \\ 1 & 1 & & & & 0 & 1 & 2 \\ 0 & 1 & & & & 0 & 0 & 1 \\ 1 & 0 & & & & 1 & 1 & 2 \end{array}$$

Proposed scheme: Group Testing with LDPC

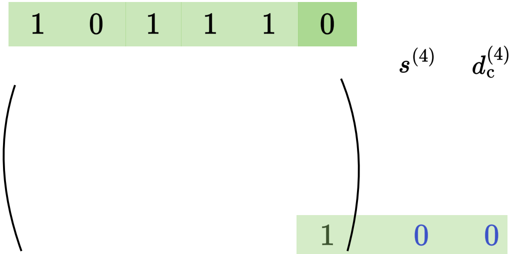
- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_\mathrm{c}^{(\ell)}$
    Infer all items as 1 (defective)
- We then peel off *resolved* items (reducing the syndrome accordingly)
- This is repeated until no item to peel



$$\begin{array}{cccccccc} 1 & 0 & 1 & 1 & 1 & ? & & \\ \bullet & & & & & \circ & s^{(3)} & d_\mathrm{c}^{(3)} \\ 1 & & & & & 0 & 1 & 1 \\ 1 & & & & & 1 & 1 & 2 \end{array}$$

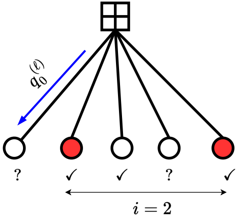
Proposed scheme: Group Testing with LDPC

- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
    - Syndrome equal zero: $s_i^{(\ell)} = 0$
      Infer all items as 0 (non-defective)
    - Syndrome equals test degree: $s_i^{(\ell)} = d_{\mathrm{c}}^{(\ell)}$
      Infer all items as 1 (defective)
- We then peel off *resolved* items (reducing the syndrome accordingly)
- This is repeated until no item to peel



Density Evolution

- $p_1^{(\ell)}$: probability that a message from a defective is *unresolved*
- $q_0^{(\ell)}$: **probability that a message to a non-defective is *resolved***
- $p_0^{(\ell)}$: probability a message from non-defective is *unresolved*
- $q_1^{(\ell)}$: probability that a message to a defective is *resolved*



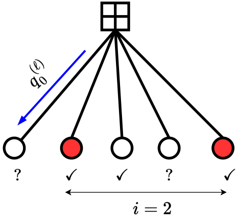
From test to item

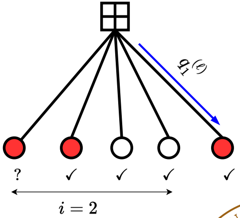
$$q_0^{(\ell)} = \sum_{i=0}^{d_\mathsf{c}-1} \binom{d_\mathsf{c}-1}{i} \gamma^i (1-\gamma)^{d_\mathsf{c}-1-i} \left(1-p_1^{(\ell-1)}\right)^i$$

$$q_1^{(\ell)} = \sum_{i=0}^{d_\mathsf{c}-1} \binom{d_\mathsf{c}-1}{i} \gamma^i (1-\gamma)^{d_\mathsf{c}-1-i} \left(1-p_0^{(\ell-1)}\right)^{d_\mathsf{c}-1-i}$$

Density Evolution

- $p_1^{(\ell)}$: probability that a message from a defective is *unresolved*
- $q_0^{(\ell)}$: **probability that a message to a non-defective is *resolved***
- $p_0^{(\ell)}$: probability a message from non-defective is *unresolved*
- $q_1^{(\ell)}$: **probability that a message to a defective is *resolved***





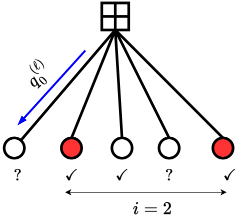
From test to item

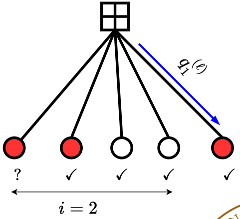
$$q_0^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c-1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1 - p_1^{(\ell-1)}\right)^i$$

$$q_1^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c-1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1 - p_0^{(\ell-1)}\right)^{d_c-1-i}$$

Density Evolution

- $p_1^{(\ell)}$: **probability that a message from a defective is *unresolved***
- $q_0^{(\ell)}$: **probability that a message to a non-defective is *resolved***
- $p_0^{(\ell)}$: **probability a message from non-defective is *unresolved***
- $q_1^{(\ell)}$: **probability that a message to a defective is *resolved***





From test to item

$$q_0^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c - 1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1 - p_1^{(\ell-1)}\right)^i$$

$$q_1^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c - 1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1 - p_0^{(\ell-1)}\right)^{d_c-1-i}$$

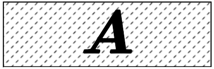
From item to test

$$p_0^{(\ell)} = \left(1 - q_0^{(\ell-1)}\right)^{d_v-1}$$

$$p_1^{(\ell)} = \left(1 - q_1^{(\ell-1)}\right)^{d_v-1} .$$

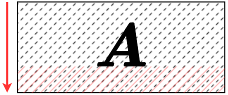
Performance Comparison

- We consider two scenarios
  - **Fixing the proportion of defective items γ and changing the rate $\Omega = \frac{m}{n}$**



Performance Comparison

- We consider two scenarios
  - **Fixing the proportion of defective items γ and changing the rate $\Omega = \frac{m}{n}$**

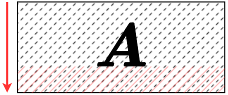


  - Same as in previous work [KAR2019]

[KAR2019] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, "Sparse graph codes for non-adaptive quantitative group testing," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2019.

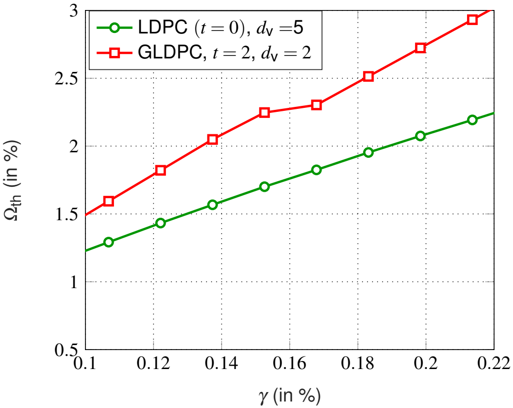
Performance Comparison

- We consider two scenarios
  - **Fixing the proportion of defective items γ and changing the rate $\Omega = \frac{m}{n}$**



  - Same as in previous work [KAR2019]

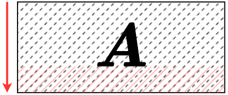
Minimum rate required for a fixed γ



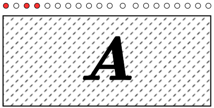
[KAR2019] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, “Sparse graph codes for non-adaptive quantitative group testing,” in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2019.

Performance Comparison

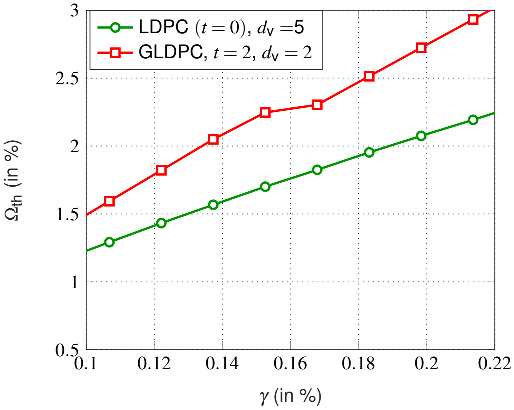
- We consider two scenarios
  - **Fixing the proportion of defective items γ and changing the rate $\Omega = \frac{m}{n}$**



  - Same as in previous work [KAR2019]
  - **Fixing the rate Ω and changing γ**

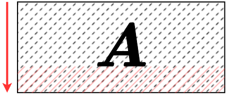


Minimum rate required for a fixed γ

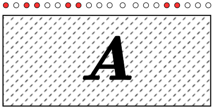


Performance Comparison

- We consider two scenarios
  - **Fixing the proportion of defective items γ and changing the rate $\Omega = \frac{m}{n}$**

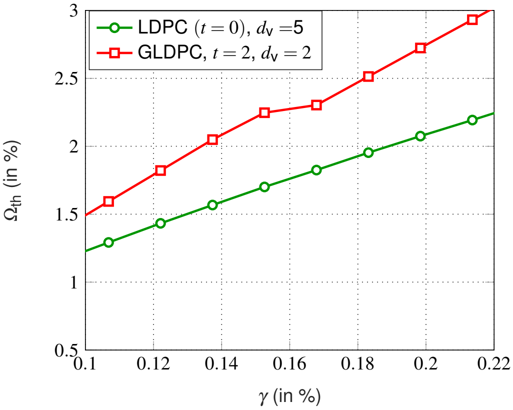


  - Same as in previous work [KAR2019]
  - **Fixing the rate Ω and changing γ**



  - A new perspective considering $\boldsymbol{A}$ (code) as fixed

Minimum rate required for a fixed γ



[KAR2019] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, “Sparse graph codes for non-adaptive quantitative group testing,” in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2019.

Performance Comparison: Fixed Rate, $\Omega = 5\%$

Table: GLDPC Based

t	d_v	γ_th
1	2	0.2487
	3	0.3708
	4	0.3510
2	2	0.3983
	3	0.3372
	4	0.2884
3	2	0.3784
	3	0.3189
	4	0.2441
5	2	0.3418
	3	0.2686
	4	0.2014

Performance Comparison: Fixed Rate, $\Omega = 5\%$

Table: GLDPC Based

t	d_v	γ_th
1	2	0.2487
	3	0.3708
	4	0.3510
2	2	0.3983
	3	0.3372
	4	0.2884
3	2	0.3784
	3	0.3189
	4	0.2441
5	2	0.3418
	3	0.2686
	4	0.2014

Table: LDPC Based

d_v	γ_th
3	0.4555
4	0.5982
5	0.6416
6	0.6464
7	0.6353
10	0.5773

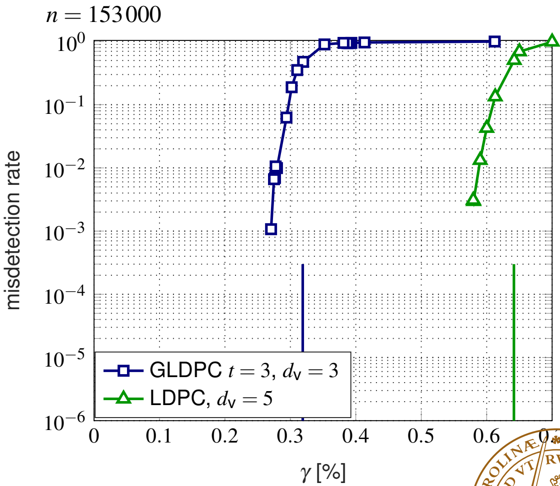
Performance Comparison: Fixed Rate, $\Omega = 5\%$

Table: GLDPC Based

t	d_v	γ_th
1	2	0.2487
	3	0.3708
	4	0.3510
2	2	0.3983
	3	0.3372
	4	0.2884
3	2	0.3784
	3	0.3189
	4	0.2441
5	2	0.3418
	3	0.2686
	4	0.2014

Table: LDPC Based

d_v	γ_th
3	0.4555
4	0.5982
5	0.6416
6	0.6464
7	0.6353
10	0.5773



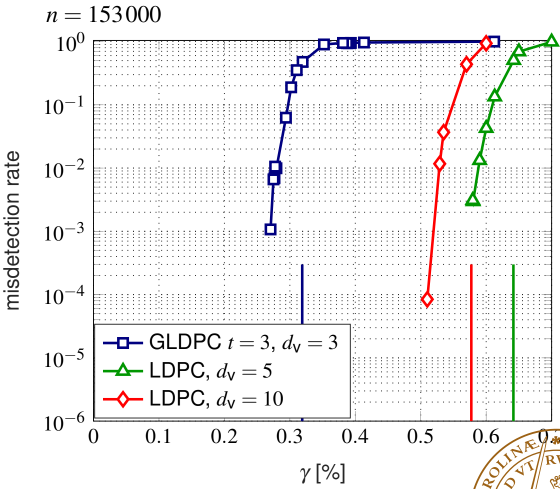
Performance Comparison: Fixed Rate, $\Omega = 5\%$

Table: GLDPC Based

t	d_v	γ_th
1	2	0.2487
	3	0.3708
	4	0.3510
2	2	0.3983
	3	0.3372
	4	0.2884
3	2	0.3784
	3	0.3189
	4	0.2441
5	2	0.3418
	3	0.2686
	4	0.2014

Table: LDPC Based

d_v	γ_th
3	0.4555
4	0.5982
5	0.6416
6	0.6464
7	0.6353
10	0.5773



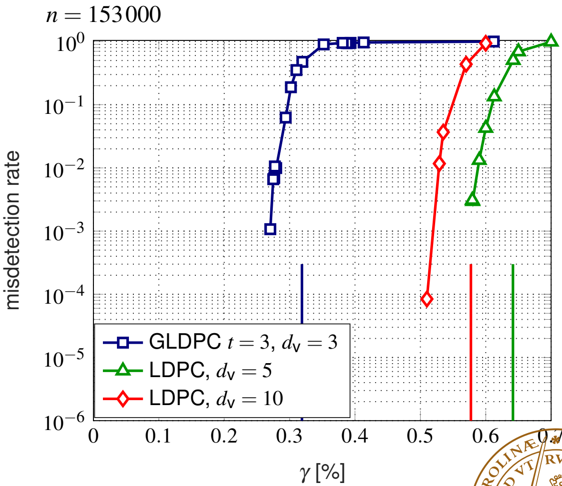
Performance Comparison: Fixed Rate, $\Omega = 5\%$

Table: GLDPC Based

t	d_{v}	γ_{th}
1	2	0.2487
	3	0.3708
	4	0.3510
2	2	0.3983
	3	0.3372
	4	0.2884
3	2	0.3784
	3	0.3189
	4	0.2441
5	2	0.3418
	3	0.2686
	4	0.2014

Table: LDPC Based

d_{v}	γ_{th}
3	0.4555
4	0.5982
5	0.6416
6	0.6464
7	0.6353
10	0.5773



- Can we get consistently better with increasing d_{v}? What about spatial coupling?

Improved Decoding with a Non-Binary Alphabet

- **Idea:** for a subset of tests, items occur only in bundles of size q

Improved Decoding with a Non-Binary Alphabet

- **Idea:** for a subset of tests, items occur only in bundles of size q

Example: $d_\mathsf{v} = 3$, $d_\mathsf{c} = 4$

$$\boldsymbol{A} = \begin{array}{cccccccc} \multicolumn{2}{c}{z_1} & \multicolumn{2}{c}{z_2} & \multicolumn{2}{c}{z_3} & \multicolumn{2}{c}{z_4} \\ \end{array}$$

$$\boldsymbol{A} = \left[\begin{array}{cccc} \boxed{1 \quad 1} & 0 \quad 0 & 0 \quad 0 & \boxed{1 \quad 1} \\ 0 \quad 0 & \boxed{1 \quad 1} & \boxed{1 \quad 1} & 0 \quad 0 \\ \boxed{1 \quad 1} & 0 \quad 0 & \boxed{1 \quad 1} & 0 \quad 0 \\ 0 \quad 0 & \boxed{1 \quad 1} & 0 \quad 0 & \boxed{1 \quad 1} \\ \hline 1 \quad 0 & 1 \quad 0 & 1 \quad 0 & 1 \quad 0 \\ 0 \quad 1 & 0 \quad 1 & 0 \quad 1 & 0 \quad 1 \end{array}\right] \qquad q = 2$$

$$x_1 \; x_2 \; x_3 \; x_4 \; x_5 \; x_6 \; x_7 \; x_8$$

Improved Decoding with a Non-Binary Alphabet

- **Idea:** for a subset of tests, items occur only in bundles of size q

  Example: $d_{\mathsf{v}} = 3$, $d_{\mathsf{c}} = 4$

$$\begin{array}{cc}
 & \begin{array}{cccc} z_1 & z_2 & z_3 & z_4 \end{array} \\
\boldsymbol{A} = & \left[\begin{array}{cccccccc}
\multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 & 0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} \\
0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 \\
\multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 \\
0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{array}\right] \\
 & \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{array}
\end{array}$$

$$q = 2$$

- Value of a bundle $z \in \{0, \ldots, q\}$: sum of included items

Improved Decoding with a Non-Binary Alphabet

- **Idea:** for a subset of tests, items occur only in bundles of size q

Example: $d_{\mathsf{v}} = 3$, $d_{\mathsf{c}} = 4$

$$\begin{array}{c|cccccccc|}
 & \multicolumn{2}{c}{z_1} & \multicolumn{2}{c}{z_2} & \multicolumn{2}{c}{z_3} & \multicolumn{2}{c}{z_4} \\
 & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 & 0 & 0 & \multicolumn{2}{c|}{\boxed{1 \quad 1}} \\
 & 0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 \\
\boldsymbol{A} = & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 \\
 & 0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 & \multicolumn{2}{c|}{\boxed{1 \quad 1}} \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8
\end{array} \qquad q = 2$$

- Value of a bundle $z \in \{0, \ldots, q\}$: sum of included items
- Compatible with standard testing: only test matrix structure affected

Improved Decoding with a Non-Binary Alphabet

- **Idea:** for a subset of tests, items occur only in bundles of size q

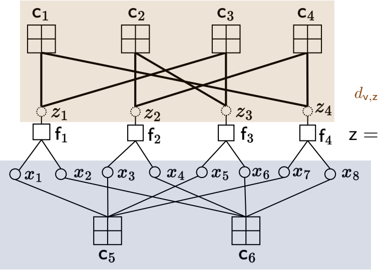
Example: $d_{\mathsf{v}} = 3$, $d_{\mathsf{c}} = 4$

$$\boldsymbol{A} = \begin{array}{c} \begin{array}{cccc} z_1 & z_2 & z_3 & z_4 \end{array} \\ \left[\begin{array}{cccccccc} \boxed{1 \quad 1} & & 0 & 0 & 0 & 0 & \boxed{1 \quad 1} & \\ 0 & 0 & \boxed{1 \quad 1} & & \boxed{1 \quad 1} & & 0 & 0 \\ \boxed{1 \quad 1} & & 0 & 0 & \boxed{1 \quad 1} & & 0 & 0 \\ 0 & 0 & \boxed{1 \quad 1} & & 0 & 0 & \boxed{1 \quad 1} & \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right] \\ \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{array} \end{array} \qquad q = 2$$

- Value of a bundle $z \in \{0, \ldots, q\}$: sum of included items
- Compatible with standard testing: only test matrix structure affected

Factor graph representation:

$\mathcal{CN}_{\mathsf{z}} = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$



$d_{\mathsf{v,z}} = 2$

$\mathsf{z} = \mathsf{f}(\boldsymbol{x}) = \sum_{i \in \mathcal{T}(\mathsf{f})} x_i$

$d_{\mathsf{v,x}} = 1$

$d_{\mathsf{v}} = d_{\mathsf{v,z}} + d_{\mathsf{v,x}}$

$\mathcal{CN}_{\mathsf{x}} = \{\mathsf{c}_5, \mathsf{c}_6\}$

Local Decoding with q-ary Variables

Extension of the erasure decoder to $q > 1$:

- messages: $\mu \in \{0, q, ?\}$
- Problem: can still only resolve $s = 0$ and $s = d_{\mathsf{c}}$, no gain with q

Local Decoding with q-ary Variables

Extension of the erasure decoder to $q > 1$:

- messages: $\mu \in \{0, q, ?\}$
- Problem: can still only resolve $s = 0$ and $s = d_{\mathsf{c}}$, no gain with q

APP decoding (SISO):

- messages are probability vectors $\mu = [P(z = 0), P(z = 1), \ldots, P(z = q)]$, computed in a trellis
- Problem: complexity grows rapidly with degree d_{c} (even for $q = 1$)

Local Decoding with q-ary Variables

Extension of the erasure decoder to $q > 1$:

- messages: $\mu \in \{0, q, ?\}$
- Problem: can still only resolve $s = 0$ and $s = d_\text{c}$, no gain with q

APP decoding (SISO):

- messages are probability vectors $\mu = [P(z=0), P(z=1), \ldots, P(z=q)]$, computed in a trellis
- Problem: complexity grows rapidly with degree d_c (even for $q = 1$)

Proposed decoder: motivated by works on counter braids [LM+2008][RG2018]

- messages $\mu = [L, U]$ consist of lower bound L and upper bound U on $z \in \{0, \ldots, q\}$
- complexity similar to erasure decoding, performance improves with larger q

---

[LM+2008] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani, "Counter braids: A novel counter architecture for per-flow measurement," *Int. Conf. Meas. Modeling Comput. Syst. (SIGMETRICS),* Annapolis, June 2008.

[RG2018] E. Rosnes and A. Graell i Amat, "Asymptotic analysis and spatial coupling of counter braids," *IEEE Transactions on Information Theory,* vol. 64, no. 11, 2018.

Local Decoding with q-ary Variables

Extension of the erasure decoder to $q > 1$:

- messages: $\mu \in \{0, q, ?\}$
- Problem: can still only resolve $s = 0$ and $s = d_\mathrm{c}$, no gain with q

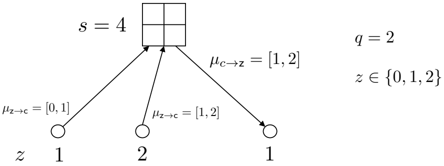
APP decoding (SISO):

- messages are probability vectors $\mu = [P(z = 0), P(z = 1), \ldots, P(z = q)]$, computed in a trellis
- Problem: complexity grows rapidly with degree d_c (even for $q = 1$)

Proposed decoder: motivated by works on counter braids [LM+2008][RG2018]

- messages $\mu = [L, U]$ consist of lower bound L and upper bound U on $z \in \{0, \ldots, q\}$
- complexity similar to erasure decoding, performance improves with larger q

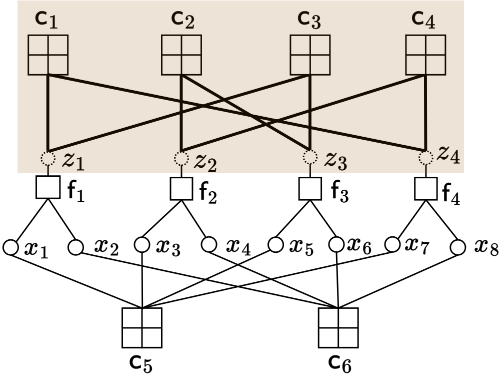
Example:



[LM+2008] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani, "Counter braids: A novel counter architecture for per-flow measurement," *Int. Conf. Meas. Modeling Comput. Syst. (SIGMETRICS)*, Annapolis, June 2008.
[RG2018] E. Rosnes and A. Graell i Amat, "Asymptotic analysis and spatial coupling of counter braids," *IEEE Transactions on Information Theory*, vol. 64, no. 11, 2018.

Message Passing between Bundles and Tests

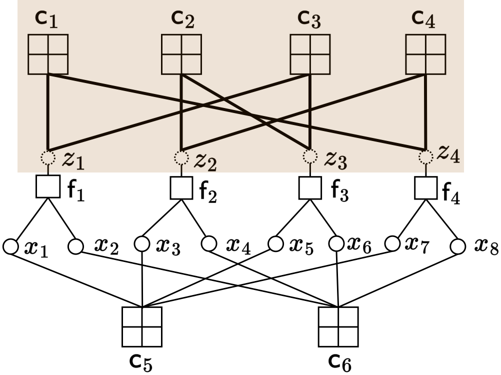
$$\mathcal{CN}_{\mathsf{z}} = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$$



$$\mathcal{CN}_{\mathsf{x}} = \{\mathsf{c}_5, \mathsf{c}_6\}$$

Message Passing between Bundles and Tests

$$\mathcal{CN}_{\mathsf{z}} = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$$



$$\mathcal{CN}_{\mathsf{x}} = \{\mathsf{c}_5, \mathsf{c}_6\}$$

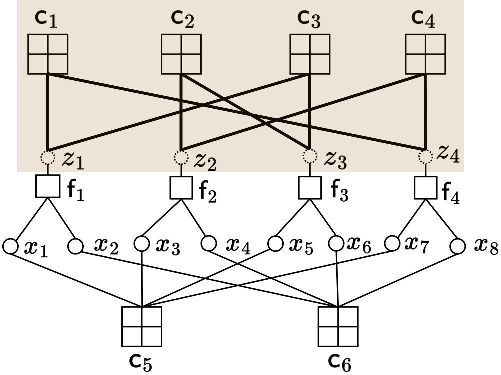
Test to bundle:

$$\mathsf{L}^{(\ell)}_{\mathsf{c}\to\mathsf{z}} = \max\left\{ s(\mathsf{c}) - \sum_{\mathsf{z}'\in\mathcal{T}(\mathsf{c})\setminus\mathsf{z}} \mathsf{U}^{(\ell-1)}_{\mathsf{z}'\to\mathsf{c}}, 0 \right\}$$

$$\mathsf{U}^{(\ell)}_{\mathsf{c}\to\mathsf{z}} = \min\left\{ s(\mathsf{c}) - \sum_{\mathsf{z}'\in\mathcal{T}(\mathsf{c})\setminus\mathsf{z}} \mathsf{L}^{(\ell-1)}_{\mathsf{z}'\to\mathsf{c}}, q \right\},$$

Message Passing between Bundles and Tests

$$\mathcal{CN}_{\mathsf{z}} = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$$



$$\mathcal{CN}_{\mathsf{x}} = \{\mathsf{c}_5, \mathsf{c}_6\}$$

Test to bundle:

$$\mathsf{L}^{(\ell)}_{\mathsf{c}\to\mathsf{z}} = \max\left\{ s(\mathsf{c}) - \sum_{\mathsf{z}'\in\mathcal{T}(\mathsf{c})\setminus\mathsf{z}} \mathsf{U}^{(\ell-1)}_{\mathsf{z}'\to\mathsf{c}}, 0 \right\}$$

$$\mathsf{U}^{(\ell)}_{\mathsf{c}\to\mathsf{z}} = \min\left\{ s(\mathsf{c}) - \sum_{\mathsf{z}'\in\mathcal{T}(\mathsf{c})\setminus\mathsf{z}} \mathsf{L}^{(\ell-1)}_{\mathsf{z}'\to\mathsf{c}}, q \right\},$$

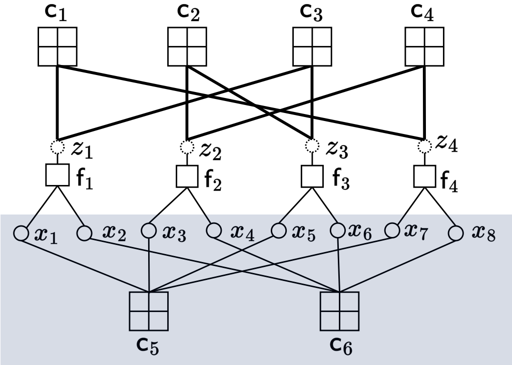
Bundle to test:

$$\mathsf{L}^{(\ell)}_{\mathsf{z}\to\mathsf{c}} = \max\left\{ \max_{\mathsf{c}'\in\mathcal{T}(\mathsf{z})\setminus\mathsf{c}} \mathsf{L}^{(\ell-1)}_{\mathsf{c}'\to\mathsf{z}},\ \mathsf{L}^{(\ell)}_{\mathsf{f}\to\mathsf{z}} \right\}$$

$$\mathsf{U}^{(\ell)}_{\mathsf{z}\to\mathsf{c}} = \min\left\{ \min_{\mathsf{c}'\in\mathcal{T}(\mathsf{z})\setminus\mathsf{c}} \mathsf{U}^{(\ell-1)}_{\mathsf{c}'\to\mathsf{z}},\ \mathsf{U}^{(\ell)}_{\mathsf{f}\to\mathsf{z}} \right\}.$$

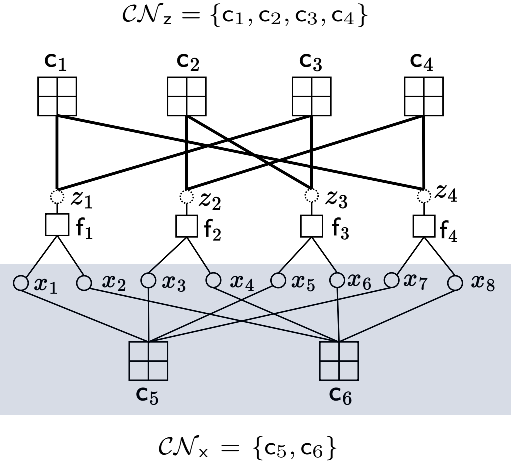
Message Passing between Items and Tests

$$\mathcal{CN}_{\mathsf{z}} = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$$



$$\mathcal{CN}_{\mathsf{x}} = \{\mathsf{c}_5, \mathsf{c}_6\}$$

Message Passing between Items and Tests



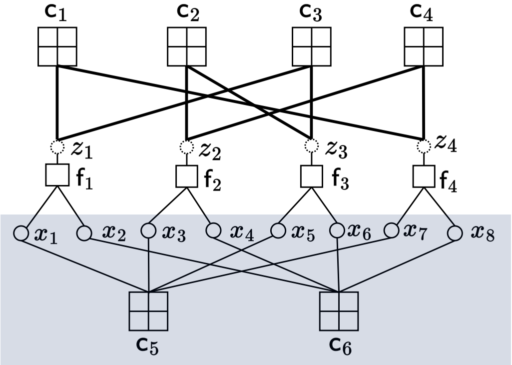
Test to item:

$$\mathsf{L}_{\mathsf{c}\to\mathsf{x}}^{(\ell)} = \max\left\{ s(\mathsf{c}) - \sum_{\mathsf{x}'\in\mathcal{T}(\mathsf{c})\backslash\mathsf{x}} \mathsf{U}_{\mathsf{x}'\to\mathsf{c}}^{(\ell-1)}, 0 \right\}$$

$$\mathsf{U}_{\mathsf{c}\to\mathsf{x}}^{(\ell)} = \min\left\{ s(\mathsf{c}) - \sum_{\mathsf{x}'\in\mathcal{T}(\mathsf{c})\backslash\mathsf{x}} L_{\mathsf{x}'\to\mathsf{c}}^{(\ell-1)}, 1 \right\}.$$

Message Passing between Items and Tests

$$\mathcal{CN}_{\mathsf{z}} = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$$



$$\mathcal{CN}_{\mathsf{x}} = \{\mathsf{c}_5, \mathsf{c}_6\}$$

Test to item:

$$\mathsf{L}^{(\ell)}_{\mathsf{c}\to\mathsf{x}} = \max\left\{ s(\mathsf{c}) - \sum_{\mathsf{x}'\in\mathcal{T}(\mathsf{c})\setminus\mathsf{x}} \mathsf{U}^{(\ell-1)}_{\mathsf{x}'\to\mathsf{c}}, 0 \right\}$$

$$\mathsf{U}^{(\ell)}_{\mathsf{c}\to\mathsf{x}} = \min\left\{ s(\mathsf{c}) - \sum_{\mathsf{x}'\in\mathcal{T}(\mathsf{c})\setminus\mathsf{x}} L^{(\ell-1)}_{\mathsf{x}'\to\mathsf{c}}, 1 \right\}.$$

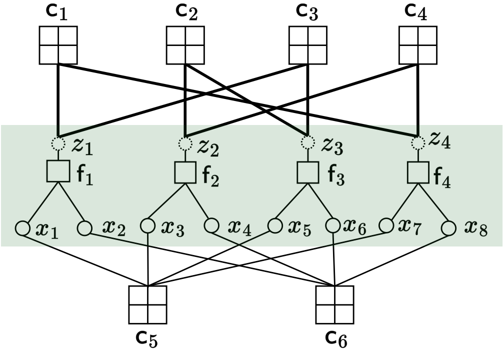
Item to test:

$$\mathsf{L}^{(\ell)}_{\mathsf{x}\to\mathsf{c}} = \max\left\{ \max_{\mathsf{c}'\in\mathcal{T}_s(\mathsf{x})\setminus\mathsf{c}} L^{(\ell-1)}_{\mathsf{c}'\to\mathsf{x}}, \mathsf{L}^{(\ell-1)}_{\mathsf{f}\to\mathsf{x}} \right\}$$

$$\mathsf{U}^{(\ell)}_{\mathsf{x}\to\mathsf{c}} = \min\left\{ \min_{\mathsf{c}'\in\mathcal{T}_s(\mathsf{x})\setminus\mathsf{c}} U^{(\ell-1)}_{\mathsf{c}'\to\mathsf{x}}, \mathsf{U}^{(\ell-1)}_{\mathsf{f}\to\mathsf{x}} \right\}.$$

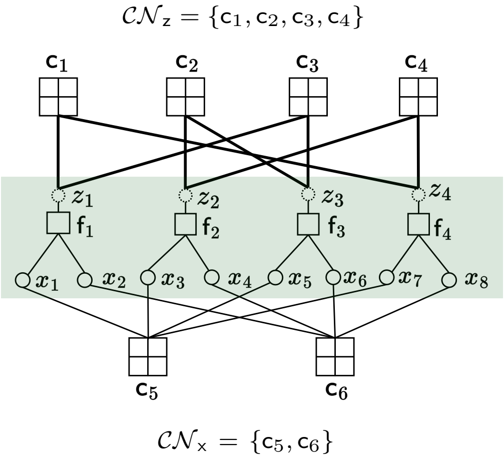
Message Passing between Bundles and Items

$$\mathcal{CN}_{\mathsf{z}} = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$$



$$\mathcal{CN}_{\mathsf{x}} = \{\mathsf{c}_5, \mathsf{c}_6\}$$

Message Passing between Bundles and Items



Bundle to item:

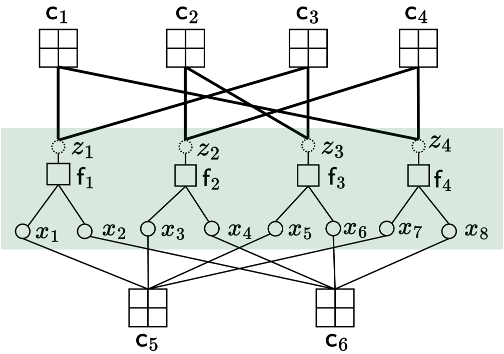
$$\mathsf{L}^{(\ell)}_{\mathsf{z}\to\mathsf{f}} = \max_{\mathsf{c}\in\mathcal{T}(\mathsf{z})} \mathsf{L}^{(\ell)}_{\mathsf{c}\to\mathsf{z}} \text{ and } \mathsf{U}^{(\ell)}_{\mathsf{z}\to\mathsf{f}} = \min_{\mathsf{c}\in\mathcal{T}(\mathsf{z})} \mathsf{U}^{(\ell)}_{\mathsf{c}\to\mathsf{z}} \,.$$

$$\mathsf{L}^{(\ell)}_{\mathsf{f}\to\mathsf{x}} = \max\left\{\mathsf{L}^{(\ell-1)}_{\mathsf{z}\to\mathsf{f}} - \sum_{\mathsf{x}'\in\mathcal{N}(\mathsf{f})\backslash\mathsf{x}} \mathsf{U}^{(\ell-1)}_{\mathsf{x}'\to\mathsf{f}}, 0\right\}$$

$$\mathsf{U}^{(\ell)}_{\mathsf{f}\to\mathsf{x}} = \min\left\{\mathsf{U}^{(\ell-1)}_{\mathsf{z}\to\mathsf{f}} - \sum_{\mathsf{x}'\in\mathcal{N}(\mathsf{f})\backslash\mathsf{x}} \mathsf{L}^{(\ell-1)}_{\mathsf{x}'\to\mathsf{f}}, 1\right\},$$

Message Passing between Bundles and Items

$$\mathcal{CN}_{\mathsf{z}} = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$$



$$\mathcal{CN}_{\mathsf{x}} = \{\mathsf{c}_5, \mathsf{c}_6\}$$

Bundle to item:

$$\mathsf{L}^{(\ell)}_{\mathsf{z}\to\mathsf{f}} = \max_{\mathsf{c}\in\mathcal{T}(\mathsf{z})} \mathsf{L}^{(\ell)}_{\mathsf{c}\to\mathsf{z}} \text{ and } \mathsf{U}^{(\ell)}_{\mathsf{z}\to\mathsf{f}} = \min_{\mathsf{c}\in\mathcal{T}(\mathsf{z})} \mathsf{U}^{(\ell)}_{\mathsf{c}\to\mathsf{z}}\,.$$

$$\mathsf{L}^{(\ell)}_{\mathsf{f}\to\mathsf{x}} = \max\left\{\mathsf{L}^{(\ell-1)}_{\mathsf{z}\to\mathsf{f}} - \sum_{\mathsf{x}'\in\mathcal{N}(\mathsf{f})\setminus\mathsf{x}} \mathsf{U}^{(\ell-1)}_{\mathsf{x}'\to\mathsf{f}}, 0\right\}$$

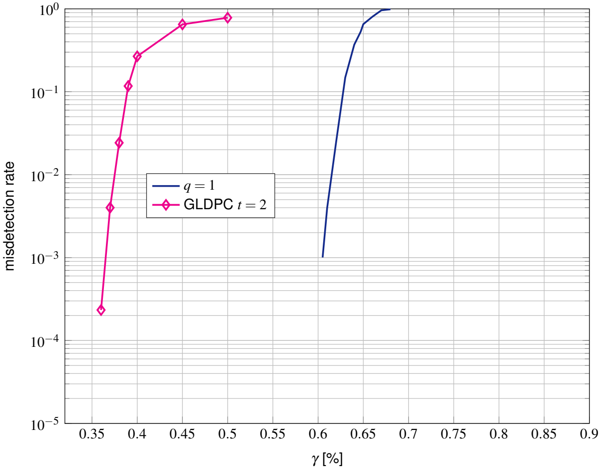
$$\mathsf{U}^{(\ell)}_{\mathsf{f}\to\mathsf{x}} = \min\left\{\mathsf{U}^{(\ell-1)}_{\mathsf{z}\to\mathsf{f}} - \sum_{\mathsf{x}'\in\mathcal{N}(\mathsf{f})\setminus\mathsf{x}} \mathsf{L}^{(\ell-1)}_{\mathsf{x}'\to\mathsf{f}}, 1\right\},$$

Item to bundle:

$$\mathsf{L}^{(\ell)}_{\mathsf{x}\to\mathsf{f}} = \max_{\mathsf{c}\in\mathcal{T}_s(\mathsf{x})} \mathsf{L}^{(\ell-1)}_{\mathsf{c}\to\mathsf{x}} \quad \text{and} \quad \mathsf{U}^{(\ell)}_{\mathsf{x}\to\mathsf{f}} = \min_{\mathsf{c}\in\mathcal{T}_s(\mathsf{x})} \mathsf{U}^{(\ell-1)}_{\mathsf{c}\to\mathsf{x}}\,.$$

$$\mathsf{L}^{(\ell)}_{\mathsf{f}\to\mathsf{z}} = \sum_{\mathsf{x}\in\mathcal{N}(\mathsf{f})} \mathsf{L}^{(\ell)}_{\mathsf{x}\to\mathsf{f}} \quad \text{and} \quad \mathsf{U}^{(\ell)}_{\mathsf{f}\to\mathsf{z}} = \sum_{\mathsf{x}\in\mathcal{N}(\mathsf{f})} \mathsf{U}^{(\ell)}_{\mathsf{x}\to\mathsf{f}}\,.$$

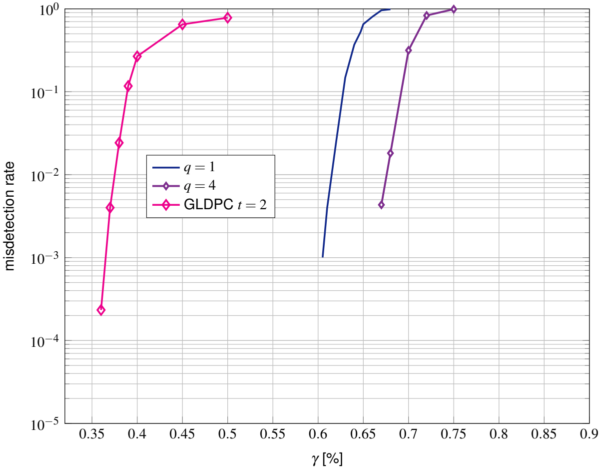
Performance Evaluation



Simulation results:

$n = 210\,000$ items
Fixed rate $\Omega = 5\,\%$, i.e., $m = 10\,500$ tests

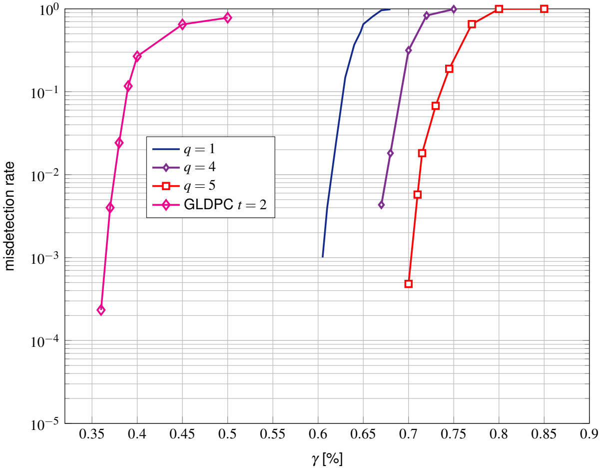
Performance Evaluation



Simulation results:

$n = 210\,000$ items
Fixed rate $\Omega = 5\,\%$, i.e., $m = 10\,500$ tests

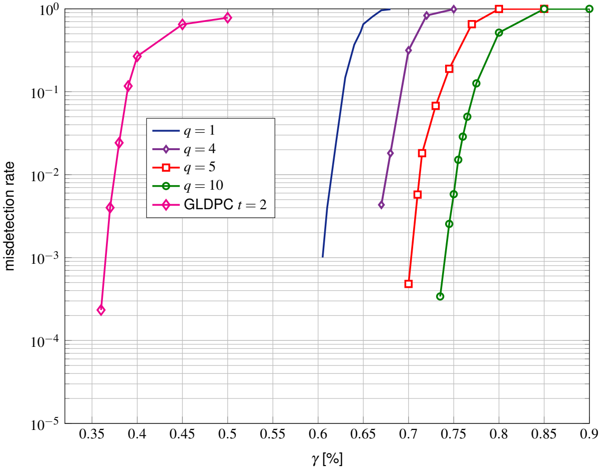
Performance Evaluation



Simulation results:

$n = 210000$ items
Fixed rate $\Omega = 5\,\%$, i.e., $m = 10500$ tests

Performance Evaluation

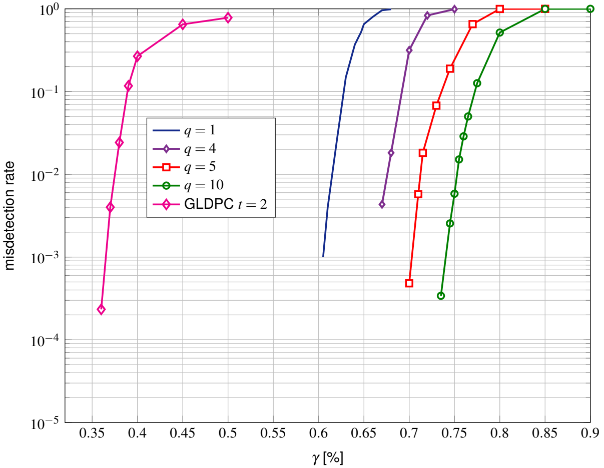


Simulation results:

$n = 210000$ items

Fixed rate $\Omega = 5\,\%$, i.e., $m = 10500$ tests

Performance Evaluation



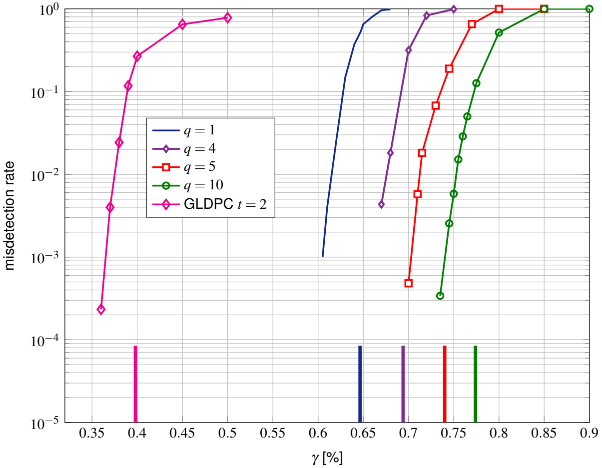
Simulation results:

$n = 210\,000$ items
Fixed rate $\Omega = 5\,\%$, i.e., $m = 10\,500$ tests

Density evolution thresholds: γ_{th}

q	$d_{\mathrm{v,x}}$	$d_{\mathrm{v}} = 4$	$d_{\mathrm{v}} = 5$	$d_{\mathrm{v}} = 6$	$d_{\mathrm{v}} = 7$	$d_{\mathrm{v}} = 8$
1		0.598	0.641	0.646	0.635	0.618
4	2	0.590	0.660	0.694	0.706	0.702
5	2	0.592	0.672	0.725	0.746	0.744
10	3	0.549	0.636	0.693	0.774	0.694

Performance Evaluation



Simulation results:

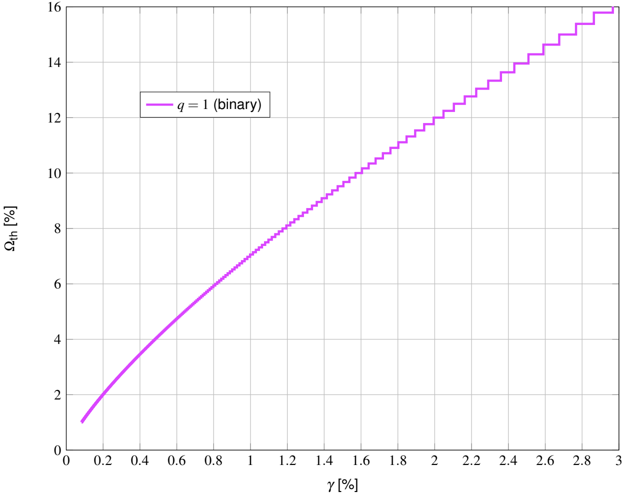
$n = 210\,000$ items

Fixed rate $\Omega = 5\,\%$, i.e., $m = 10\,500$ tests

Density evolution thresholds: γ_{th}

q	$d_{\text{v,x}}$	$d_{\text{v}} = 4$	$d_{\text{v}} = 5$	$d_{\text{v}} = 6$	$d_{\text{v}} = 7$	$d_{\text{v}} = 8$
1		0.598	0.641	0.646	0.635	0.618
4	2	0.590	0.660	0.694	0.706	0.702
5	2	0.592	0.672	0.725	0.746	0.744
10	3	0.549	0.636	0.693	0.774	0.694

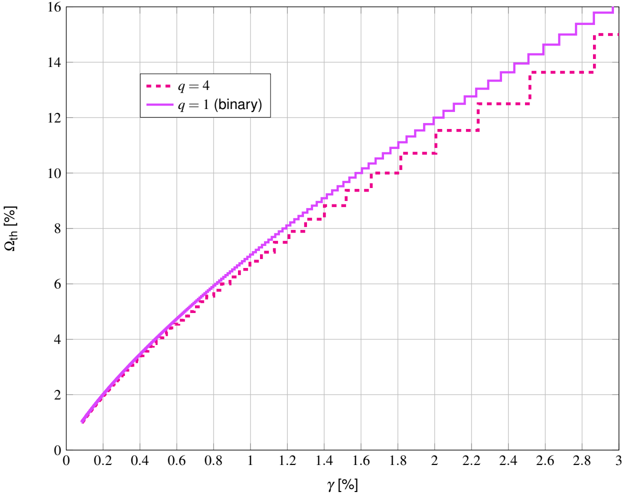
Minimum Rate Ω_{th} for a Fixed γ



$$\Omega^*_{\text{th}} = \frac{d_{\text{v}}}{d_{\text{c}}} = \frac{m}{n}$$

(smaller is better)

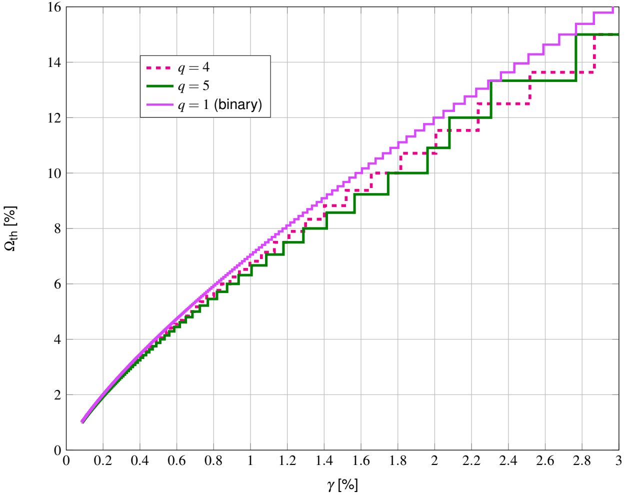
Minimum Rate Ω_{th} for a Fixed γ



$$\Omega^*_{\mathsf{th}} = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}} = \frac{m}{n}$$

(smaller is better)

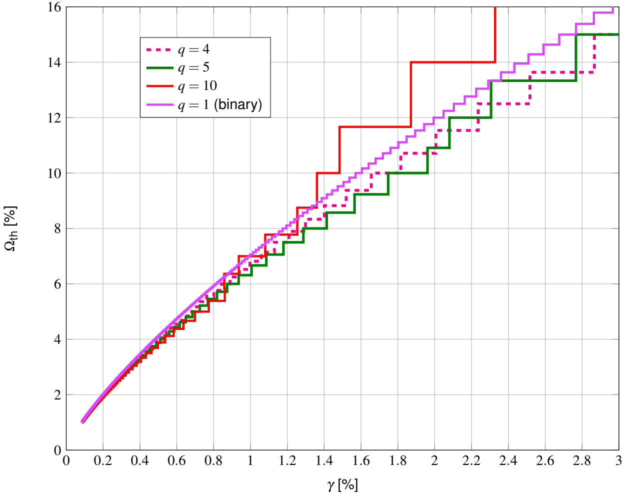
Minimum Rate Ω_{th} for a Fixed γ



$$\Omega^*_{\text{th}} = \frac{d_{\text{v}}}{d_{\text{c}}} = \frac{m}{n}$$

(smaller is better)

Minimum Rate Ω_{th} for a Fixed γ

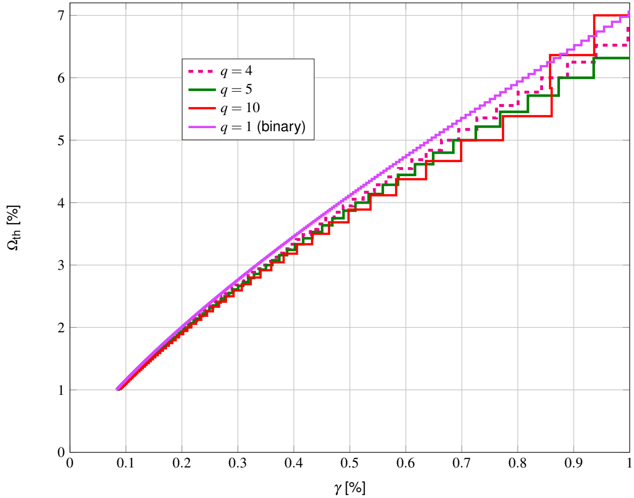


$$\Omega_{\mathrm{th}}^{*} = \frac{d_{\mathrm{v}}}{d_{\mathrm{c}}} = \frac{m}{n}$$

(smaller is better)

Minimum Rate Ω_{th} for a Fixed γ

Consider a smaller range:



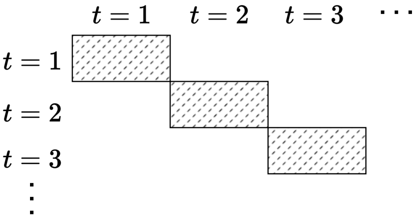
$$\Omega_{\text{th}}^* = \frac{d_{\text{v}}}{d_{\text{c}}} = \frac{m}{n}$$

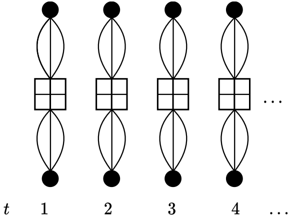
(smaller is better)

Observe:
best q depends on range of γ

Group Testing with Spatial Coupling

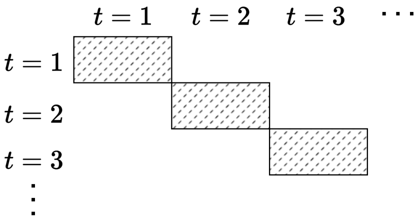
- **Classical approach**: test each block of items separately

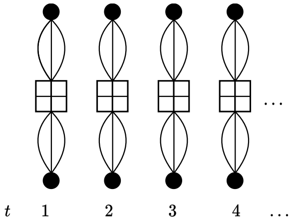




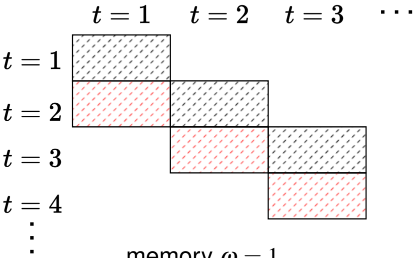
Group Testing with Spatial Coupling

- **Classical approach**: test each block of items separately

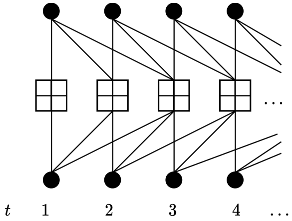




- **Spatial Coupling**: Interconnect blocks (motivated by results in coding theory)



memory $\omega = 1$



Spatial coupling: Performance for Fixed Rate $(q = 1)$

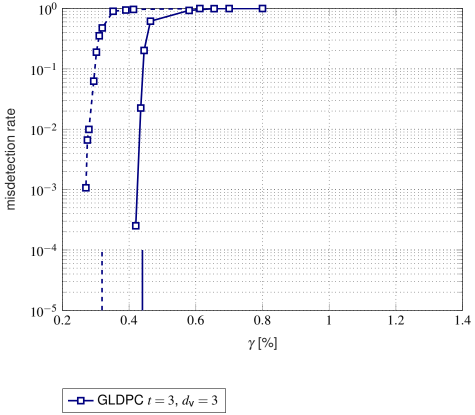
Table: γ_{th} for $\Omega = 5\%$ with GLDPC Code-Based

t	d_{v}	$\omega = 0$	$\omega = 1$	$\omega = 5$	$\omega = 10$
1	3	0.3708	0.4166	0.4166	0.4166
	4	0.3510	0.4395	0.4425	0.4425
3	3	0.3189	0.4257	0.4379	0.4395
	4	0.2441	0.3662	0.4028	0.4028
5	3	0.2686	0.3784	0.4089	0.4089
	4	0.2014	0.3159	0.3769	0.3769

Spatial coupling: Performance for Fixed Rate $(q = 1)$

Table: γ_{th} for $\Omega = 5\%$ with GLDPC Code-Based

t	d_{v}	$\omega = 0$	$\omega = 1$	$\omega = 5$	$\omega = 10$
1	3	0.3708	0.4166	0.4166	0.4166
	4	0.3510	0.4395	0.4425	0.4425
3	3	0.3189	0.4257	0.4379	0.4395
	4	0.2441	0.3662	0.4028	0.4028
5	3	0.2686	0.3784	0.4089	0.4089
	4	0.2014	0.3159	0.3769	0.3769



- $n = 153\,000$, $L = 200$, $\omega = 5$
- solid(coupled) dashed(uncoupled)

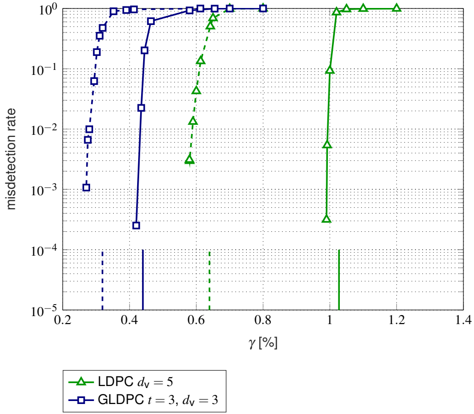
Spatial coupling: Performance for Fixed Rate $(q = 1)$

Table: γ_{th} for $\Omega = 5\%$ with GLDPC Code-Based

t	d_v	$\omega = 0$	$\omega = 1$	$\omega = 5$	$\omega = 10$
1	3	0.3708	0.4166	0.4166	0.4166
	4	0.3510	0.4395	0.4425	0.4425
3	3	0.3189	0.4257	0.4379	0.4395
	4	0.2441	0.3662	0.4028	0.4028
5	3	0.2686	0.3784	0.4089	0.4089
	4	0.2014	0.3159	0.3769	0.3769

Table: γ_{th} for $\Omega = 5\%$ with LDPC Code-Based

d_v	$\omega = 0$	$\omega = 1$	$\omega = 5$	$\omega = 10$
4	0.5982	0.8423	0.8540	0.8540
5	0.6416	0.9682	1.0274	1.0250
6	0.6464	1.0044	1.1325	1.1327
10	0.5773	0.9188	1.2814	1.2816



- $n = 153\,000$, $L = 200$, $\omega = 5$
- solid(coupled) dashed(uncoupled)

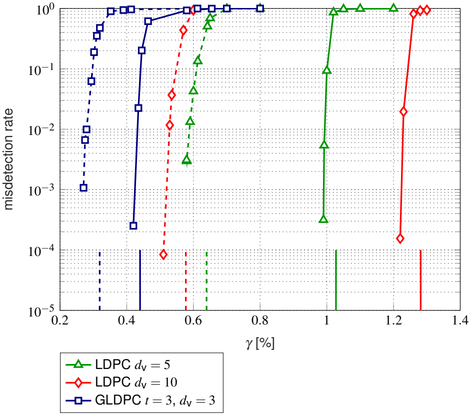
Spatial coupling: Performance for Fixed Rate $(q = 1)$

Table: γ_{th} for $\Omega = 5\%$ with GLDPC Code-Based

t	d_{v}	$\omega = 0$	$\omega = 1$	$\omega = 5$	$\omega = 10$
1	3	0.3708	0.4166	0.4166	0.4166
	4	0.3510	0.4395	0.4425	0.4425
3	3	0.3189	0.4257	0.4379	0.4395
	4	0.2441	0.3662	0.4028	0.4028
5	3	0.2686	0.3784	0.4089	0.4089
	4	0.2014	0.3159	0.3769	0.3769

Table: γ_{th} for $\Omega = 5\%$ with LDPC Code-Based

d_{v}	$\omega = 0$	$\omega = 1$	$\omega = 5$	$\omega = 10$
4	0.5982	0.8423	0.8540	0.8540
5	0.6416	0.9682	1.0274	1.0250
6	0.6464	1.0044	1.1325	1.1327
10	0.5773	0.9188	1.2814	1.2816



- $n = 153\,000$, $L = 200$, $\omega = 5$
- solid(coupled) dashed(uncoupled)

Proving threshold saturation: minimum Ω for a fixed γ $(q=1)$

- **Vector admissible system:** [YED2012] a recursion $(\mathbf{f},\boldsymbol{g})$ with

$$\boldsymbol{x}^{(\ell)} = \mathbf{f}\left(\boldsymbol{g}(\boldsymbol{x}^{(\ell-1)});\varepsilon\right) \;, \quad \boldsymbol{x}^{(0)} = \mathbf{1} \;, \; \varepsilon \in [0,1]$$

where $\mathbf{f}(\boldsymbol{x}) = \;[f_1(\boldsymbol{x}),\cdots,f_d(\boldsymbol{x})]$ and $\boldsymbol{g}(\boldsymbol{x}) = \;[g_1(\boldsymbol{x}),\cdots,g_d(\boldsymbol{x})]$ are twice continuously differentiable and strictly increasing in all arguments.

[YED2012] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, “A simple proof of threshold saturation for coupled vector recursions,” in*Proc. IEEE Inf. Theory Workshop (ITW)*, 2012.

Proving threshold saturation: minimum Ω for a fixed γ $(q = 1)$

- **Vector admissible system:** [YED2012] a recursion $(\mathbf{f},\boldsymbol{g})$ with

$$\boldsymbol{x}^{(\ell)} = \mathbf{f}\left(\boldsymbol{g}(\boldsymbol{x}^{(\ell-1)});\varepsilon\right) \;, \quad \boldsymbol{x}^{(0)} = \mathbf{1} \;,\; \varepsilon \in [0,1]$$

  where $\mathbf{f}(\boldsymbol{x}) = \;\; [f_1(\boldsymbol{x}),\cdots,f_d(\boldsymbol{x})]$ and $\boldsymbol{g}(\boldsymbol{x}) = \;\; [g_1(\boldsymbol{x}),\cdots,g_d(\boldsymbol{x})]$ are twice continuously differentiable and strictly increasing in all arguments.

- Setting $\varepsilon = 1 - \frac{1}{d_{\mathsf{c}}}$ we get from density evolution equations:

$$\mathbf{f}(y_0,y_1;\varepsilon) = \left[1-(1-y_1)^{\frac{\varepsilon}{1-\varepsilon}}, \quad 1-(1-y_0)^{\frac{\varepsilon}{1-\varepsilon}}\right]$$

$$\boldsymbol{g}(x_0,x_1) = \left[(1-\gamma)\cdot x_0^{d_{\mathsf{v}}-1}, \quad \gamma \cdot x_1^{d_{\mathsf{v}}-1}\right]$$

- Threshold saturation occurs

[YED2012] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, “A simple proof of threshold saturation for coupled vector recursions,” in*Proc. IEEE Inf. Theory Workshop (ITW)*, 2012.

Proving threshold saturation: minimum Ω for a fixed γ $(q=1)$

- **Vector admissible system:** [YED2012] a recursion $(\mathbf{f},\boldsymbol{g})$ with

$$\boldsymbol{x}^{(\ell)}=\mathbf{f}\left(\boldsymbol{g}(\boldsymbol{x}^{(\ell-1)});\varepsilon\right)\ ,\quad \boldsymbol{x}^{(0)}=\mathbf{1}\ ,\ \varepsilon\in[0,1]$$

where $\mathbf{f}(\boldsymbol{x})=\ [f_1(\boldsymbol{x}),\cdots,f_d(\boldsymbol{x})]$ and $\boldsymbol{g}(\boldsymbol{x})=\ [g_1(\boldsymbol{x}),\cdots,g_d(\boldsymbol{x})]$ are twice continuously differentiable and strictly increasing in all arguments.

- Setting $\varepsilon=1-\frac{1}{d_{\mathsf{c}}}$ we get from density evolution equations:

$$\mathbf{f}(y_0,y_1;\varepsilon)=\left[1-(1-y_1)^{\frac{\varepsilon}{1-\varepsilon}},\quad 1-(1-y_0)^{\frac{\varepsilon}{1-\varepsilon}}\right]$$

$$\boldsymbol{g}(x_0,x_1)=\left[(1-\gamma)\cdot x_0^{d_{\mathsf{v}}-1},\quad \gamma\cdot x_1^{d_{\mathsf{v}}-1}\right]$$

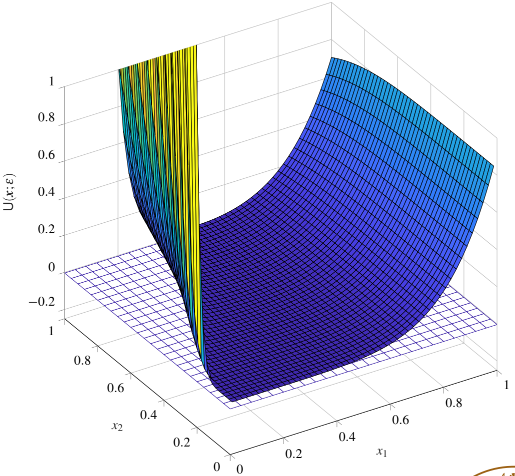
- Threshold saturation occurs
- The potential function is then given as

$$\mathrm{U}(\boldsymbol{x};\varepsilon)=\int_0^1\Big(\big(z(\lambda)-\mathbf{f}\left(\boldsymbol{g}(z(\lambda));\varepsilon\right)\big)\boldsymbol{D}\boldsymbol{g}'(z(\lambda))\Big)\cdot z'(\lambda)\mathrm{d}\lambda$$

[YED2012] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, “A simple proof of threshold saturation for coupled vector recursions,” in*Proc. IEEE Inf. Theory Workshop (ITW)*, 2012.

Potential function $(q = 1)$

$$\mathsf{U}(\boldsymbol{x};\varepsilon) = (1-p)x_1^{d_\mathsf{v}-1}\left((1-\varepsilon)\frac{1-\left(1-px_2^{d_\mathsf{v}-1}\right)^{\frac{1}{1-\varepsilon}}}{px_2^{d_\mathsf{v}-1}} + \frac{(d_\mathsf{v}-1)}{d_\mathsf{v}}x_1 - 1\right)$$
$$+ px_2^{d_\mathsf{v}-1}\left((1-\varepsilon)\frac{1-\left(1-(1-p)x_2^{d_\mathsf{v}-1}\right)^{\frac{1}{1-\varepsilon}}}{(1-p)x_1^{d_\mathsf{v}-1}} + \frac{(d_\mathsf{v}-1)}{d_\mathsf{v}}x_2 - 1\right)$$

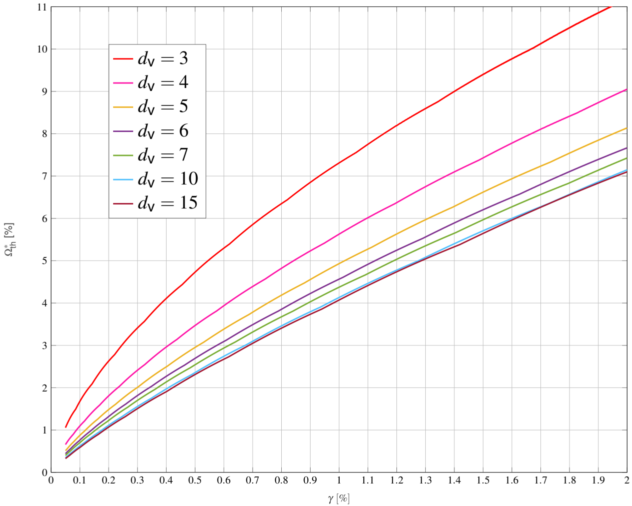


Potential threshold:

$$\varepsilon^* = \sup\{\varepsilon \in [0,1] \mid \min_{\boldsymbol{x}} \mathsf{U}(\boldsymbol{x};\varepsilon) \geq 0\}\,.$$

$d_\mathsf{v} = 6$, $\gamma = 1\%$ with $\varepsilon^* = 0.9924$. $\mathsf{U}(\boldsymbol{x};\varepsilon)$ is above the $z = 0$ plane since $\varepsilon = 0.9667 < \varepsilon^*$.

Potential thresholds $(q = 1)$



$$\Omega^*_{\mathsf{th}} = \frac{d_\mathsf{v}}{d_\mathsf{c}} = d_\mathsf{v}(1 - \varepsilon^*)\,.$$

$$\varepsilon^* = \sup\{\varepsilon \in [0,1] \,|\, \min_{\boldsymbol{x}} \mathsf{U}(\boldsymbol{x};\varepsilon) \geq 0\}\,.$$

The minimum rate Ω^*_{th} for a fixed γ computed from the potential threshold ε^*.

Conclusions and Outlook

Conclusions

- Using a simple LDPC code significantly outperforms a GLDPC construction
- We can measure the performance by two different approaches
  - Fixing the proportion γ and determining minimum rate Ω
  - Fixing the rate, Ω and determining the maximum γ

with t-error-correcting component code

Conclusions and Outlook

Conclusions

- Using a simple LDPC code significantly outperforms a GLDPC construction
- We can measure the performance by two different approaches
  - Fixing the proportion γ and determining minimum rate Ω
  - Fixing the rate, Ω and determining the maximum γ

  with t-error-correcting component code
- Bundling of tests: non-binary messages can further improve performance

Conclusions and Outlook

Conclusions

- Using a simple LDPC code significantly outperforms a GLDPC construction
- We can measure the performance by two different approaches
  - Fixing the proportion γ and determining minimum rate Ω
  - Fixing the rate, Ω and determining the maximum γ

  with t-error-correcting component code
- Bundling of tests: non-binary messages can further improve performance
- With spatial coupling we can improve the performance of the binary scheme
- Threshold saturation: with coupling the BP decoder achieves the potential threshold

Conclusions and Outlook

Conclusions

- Using a simple LDPC code significantly outperforms a GLDPC construction
- We can measure the performance by two different approaches
  - Fixing the proportion γ and determining minimum rate Ω
  - Fixing the rate, Ω and determining the maximum γ

  with t-error-correcting component code

- Bundling of tests: non-binary messages can further improve performance

- With spatial coupling we can improve the performance of the binary scheme
- Threshold saturation: with coupling the BP decoder achieves the potential threshold

Outlook

- Spatial coupling with q-bundles
- Looking at soft message passing