On the Computational Complexity of Blind Detection of Binary Linear Codes

Alexios Balatsoukas-Stimming and Aris Filos-Ratsikas

École polytechnique fédérale de Lausanne, Switzerland

July 11, 2019
IEEE International Symposium on Information Theory

EPFL

Telecommunications Circuits Laboratory

Blind Detection of Channel Codes

What is blind code detection?

Definition

Given a set of candidate channel codes $\mathcal{C}$, determine which channel code $C \in \mathcal{C}$ is being used based on noisy channel observations.

Blind Detection of Channel Codes

What is blind code detection?

Definition

Given a set of candidate channel codes $\mathcal{C}$, determine which channel code $C \in \mathcal{C}$ is being used based on noisy channel observations.

Application examples:

- Cognitive radio.

Blind Detection of Channel Codes

What is blind code detection?

Definition

Given a set of candidate channel codes $\mathcal{C}$, determine which channel code $C \in \mathcal{C}$ is being used based on noisy channel observations.

Application examples:

- Cognitive radio.
- Eavesdroppping.

Blind Detection of Channel Codes

What is blind code detection?

Definition

Given a set of candidate channel codes $\mathcal{C}$, determine which channel code $C \in \mathcal{C}$ is being used based on noisy channel observations.

Application examples:

- Cognitive radio.
- Eavesdroppping.
- Control channels.

Blind Detection of Channel Codes

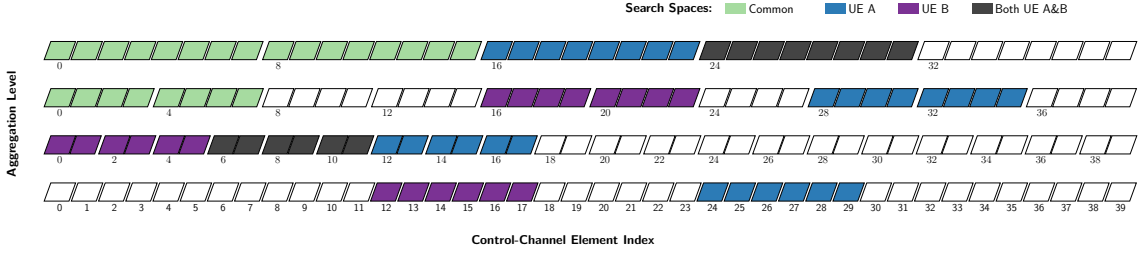
What is blind code detection?

Definition

Given a set of candidate channel codes $\mathcal{C}$, determine which channel code $C \in \mathcal{C}$ is being used based on noisy channel observations.

Application examples:

- Cognitive radio.
- Eavesdroppping.
- Control channels.



Heuristic Blind Detection Approaches

- **Likelihood ratio tests** (e.g., [1,2]):
  - Binary linear codes satisfy parity constraints: $c_{i_1} + \cdots + c_{i_m} = 0$.
  - Calculate probability $p(c_{i_1} + \cdots + c_{i_m} = 0 | y_{i_1}, \ldots, y_{i_m})$ for parity constraints and pick most likely code.

[1] R. Moosavi, E. G. Larsson, "A Fast Scheme for Blind Identification of Channel Codes," IEEE GLOBECOM, Dec. 2011.

[2] T. Xia, H.-C. Wu, "Novel Blind Identification of LDPC Codes Using Average LLR of Syndrome a Posteriori Probability," IEEE TSP, Feb. 2014.

Heuristic Blind Detection Approaches

- **Likelihood ratio tests** (e.g., [1,2]):
  - Binary linear codes satisfy parity constraints: $c_{i_1} + \cdots + c_{i_m} = 0$.
  - Calculate probability $p(c_{i_1} + \cdots + c_{i_m} = 0 | y_{i_1}, \ldots, y_{i_m})$ for parity constraints and pick most likely code.

- **Partial decoding algorithms** (e.g., [3,4]):
  - Use a (simplified) decoding algorithm to calculate a quality metric for the codeword.
  - Repeat for all candidate codes and pick the highest quality metric code.

[1] R. Moosavi, E. G. Larsson, "A Fast Scheme for Blind Identification of Channel Codes," IEEE GLOBECOM, Dec. 2011.

[2] T. Xia, H.-C. Wu, "Novel Blind Identification of LDPC Codes Using Average LLR of Syndrome a Posteriori Probability," IEEE TSP, Feb. 2014.

[3] P. Giard, A. Balatsoukas-Stimming, and A. Burg, "Blind Detection of Polar Codes," IEEE SiPS, Oct. 2017.

[4] P. Giard, A. Balatsoukas-Stimming, and A. Burg, "On the Tradeoff Between Accuracy and Complexity in Blind Detection of Polar Codes," ISTC, Dec. 2018.

Heuristic Blind Detection Approaches

- **Likelihood ratio tests** (e.g., [1,2]):
  - Binary linear codes satisfy parity constraints: $c_{i_1} + \cdots + c_{i_m} = 0$.
  - Calculate probability $p(c_{i_1} + \cdots + c_{i_m} = 0 | y_{i_1}, \ldots, y_{i_m})$ for parity constraints and pick most likely code.

- **Partial decoding algorithms** (e.g., [3,4]):
  - Use a (simplified) decoding algorithm to calculate a quality metric for the codeword.
  - Repeat for all candidate codes and pick the highest quality metric code.

- **Side-information:** (e.g., [5])
  - Embed side-information into codeword bits (i.e., user ID, CRC).
  - Decode and check whether side-information matches what is expected.

[1] R. Moosavi, E. G. Larsson, "A Fast Scheme for Blind Identification of Channel Codes," IEEE GLOBECOM, Dec. 2011.

[2] T. Xia, H.-C. Wu, "Novel Blind Identification of LDPC Codes Using Average LLR of Syndrome a Posteriori Probability," IEEE TSP, Feb. 2014.

[3] P. Giard, A. Balatsoukas-Stimming, and A. Burg, "Blind Detection of Polar Codes," IEEE SiPS, Oct. 2017.

[4] P. Giard, A. Balatsoukas-Stimming, and A. Burg, "On the Tradeoff Between Accuracy and Complexity in Blind Detection of Polar Codes," ISTC, Dec. 2018.

[5] C. Condo, S. A. Hashemi, A. Ardakani, F. Ercan, W. J. Gross, "Design and Implementation of a Polar Codes Blind Detection Scheme," IEEE TCAS-II, Sep. 2018.

Contributions & Outline

- It seems to be generally accepted that the problem is “difficult.”

Contributions & Outline

- It seems to be generally accepted that the problem is "difficult."
- **However:** the difficulty of blind detection of channel codes **has not been formalized**.

Contributions & Outline

- It seems to be generally accepted that the problem is "difficult."
- **However:** the difficulty of blind detection of channel codes **has not been formalized**.

Contribution

We show that (one formulation of) blind detection of binary linear codes is NP-hard!

Contributions & Outline

- It seems to be generally accepted that the problem is "difficult."
- **However:** the difficulty of blind detection of channel codes **has not been formalized**.

Contribution

We show that (one formulation of) blind detection of binary linear codes is NP-hard!

Outline

- *Binary linear codes background.*

Contributions & Outline

- It seems to be generally accepted that the problem is "difficult."
- **However:** the difficulty of blind detection of channel codes **has not been formalized**.

Contribution

We show that (one formulation of) blind detection of binary linear codes is NP-hard!

Outline

- *Binary linear codes background.*
- *Formulation of the* Minimum Distance Code Detection *problem.*

Contributions & Outline

- It seems to be generally accepted that the problem is "difficult."
- **However:** the difficulty of blind detection of channel codes **has not been formalized**.

Contribution

We show that (one formulation of) blind detection of binary linear codes is NP-hard!

Outline

- *Binary linear codes background.*
- *Formulation of the* Minimum Distance Code Detection *problem.*
- *Proof of NP-hardness.*

Contributions & Outline

- It seems to be generally accepted that the problem is "difficult."
- **However:** the difficulty of blind detection of channel codes **has not been formalized**.

Contribution

We show that (one formulation of) blind detection of binary linear codes is NP-hard!

Outline

- *Binary linear codes background.*
- *Formulation of the* Minimum Distance Code Detection *problem.*
- *Proof of NP-hardness.*
- *Open problems.*

Binary Linear Codes

- A binary code C of length n is a set of codewords $\mathbf{c} \in \{0,1\}^n$.

Binary Linear Codes

- A binary code C of length n is a set of codewords $\mathbf{c} \in \{0,1\}^n$.
- For a *linear* code, if $\mathbf{c}_1, \mathbf{c}_2 \in C$ then $\mathbf{c}_1 + \mathbf{c}_2 \in C$.

Binary Linear Codes

- A binary code C of length n is a set of codewords $\mathbf{c} \in \{0,1\}^n$.
- For a *linear* code, if $\mathbf{c}_1, \mathbf{c}_2 \in C$ then $\mathbf{c}_1 + \mathbf{c}_2 \in C$.
- The *dimension* of a binary linear code is $k = \log_2 |C|$.

Binary Linear Codes

- A binary code C of length n is a set of codewords $\mathbf{c} \in \{0,1\}^n$.
- For a *linear* code, if $\mathbf{c}_1, \mathbf{c}_2 \in C$ then $\mathbf{c}_1 + \mathbf{c}_2 \in C$.
- The *dimension* of a binary linear code is $k = \log_2 |C|$.
- Any binary linear code can be represented using a *generator matrix* $\mathbf{G}$.

$$C = \left\{ \mathbf{uG} : \mathbf{u} \in \{0,1\}^k \right\}$$

Binary Linear Codes

- A binary code C of length n is a set of codewords $\mathbf{c} \in \{0,1\}^n$.
- For a *linear* code, if $\mathbf{c}_1, \mathbf{c}_2 \in C$ then $\mathbf{c}_1 + \mathbf{c}_2 \in C$.
- The *dimension* of a binary linear code is $k = \log_2 |C|$.
- Any binary linear code can be represented using a *generator matrix* $\mathbf{G}$.

$$C = \left\{ \mathbf{uG} : \mathbf{u} \in \{0,1\}^k \right\}$$

- Transmission takes place over a BSC(p):

$$\mathbf{x} = \mathbf{c} + \mathbf{e}$$

Minimum Distance Code Detection (MDCD)

- Let $\mathcal{C}$ denote a set of binary linear codes of dimension k.

Minimum Distance Code Detection (MDCD)

- Let $\mathcal{C}$ denote a set of binary linear codes of dimension k.
- Each $C \in \mathcal{C}$ is given by a generator matrix $\mathbf{G}$.

Minimum Distance Code Detection (MDCD)

- Let $\mathcal{C}$ denote a set of binary linear codes of dimension k.
- Each $C \in \mathcal{C}$ is given by a generator matrix $\mathbf{G}$.
- Let the $N \times n$ *observation matrix* be defined as: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T & \dots & \mathbf{x}_N^T \end{bmatrix}^T$.

Minimum Distance Code Detection (MDCD)

- Let $\mathcal{C}$ denote a set of binary linear codes of dimension k.
- Each $C \in \mathcal{C}$ is given by a generator matrix $\mathbf{G}$.
- Let the $N \times n$ *observation matrix* be defined as: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T & \dots & \mathbf{x}_N^T \end{bmatrix}^T$.

Problem (Minimum Distance Code Detection)

Input: $\mathbf{X}$, $\mathcal{C}$.

Output: *A generator matrix* $\mathbf{G}$ *of a binary linear code* $C_{MDCD} \in \mathcal{C}$ *such that:*

$$C_{MDCD} = \arg\min_{C \in \mathcal{C}} \sum_{i=1}^{N} d(\mathbf{x}_i, C)$$

Minimum Distance Code Detection (MDCD)

- Let $\mathcal{C}$ denote a set of binary linear codes of dimension k.
- Each $C \in \mathcal{C}$ is given by a generator matrix $\mathbf{G}$.
- Let the $N \times n$ *observation matrix* be defined as: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T & \dots & \mathbf{x}_N^T \end{bmatrix}^T$.

Problem (Minimum Distance Code Detection)

Input: $\mathbf{X}$, $\mathcal{C}$.

Output: *A generator matrix* $\mathbf{G}$ *of a binary linear code* $C_{MDCD} \in \mathcal{C}$ *such that:*

$$C_{MDCD} = \arg\min_{C \in \mathcal{C}} \sum_{i=1}^{N} d(\mathbf{x}_i, C),$$

where the distance $d(\mathbf{x}_i, C)$ *is defined as:*

$$d(\mathbf{x}_i, C) = \min_{\mathbf{c} \in C} d_H(\mathbf{x}_i, \mathbf{c}).$$

Main Result & Proof Outline

Theorem

The Minimum Distance Code Detection *problem is NP-hard.*

Main Result & Proof Outline

Theorem

The Minimum Distance Code Detection *problem is NP-hard.*

- The above result **justifies a large body of heuristics-based related work**.

Main Result & Proof Outline

Theorem

The Minimum Distance Code Detection *problem is NP-hard.*

- The above result **justifies a large body of heuristics-based related work**.

Proof outline:

1. We consider an instance of the Minimum Distance Decoding (MDD) problem, which is NP-hard.

Main Result & Proof Outline

Theorem

The Minimum Distance Code Detection *problem is NP-hard.*

- The above result **justifies a large body of heuristics-based related work**.

Proof outline:

1. We consider an instance of the Minimum Distance Decoding (MDD) problem, which is NP-hard.
2. Suppose that a polynomial-time algorithm for MDCD exists.

Main Result & Proof Outline

Theorem

The Minimum Distance Code Detection *problem is NP-hard.*

- The above result **justifies a large body of heuristics-based related work**.

Proof outline:

1. We consider an instance of the Minimum Distance Decoding (MDD) problem, which is NP-hard.
2. Suppose that a polynomial-time algorithm for MDCD exists.
3. We construct an iterative algorithm for MDD which uses the above algorithm as a subroutine k times.

Main Result & Proof Outline

Theorem

The Minimum Distance Code Detection *problem is NP-hard.*

- The above result **justifies a large body of heuristics-based related work**.

Proof outline:

1. We consider an instance of the Minimum Distance Decoding (MDD) problem, which is NP-hard.
2. Suppose that a polynomial-time algorithm for MDCD exists.
3. We construct an iterative algorithm for MDD which uses the above algorithm as a subroutine k times.
4. This would imply that MDD can be solved in polynomial time.

Main Result & Proof Outline

Theorem

The Minimum Distance Code Detection *problem is NP-hard.*

- The above result **justifies a large body of heuristics-based related work**.

Proof outline (Turing reduction):

1. We consider an instance of the Minimum Distance Decoding (MDD) problem, which is NP-hard.
2. Suppose that a polynomial-time algorithm for MDCD exists.
3. We construct an iterative algorithm for MDD which uses the above algorithm as a subroutine k times.
4. This would imply that MDD can be solved in polynomial time.

Minimum Distance Decoding

Problem (MDD)

Input: *A generator matrix* $\mathbf{G}$ *and an* n*-bit binary vector* $\mathbf{y}$*.*

Output: *An* n*-bit binary vector* $\hat{\mathbf{c}}$ *such that:*

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c} \in C} d_H(\mathbf{y}, \mathbf{c}).$$

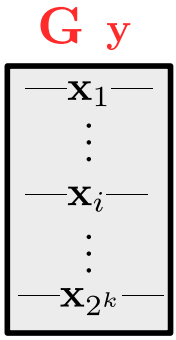
Minimum Distance Decoding

Problem (MDD)

Input: *A generator matrix* $\mathbf{G}$ *and an* n*-bit binary vector* $\mathbf{y}$*.*

Output: *An* n*-bit binary vector* $\hat{\mathbf{c}}$ *such that:*

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c} \in C} d_H(\mathbf{y}, \mathbf{c}).$$



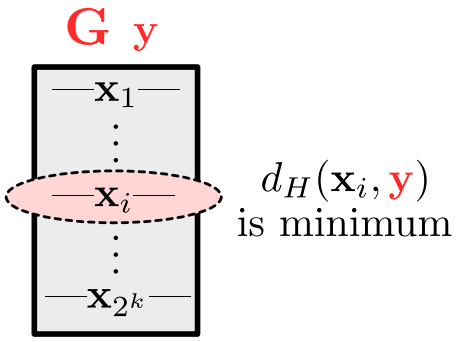
Minimum Distance Decoding

Problem (MDD)

Input: *A generator matrix* $\mathbf{G}$ *and an* n*-bit binary vector* $\mathbf{y}$*.*

Output: *An* n*-bit binary vector* $\hat{\mathbf{c}}$ *such that:*

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c} \in C} d_H(\mathbf{y}, \mathbf{c}).$$



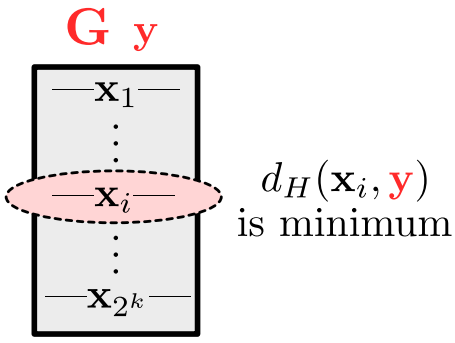
Minimum Distance Decoding

Problem (MDD)

Input: *A generator matrix* $\mathbf{G}$ *and an* n*-bit binary vector* $\mathbf{y}$*.*

Output: *An* n*-bit binary vector* $\hat{\mathbf{c}}$ *such that:*

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c} \in C} d_H(\mathbf{y}, \mathbf{c}).$$

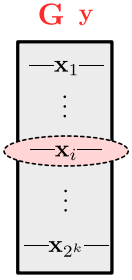


Theorem (Berlekamp et al., 1978)

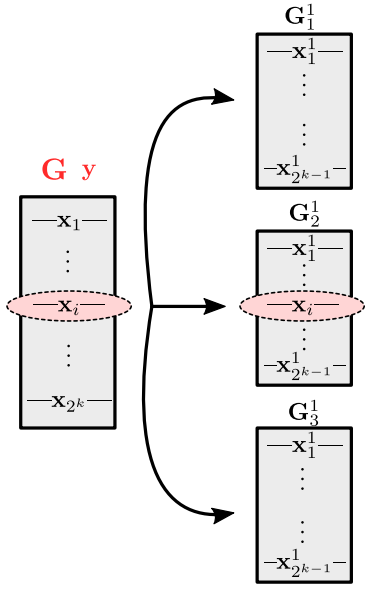
The Minimum Distance Decoding *problem is NP-hard [1].*

[1] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," IEEE Trans. Inf. Theory, May 1978.

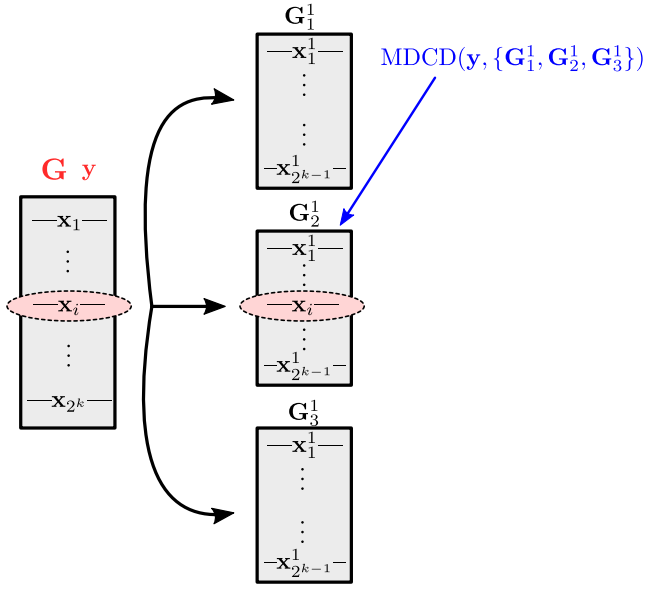
Using MDCD to Solve MDD



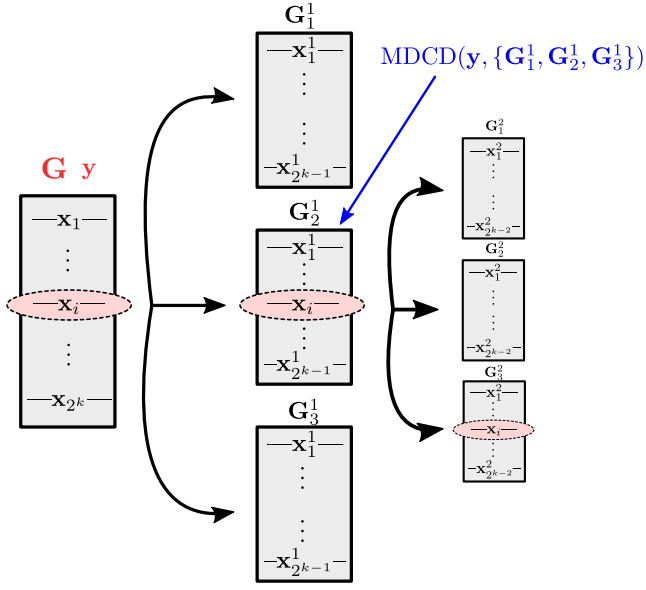
Using MDCD to Solve MDD



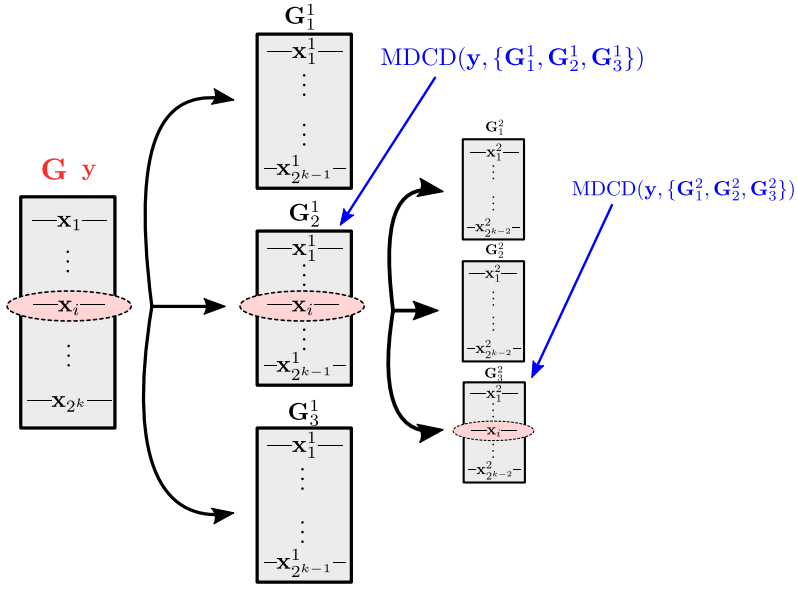
Using MDCD to Solve MDD



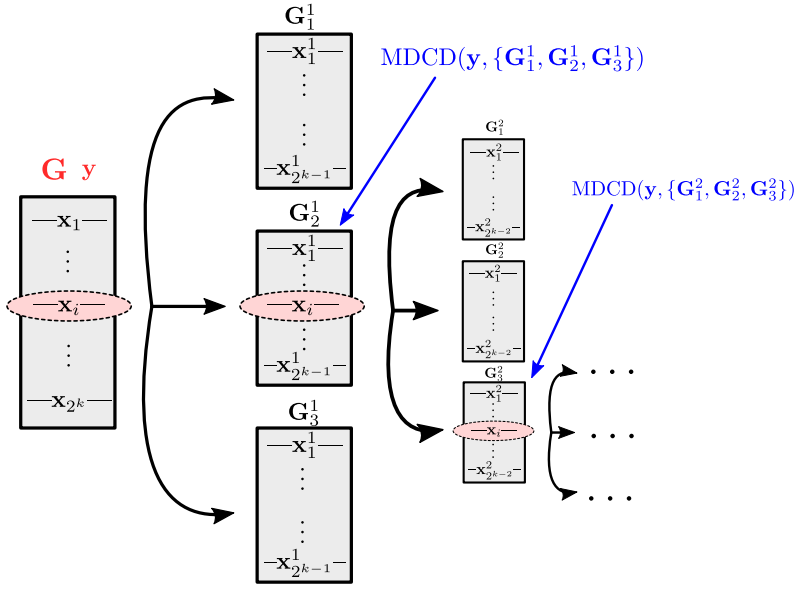
Using MDCD to Solve MDD



Using MDCD to Solve MDD



Using MDCD to Solve MDD



Using MDCD to Solve MDD

- More formally, MDD can be solved using MDCD as:

Algorithm

$\mathbf{G}^{(k)} = \mathbf{G}$;
$l = k$;
while $l > 0$ **do**
  $\{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3\} = \textsc{SplitCover}(\mathbf{G}^{(l)})$;
  $\mathbf{G}^{(l-1)} = \mathit{MDCD}(\mathbf{y}, \{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3\})$;
  $l = l - 1$;
$\hat{\mathbf{c}} = \mathbf{G}^{(0)}$;

Using MDCD to Solve MDD

- More formally, MDD can be solved using MDCD as:

Algorithm

$\mathbf{G}^{(k)} = \mathbf{G}$;
$l = k$;
while $l > 0$ **do**
  $\{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3\} = \textsc{SplitCover}(\mathbf{G}^{(l)})$;
  $\mathbf{G}^{(l-1)} = \mathit{MDCD}(\mathbf{y}, \{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3\})$;
  $l = l - 1$;
$\hat{\mathbf{c}} = \mathbf{G}^{(0)}$;

- **How is SplitCover defined?**

The SplitCover Function

- **Required properties:**
  1. Resulting codes must be linear.

The SplitCover Function

- **Required properties:**
  1. Resulting codes must be linear.
  2. $\mathbf{G}_1 \cup \mathbf{G}_2 \cup \mathbf{G}_3 = \mathbf{G}$

The SplitCover Function

- **Required properties:**
  1. Resulting codes must be linear.
  2. $\mathbf{G}_1 \cup \mathbf{G}_2 \cup \mathbf{G}_3 = \mathbf{G}$
  3. $\dim(\mathbf{G}_1) = \dim(\mathbf{G}_2) = \dim(\mathbf{G}_3) = \dim(\mathbf{G}) - 1$

The SplitCover Function

- **Required properties:**
  1. Resulting codes must be linear.
  2. $\mathbf{G}_1 \cup \mathbf{G}_2 \cup \mathbf{G}_3 = \mathbf{G}$
  3. $\dim(\mathbf{G}_1) = \dim(\mathbf{G}_2) = \dim(\mathbf{G}_3) = \dim(\mathbf{G}) - 1$

Algorithm

Function SplitCover($\mathbf{G}$):

$$\mathbf{G}_1 = \begin{bmatrix} \mathbf{g}_1^T & \mathbf{g}_3^T & \dots & \mathbf{g}_k^T \end{bmatrix}^T;$$

$$\mathbf{G}_2 = \begin{bmatrix} \mathbf{g}_2^T & \mathbf{g}_3^T & \dots & \mathbf{g}_k^T \end{bmatrix}^T;$$

$$\mathbf{G}_3 = \begin{bmatrix} (\mathbf{g}_1 + \mathbf{g}_2)^T & \mathbf{g}_3^T & \dots & \mathbf{g}_k^T \end{bmatrix}^T;$$

return $\{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3\}$;

The SplitCover Function

- **Required properties:**
  1. Resulting codes must be linear.
  2. $\mathbf{G}_1 \cup \mathbf{G}_2 \cup \mathbf{G}_3 = \mathbf{G}$
  3. $\dim(\mathbf{G}_1) = \dim(\mathbf{G}_2) = \dim(\mathbf{G}_3) = \dim(\mathbf{G}) - 1$

Algorithm

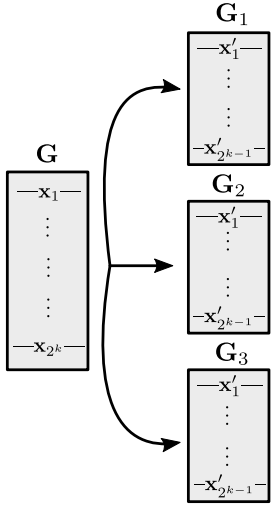
Function SplitCover($\mathbf{G}$):

$$\mathbf{G}_1 = \begin{bmatrix} \mathbf{g}_1^T & \mathbf{g}_3^T & \dots & \mathbf{g}_k^T \end{bmatrix}^T;$$

$$\mathbf{G}_2 = \begin{bmatrix} \mathbf{g}_2^T & \mathbf{g}_3^T & \dots & \mathbf{g}_k^T \end{bmatrix}^T;$$

$$\mathbf{G}_3 = \begin{bmatrix} (\mathbf{g}_1 + \mathbf{g}_2)^T & \mathbf{g}_3^T & \dots & \mathbf{g}_k^T \end{bmatrix}^T;$$

return $\{\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3\}$;



Open Problems

- **Maximum likelihood code detection (MLCD)** minimizes the probability of detection error:

$$C_{\mathsf{MLCD}} = \arg\max_{C \in \mathcal{C}} \prod_{i=1}^{N} \sum_{\mathbf{c} \in C} \left(\frac{p}{1-p} \right)^{d_{\mathsf{H}}(\mathbf{x}_i, \mathbf{c})} .$$

Open Problems

- **Maximum likelihood code detection (MLCD)** minimizes the probability of detection error:

$$C_{\text{MLCD}} = \arg\max_{C \in \mathcal{C}} \prod_{i=1}^{N} \sum_{\mathbf{c} \in C} \left(\frac{p}{1-p} \right)^{d_{\text{H}}(\mathbf{x}_i, \mathbf{c})} .$$

MLCD $\overset{\text{max-log}}{\longrightarrow}$ MDCD, but this does not imply something about its complexity.

Open Problems

- **Maximum likelihood code detection (MLCD)** minimizes the probability of detection error:

$$C_{\text{MLCD}} = \arg\max_{C \in \mathcal{C}} \prod_{i=1}^{N} \sum_{\mathbf{c} \in C} \left(\frac{p}{1-p} \right)^{d_{\text{H}}(\mathbf{x}_i, \mathbf{c})} .$$

  MLCD $\overset{\text{max-log}}{\longrightarrow}$ MDCD, but this does not imply something about its complexity.

- **Complexity for specific classes of codes** (e.g., LDPC codes, polar codes)
  1. NP-hardness of MDD for this class.
  2. A SplitCover-like procedure for this class.

Open Problems

- **Maximum likelihood code detection (MLCD)** minimizes the probability of detection error:

$$C_{\mathsf{MLCD}} = \arg\max_{C \in \mathcal{C}} \prod_{i=1}^{N} \sum_{\mathbf{c} \in C} \left(\frac{p}{1-p} \right)^{d_{\mathsf{H}}(\mathbf{x}_i, \mathbf{c})} .$$

  MLCD $\overset{\text{max-log}}{\longrightarrow}$ MDCD, but this does not imply something about its complexity.

- **Complexity for specific classes of codes** (e.g., LDPC codes, polar codes)
  1. NP-hardness of MDD for this class.
  2. A SplitCover-like procedure for this class.

- Distinguish between **codes** (dimension k) and **noise** (dimension n).

Thanks!

Questions?