Linear Programming Decoding for Low-Density Parity-Check Codes

Shuai Zhang
Supervisor: Prof. Christian Schlegel
HCDC

January 24, 2008
Outline

1 Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2 LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3 LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Message-Passing

4 Others
Outline

1. **Introduction**
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2. **LP Decoding for Error-Correcting Codes**
   - Fractional Distance
   - C-Symmetry

3. **LP for LDPC**
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Message-Passing

4. **Others**
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Introduction</strong></td>
</tr>
<tr>
<td>• ML Decoding</td>
</tr>
<tr>
<td>• Linear Programming</td>
</tr>
<tr>
<td>• LP Relaxation</td>
</tr>
<tr>
<td><strong>2 LP Decoding for Error-Correcting Codes</strong></td>
</tr>
<tr>
<td>• Fractional Distance</td>
</tr>
<tr>
<td>• C-Symmetry</td>
</tr>
<tr>
<td><strong>3 LP for LDPC</strong></td>
</tr>
<tr>
<td>• LP Polytope for LDPC</td>
</tr>
<tr>
<td>• $d_{frac}$</td>
</tr>
<tr>
<td>• Comparison with Massage-Passing</td>
</tr>
<tr>
<td><strong>4 Others</strong></td>
</tr>
</tbody>
</table>

Shuai Zhang

Linear Programming Decoding for LDPC
Find the codeword \( y^* \) that maximizes the likelihood of what was received from the channel \( \tilde{y} \), given codeword \( y \) was sent:

\[
y^* = \arg \max_{y \in C} \Pr [\tilde{y} \text{ received} | y \text{ transmitted}]
\]

ML decoding is optimal but is NP-hard in general.
Linear Likelihood Cost Function

Given a received word \( \tilde{y} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n] \), the log-likelihood ratio \( \gamma_i \) of a code bit \( y_i \) is:

\[
\gamma_i = \ln \left( \frac{\Pr[\tilde{y}_i|y_i = 0]}{\Pr[\tilde{y}_i|y_i = 1]} \right)
\]
Linear Likelihood Cost Function

Given a received word $\tilde{y} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n]$, the log-likelihood ratio $\gamma_i$ of a code bit $y_i$ is:

$$
\gamma_i = \ln \left( \frac{\Pr[\tilde{y}_i | y_i = 0]}{\Pr[\tilde{y}_i | y_i = 1]} \right)
$$

$$
y_i^* = \begin{cases} 
0, & \text{if } \gamma_i > 0 \\
1, & \text{if } \gamma_i < 0 
\end{cases}
$$
Given a received word $\tilde{y} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n]$, the log-likelihood ratio $\gamma_i$ of a code bit $y_i$ is:

$$\gamma_i = \ln \left( \frac{\Pr[\tilde{y}_i | y_i = 0]}{\Pr[\tilde{y}_i | y_i = 1]} \right)$$

$$y_i^* = \begin{cases} 
0, & \text{if } \gamma_i > 0 \\
1, & \text{if } \gamma_i < 0
\end{cases}$$

Recall, for BSC with $p < \frac{1}{2}$:

$$\gamma_i = \begin{cases} 
\ln \left( \frac{1-p}{p} \right) & \rightarrow 1, \text{ if } \tilde{y}_i = 0 \\
\ln \left( \frac{p}{1-p} \right) = -\ln \left( \frac{1-p}{p} \right) & \rightarrow -1, \text{ if } \tilde{y}_i = 1
\end{cases}$$
Linear Likelihood Cost Function

Given a received word $\tilde{y} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n]$, the log-likelihood ratio $\gamma_i$ of a code bit $y_i$ is:

$$
\gamma_i = \ln \left( \frac{\Pr[\tilde{y}_i|y_i = 0]}{\Pr[\tilde{y}_i|y_i = 1]} \right)
$$

$$
y_i^* = \begin{cases} 
0, & \text{if } \gamma_i > 0 \\
1, & \text{if } \gamma_i < 0 
\end{cases}
$$

Recall, for BSC with $p < \frac{1}{2}$:

$$
\gamma_i = \begin{cases} 
\ln \left( \frac{1-p}{p} \right) \rightarrow 1, & \text{if } \tilde{y}_i = 0 \\
\ln \left( \frac{p}{1-p} \right) = - \ln \left( \frac{1-p}{p} \right) \rightarrow -1, & \text{if } \tilde{y}_i = 1
\end{cases}
$$

Define $\gamma_i$ as the cost of $y_i$, and $\sum_{i=1}^{n} \gamma_i y_i$ as the cost of $y$. 
**Theorem.** For any binary-input memoryless channel, the codeword of minimum cost is the ML codeword.
**Theorem.** For any binary-input memoryless channel, the codeword of minimum cost is the ML codeword.

**Proof.**

\[ y^* = \arg \max_{y \in C} \Pr \left[ \tilde{y} \text{ received} \mid y \text{ transmitted} \right] \]

\[ = \arg \max_{y \in C} \left( \prod_{i=1}^{n} \Pr_{\text{noise}} [\tilde{y}_i \mid y_i] \right) \]

\[ = \arg \min_{y \in C} \left( -\ln \prod_{i=1}^{n} \Pr_{\text{noise}} [\tilde{y}_i \mid y_i] \right) \]

\[ = \arg \min_{y \in C} \left( -\sum_{i=1}^{n} \ln \Pr_{\text{noise}} [\tilde{y}_i \mid y_i] \right) \]

\[ = \arg \min_{y \in C} \left( \sum_{i=1}^{n} \left( \ln \Pr_{\text{noise}} [\tilde{y}_i \mid y_i = 0] - \ln \Pr_{\text{noise}} [\tilde{y}_i \mid y_i] \right) \right) \]

\[ = \arg \min_{y \in C} \left( \sum_{i:y_i=1}^{n} \ln \left( \frac{\Pr_{\text{noise}} [\tilde{y}_i \mid y_i = 0]}{\Pr_{\text{noise}} [\tilde{y}_i \mid y_i = 1]} \right) \right) \]

\[ = \arg \min_{y \in C} \sum_{i=1}^{n} \gamma_i y_i \]
Outline

1. Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2. LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3. LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Message-Passing

4. Others
Linear programming (LP) problems involve the optimization of a linear objective function, subject to linear equality and inequality constraints.

The constraints define a convex polytope (domain).

- Every LP has an associated polytope.
- The goal is to find a point in the polytope where the objective function achieves the optimal value.
- The optimum point in an LP is always obtained at a vertex of the polytope. So the set of the vertices is the feasible solution space.
Problem: the solution to the LP may contain *real* values, however only *integers* may be meaningful.

Restrict all variables to be integers $\Rightarrow$ an integer linear programming (ILP) problem

ILP is NP-hard, whereas LP can be solved efficiently.

Binary integer programming is the special case of ILP where variables are required to be 0 or 1, but is also NP-hard.
Problem: the solution to the LP may contain real values, however only integers may be meaningful.

Restrict all variables to be integers ⇒ an integer linear programming (ILP) problem

ILP is NP-hard, whereas LP can be solved efficiently.

Binary integer programming is the special case of ILP where variables are required to be 0 or 1, but is also NP-hard.

A natural way to solve an ILP is to remove the integer constraints ($\{0, 1\}^n \rightarrow [0, 1]^n$), solve the resulting LP, and somehow massage the solution into one that is meaningful. This technique is called linear programming relaxation.
Outline

1 Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2 LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3 LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Message-Passing

4 Others
ILP

Objective function: \( \min_{y \in C} \sum_{i=1}^{n} \gamma_i y_i \)

Constraints: \( y_i = 0, 1 \)

LP relaxation

Objective function: \( \min_{y \in V(P)} \sum_{i=1}^{n} \gamma_i y_i \)

Constraints (polytope \( P \)): \( 0 \leq y_i \leq 1 \)

\( \vdots \)
Vertex Cover Problem

Given an undirected graph $G = (V, E)$, and a cost $\gamma_i$ for each $i \in V$, find the lowest-cost subset $S \subseteq V$ s.t. that every edge in $E$ has at least one endpoint in $S$.

1. A variable $v_i$ for each $i \in V$ indicates whether or not to include $i$ in $S$:

   $$v_i = \begin{cases} 
   0, & \text{then } i \notin S \\
   1, & \text{then } i \in S 
   \end{cases}$$

2. A constraint to force at least one node of each edge is in $S$:

   $$\forall (i, j) \in E, v_i + v_j \geq 1$$

3. Objective function:

   $$\min \sum_{i \in V} \gamma_i v_i$$
**Vertex Cover Problem**

- **ILP:**
  - **Objective function:** \( \min \sum_{i \in V} \gamma_i v_i \)
  - **Constraints:** \( \forall i \in V, v_i \in \{0, 1\} \)
    \( \forall (i,j) \in E, v_i + v_j \geq 1 \)
  - **Output:** \( S = \{i \in V | v_i = 1\} \)
Vertex Cover Problem

- **ILP:**
  - Objective function: $\min \sum_{i \in V} \gamma_i v_i$
  - Constraints: $\forall i \in V, v_i \in \{0, 1\}$
  - $\forall (i, j) \in E, v_i + v_j \geq 1$
  - Output: $S = \{i \in V | v_i = 1\}$

- **LP relaxation:**
  - Objective function: $\min \sum_{i \in V} \gamma_i v_i$
  - Constraints: $\forall i \in V, 0 \leq v_i \leq 1$
  - $\forall (i, j) \in E, v_i + v_j \geq 1$
An Example of Vertex Cover Problem

Input: a cycle on three vertices, each with a cost of 1.
An Example of Vertex Cover Problem

Input: a cycle on three vertices, each with a cost of 1.

Objective function: \( \min(v_1 + v_2 + v_3) \)

Constraints:
- \( 0 \leq v_1, v_2, v_3 \leq 1 \)
- \( v_1 + v_2 \geq 1 \)
- \( v_1 + v_3 \geq 1 \)
- \( v_2 + v_3 \geq 1 \)

Recall general-purpose convex optimization algorithms:

- **Simplex**: not polynomial but practical
- **Ellipsoid**: polynomial but slow in practical
An Example of Vertex Cover Problem

> with(simplex):

> cnsts :=

\{ x + y \geq 1, x + z \geq 1, y + z \geq 1, x \leq 1, y \leq 1, z \leq 1 \}:

> obj := x + y + z:

> minimize(obj, cnsts, NONNEGATIVE);

\{ x = \frac{1}{2}, y = \frac{1}{2}, z = \frac{1}{2} \}
Outline

1 Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2 LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3 LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Message-Passing

4 Others
1. Define LP variables $f_i$ for each code bit, $f_i \in [0, 1]$.

2. With additional linear constraints on $f_i$, obtain a polytope $P \subseteq [0, 1]^n$.

3. We say $P$ is proper if the set of integral points in $P$ is exactly the codewords set $C$: $P \cap \{0, 1\}^n = C$.

4. Let $V(P)$ be the set of vertices of $P$. A vertex is a point in the polytope that cannot be expressed as the convex combination of other points in the polytope.

Every binary word $\{0, 1\}^n$ is the vertex of $[0, 1]^n$. Hence every codeword $y \in C$ is a vertex of $P$.

$$C \subseteq V(P) \subseteq P \subseteq [0, 1]^n$$
Define the cost of a point \( f = [f_1, f_2, \ldots, f_n] \in P \) as \( \sum_{i=1}^{n} \gamma_i f_i \).

LP:

\[
\min \sum_{i=1}^{n} \gamma_i f_i \quad \text{s.t.} \quad f \in P
\]

Integral solution: output it as the transmitted codeword

Fractional solution: output error

Note: Given \( P \) proper, if it outputs an integral solution, then the solution is the ML codeword. We call this the **ML certificate property**. This property is one of the unique advantages of LP decoding.
Introduction
LP Decoding for Error-Correcting Codes
LP for LDPC
Others
Fractional Distance
C-Symmetry
Shuai Zhang
Linear Programming Decoding for LDPC
The LP decoder succeeds if the transmitted codeword is the *unique* optimal solution to the LP. (In the case of multiple LP optima, also assume the LP decoder fails.)

\[
\Pr[\text{error}\mid y \text{ transmitted}] = \Pr \left[ \exists f \in P, f \neq y : \sum_i \gamma_i f_i \leq \sum_i \gamma_i y_i \right]
\]
Outline

1 Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2 LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3 LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Message-Passing

4 Others
Given a proper $P$, the *classical distance* is defined as

$$d = \min_{y, y' \in (V(P) \cap \{0, 1\}^n) \atop y \neq y'} \sum_{i=1}^{n} |y_i - y'_i|$$
Given a proper $P$, the *classical distance* is defined as

$$d = \min_{y, y' \in (V(P) \cap \{0, 1\}^n)} \sum_{i=1}^{n} |y_i - y'_i| = \min_{y \in (V(P) \cap \{0, 1\}^n)} \sum_{i=1}^{n} y_i$$
Given a proper $P$, the classical distance is defined as

$$d = \min_{y, y' \in (V(P) \cap \{0,1\}^n), y \neq y'} \sum_{i=1}^{n} |y_i - y'_i| = \min_{y \in (V(P) \cap \{0,1\}^n), y \neq 0^n} \sum_{i=1}^{n} y_i$$

**Definition.** The fractional distance of a proper $P$ is the minimum “Hamming distance” between an integral vertex (codeword) and any other vertex:

$$d_{frac} = \min_{y \in (V(P) \cap \{0,1\}^n), f \in V(P), f \neq y} \sum_{i=1}^{n} |y_i - f_i|$$
Given a proper $P$, the classical distance is defined as

$$d = \min_{y, y' \in (V(P) \cap \{0, 1\}^n) \atop y \neq y'} \sum_{i=1}^{n} |y_i - y'_i| = \min_{y \in (V(P) \cap \{0, 1\}^n) \atop y \neq 0^n} \sum_{i=1}^{n} y_i$$

**Definition.** The fractional distance of a proper $P$ is the minimum “Hamming distance” between an integral vertex (codeword) and any other vertex:

$$d_{frac} = \min_{y \in (V(P) \cap \{0, 1\}^n) \atop f \in V(P) \atop f \neq y} \sum_{i=1}^{n} |y_i - f_i| \leq d$$
**Theorem.** The LP decoder using a proper $P$ is successful if at most $\left\lfloor \frac{d_{frac}}{2} \right\rfloor - 1$ bits are flipped by the BSC.
**Theorem.** The LP decoder using a proper $P$ is successful if at most $\left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped by the BSC.

**Proof.** Let $y$ be the transmitted codeword and $\tilde{y}$ be the received. Suppose the LP decoder fails when $\leq \left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped. $\exists f^* \in V(P), f^* \neq y$ is an solution to the LP.
Theorem. The LP decoder using a proper $P$ is successful if at most $\left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped by the BSC.

Proof. Let $y$ be the transmitted codeword and $\tilde{y}$ be the received. Suppose the LP decoder fails when $\leq \left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped. $\exists f^* \in V(P), f^* \neq y$ is an solution to the LP.

Let $f_i = |f_i^* - y_i| = \begin{cases} f_i^*, & \text{if } y_i = 0 \\ 1 - f_i^*, & \text{if } y_i = 1 \end{cases}$, then

$$\sum_{i=1}^{n} f_i = \sum_{i=1}^{n} |f_i^* - y_i|$$
**Theorem.** The LP decoder using a proper $P$ is successful if at most $\left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped by the BSC.

**Proof.** Let $y$ be the transmitted codeword and $\tilde{y}$ be the received. Suppose the LP decoder fails when at most $\leq \left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped. $\exists f^* \in V(P), f^* \neq y$ is an solution to the LP.

Let $f_i = |f_i^* - y_i| = \begin{cases} f_i^*, & \text{if } y_i = 0 \\ 1 - f_i^*, & \text{if } y_i = 1 \end{cases}$, then

$$\sum_{i=1}^{n} f_i = \sum_{i=1}^{n} |f_i^* - y_i| \geq d_{frac}$$
**Theorem.** The LP decoder using a proper $P$ is successful if at most $\left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped by the BSC.

**Proof.** Let $y$ be the transmitted codeword and $\tilde{y}$ be the received.

Suppose the LP decoder fails when $\leq \left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped. $\exists f^* \in V(P), f^* \neq y$ is an solution to the LP.

Let $f_i = |f_i^* - y_i| = \begin{cases} f_i^*, & \text{if } y_i = 0 \\ 1 - f_i^*, & \text{if } y_i = 1 \end{cases}$, then

$$\sum_{i=1}^{n} f_i = \sum_{i=1}^{n} |f_i^* - y_i| \geq d_{frac}$$

Let $E = \{i | \tilde{y}_i \neq y_i\}$, then $\sum_{i \in E} f_i \leq |E|$
Theorem. The LP decoder using a proper $P$ is successful if at most $\left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped by the BSC.

Proof. Let $y$ be the transmitted codeword and $\tilde{y}$ be the received.

Suppose the LP decoder fails when $\leq \left\lceil \frac{d_{frac}}{2} \right\rceil - 1$ bits are flipped. $\exists f^* \in V(P), f^* \neq y$ is an solution to the LP.

Let $f_i = |f_i^* - y_i| = \begin{cases} f_i^*, & \text{if } y_i = 0 \\ 1 - f_i^*, & \text{if } y_i = 1 \end{cases}$, then

$$\sum_{i=1}^{n} f_i = \sum_{i=1}^{n} |f_i^* - y_i| \geq d_{frac}$$

Let $E = \{i | \tilde{y}_i \neq y_i\}$, then $\sum_{i \in E} f_i \leq |E| \leq \left\lceil \frac{d_{frac}}{2} \right\rceil - 1$
\[ \sum_{i=1}^{n} f_i = \sum_{i \in E} f_i + \sum_{i \notin E} f_i \geq d_{frac} \]
\[
\sum_{i=1}^{n} f_i = \sum_{i \in E} f_i + \sum_{i \not\in E} f_i \geq d_{\text{frac}}
\]

\[
\sum_{i \not\in E} f_i \geq d_{\text{frac}} - \sum_{i \in E} f_i
\]

\[
\geq d_{\text{frac}} - \left\lceil \frac{d_{\text{frac}}}{2} \right\rceil + 1
\]

\[
\geq \left\lceil \frac{d_{\text{frac}}}{2} \right\rceil + 1
\]

\[
> \sum_{i \in E} f_i
\]

\[
\sum_{i \not\in E} f_i - \sum_{i \in E} f_i > 0
\]
Since \( f^* \) is an solution: 
\[
\sum_{i=1}^{n} \gamma_i f_i^* \leq \sum_{i=1}^{n} \gamma_i y_i
\]
\[
\sum_{i=1}^{n} \gamma_i f_i^* - \sum_{i=1}^{n} \gamma_i y_i = \sum_{i:y_i=0}^{n} \gamma_i (f_i^* - y_i)
\]
\[
= \sum_{i:y_i=0}^{n} \gamma_i f_i^* - \sum_{i:y_i=1}^{n} \gamma_i (1 - f_i^*)
\]
\[
= \sum_{i:y_i=0}^{n} \gamma_i f_i - \sum_{i:y_i=1}^{n} \gamma_i f_i
\]
\[
= \left( \sum_{i:y_i=0, \tilde{y}_i=0} f_i - \sum_{i:y_i=0, \tilde{y}_i=1} f_i \right) - \left( \sum_{i:y_i=1, \tilde{y}_i=0} f_i - \sum_{i:y_i=1, \tilde{y}_i=1} f_i \right)
\]
\[
= \left( \sum_{i:y_i=0, \tilde{y}_i=0} f_i + \sum_{i:y_i=1, \tilde{y}_i=1} f_i \right) - \left( \sum_{i:y_i=1, \tilde{y}_i=0} f_i + \sum_{i:y_i=0, \tilde{y}_i=1} f_i \right)
\]
\[
= \sum_{i \notin E} f_i - \sum_{i \in E} f_i
\]
\[
\leq 0 \quad \blacksquare
\]
Outline

1 Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2 LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3 LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Massage-Passing

4 Others
To have the all-zeros assumption, we need this concept.
To have the all-zeros assumption, we need this concept.

**Definition.**

For a point $f \in [0, 1]^n$, define its relative point $f^y \in [0, 1]^n$ with respect to codeword $y$ as follows:

$$\forall i \in \{1, \ldots, n\}, f_i^y = |f_i - y_i|$$
To have the all-zeros assumption, we need this concept.

**Definition.**

- For a point $f \in [0, 1]^n$, define its relative point $f[y] \in [0, 1]^n$ with respect to codeword $y$ as follows:

$$\forall i \in \{1, \ldots, n\}, f_i[y] = |f_i - y_i|$$

- A proper $P$ for a binary code $C$ is **$C$-symmetric** if $\forall f \in V(P)$ and $\forall y \in C$, the relative point $f[y] \in P$. 
Theorem. Let $P$ be $C$-symmetric. Then, $\forall f \in V(P)$ and $\forall y \in C$, the relative point $f[y] \in V(P)$. 
**Theorem.** Let $P$ be $C$-symmetric. Then, $\forall f \in V(P)$ and $\forall y \in C$, the relative point $f^{[y]} \in V(P)$.

**Theorem.** The fractional distance of a $C$-symmetric polytope $P$ for a binary linear code $C$ is equal to the minimum weight of non-zero vertices of $P$:

$$d_{frac} = \min_{f \in (V(P) \setminus 0^n)} \sum_{i=1}^{n} f_i$$
Theorem. Let $P$ be $C$-symmetric. Then, $\forall f \in V(P)$ and $\forall y \in C$, the relative point $f[y] \in V(P)$.

Theorem. The fractional distance of a $C$-symmetric polytope $P$ for a binary linear code $C$ is equal to the minimum weight of non-zero vertices of $P$:

$$d_{frac} = \min_{f \in (V(P) \setminus \{0^n\})} \sum_{i=1}^{n} f_i$$

The fractional distance of a $C$-symmetric polytope $P$ for a binary linear code $C$ can be computed efficiently.
**Theorem.** For any LP decoder using a C-symmetric polytope under a binary-input memoryless symmetric channel, the probability that the LP decoder fails is independent of the codeword that is transmitted.
All-Zeros Assumption

\[
\text{Pr}[\text{error}|y \text{ transmitted}] = \text{Pr} \left[ \exists f \in P, f \neq y : \sum_i \gamma_i f_i \leq \sum_i \gamma_i y_i \right]
\]
All-Zeros Assumption

\[
\Pr[\text{error}|y \text{ transmitted}] = \Pr \left[ \exists f \in P, f \neq y : \sum_i \gamma_i f_i \leq \sum_i \gamma_i y_i \right]
\]

\[
\Pr[\text{error}|y \text{ transmitted}] = \Pr[\text{error}|0^n \text{ transmitted}]
\]

\[
= \Pr \left[ \exists f \in P, f \neq 0^n : \sum_i \gamma_i f_i \leq 0 \right]
\]
All-Zeros Assumption

\[
\Pr[\text{error}|y \text{ transmitted}] = \Pr \left[ \exists f \in P, f \neq y : \sum_i \gamma_i f_i \leq \sum_i \gamma_i y_i \right]
\]

\[
\Pr[\text{error}|y \text{ transmitted}] = \Pr[\text{error}|0^n \text{ transmitted}]
\]

\[
= \Pr \left[ \exists f \in P, f \neq 0^n : \sum_i \gamma_i f_i \leq 0 \right]
\]

**Corollary.** For any binary linear code \(C\) over any binary-input memoryless symmetric channel, the LP decoder using \(C\)-symmetric polytope \(P\) will fail iff there is some non-zero point in \(P\) with cost less than or equal to zero, given \(0^n\) is transmitted.
Outline

1. Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2. LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3. LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Massage-Passing

4. Others
Notations

1. $C$: a linear code
2. $H = [h_{ji}]_{(n-k) \times n}$: the parity check matrix
3. $G$: the factor graph
4. $I$: the set of variable nodes
5. $J$: the set of check nodes
6. $N(j) = \{i \in I | h_{ji} = 1\}$: the set of neighbor variable nodes of a check node $j \in J$
7. $\deg_l^+$: the maximum variable degree
8. $\deg_l^-$: the minimum variable degree
9. $\deg_r^+$: the maximum check degree
10. $\deg_r^-$: the minimum check degree
Outline

1 Introduction
   • ML Decoding
   • Linear Programming
   • LP Relaxation

2 LP Decoding for Error-Correcting Codes
   • Fractional Distance
   • C-Symmetry

3 LP for LDPC
   • LP Polytope for LDPC
   • $d_{frac}$
   • Comparison with Massage-Passing

4 Others

Shuai Zhang

Linear Programming Decoding for LDPC
Observation

Each check node defines a “local code”.

\[ E_j = \{S \subseteq N(j) : |S| \text{ even}\} \]: each such \( S \) corresponds to a local codeword set, defined by setting

\[
y_i = \begin{cases} 
1, & \text{if } i \in S \\
0, & \text{if } i \in (N(j) \setminus S) \\
0 \text{ or } 1, & \text{if } i \in (I \setminus N(j)) 
\end{cases}
\]
Observation

Each check node defines a “local code”. 

\[ E_j = \{ S \subseteq N(j) : |S| \text{ even} \} : \text{each such } S \text{ corresponds to a local codeword set, defined by setting} \]

\[ y_i = \begin{cases} 
1, & \text{if } i \in S \\
0, & \text{if } i \in (N(j) \setminus S) \\
0 \text{ or } 1, & \text{if } i \in (I \setminus N(j)) 
\end{cases} \]

The global code corresponds to the intersection of all the local codes, i.e., defined by all check nodes.

In LP terminology, each check node defines a “local polytope”: the set of convex combinations of local codes. The global polytope will be the intersection of all these local polytopes.
• Code bits: \((f_1, f_2, \ldots, f_n), 0 \leq f_i \leq 1\)

• An auxiliary variable \(w_{j,S}\), which indicate that the codeword satisfies check node \(j\) using the configuration \(S\):

\[
    w_{j,S} = \begin{cases} 
        1, & \text{use the configuration } S \\
        0, & \text{not use the configuration } S 
    \end{cases}
\]

and must satisfy \(\forall S \in E_j, 0 \leq w_{j,S} \leq 1\).

In addition, in a codeword, each parity check is satisfied with one particular even-sized subset of nodes in its neighborhood set to one: \(\sum_{S \in E_j} w_{j,S} = 1\).

• The indicator \(f_i\) at each variable node \(i\) must be consistent with the point in the local codeword polytope defined by \(w = [w_{j,S_1}, w_{j,S_2}, \ldots], S_k \in E_j\) for check node \(j\):

\[
    \forall i \in N(j), f_i = \sum_{S \in E_j, S \ni i} w_{j,S}
\]
For each $j \in J$, the local polytope

$$Q_j = \left\{ (f, w) \mid \begin{array}{l}
0 \leq f_i \leq 1, \quad \forall i \in I; \\
0 \leq w_{j,S} \leq 1, \quad \forall S \in E_j; \\
\sum_{S \in E_j} w_{j,S} = 1; \\
f_i = \sum_{S \in E_j, S \ni i} w_{j,S}, \quad \forall i \in N(j). 
\end{array} \right\}$$

The global polytope:

$$Q = \bigcap_{j \in J} Q_j$$
LP for LDPC

\[ \min \sum_{i=1}^{n} \gamma_i f_i \quad \text{s.t.} \quad (f, w) \in Q \]
LP for LDPC

\[
\min \sum_{i=1}^{n} \gamma_i f_i \quad \text{s.t.} \quad (f, w) \in Q
\]

Note: \( Q \) contains auxiliary variables \( w \), so consider the projection of \( Q \) onto the subspace defined by:

\[
Proj(Q) = \{ f | \exists w \text{ s.t. } (f, w) \in Q \} = \bigcap_{j} Proj(Q_j)
\]

where

\[
Proj(Q_j) = \{ f | \exists w \text{ s.t. } (f, w) \in Q_j \}
\]
\[
\min \sum_{i=1}^{n} \gamma_i f_i \quad \text{s.t.} \quad (f, w) \in Q
\]

Note: \(Q\) contains auxiliary variables \(w\), so consider the projection of \(Q\) onto the subspace defined by:

\[
\text{Proj}(Q) = \{f | \exists w \text{ s.t. } (f, w) \in Q\} = \bigcap_{j} \text{Proj}(Q_j)
\]

where

\[
\text{Proj}(Q_j) = \{f | \exists w \text{ s.t. } (f, w) \in Q_j\}
\]

\[
\min \sum_{i=1}^{n} \gamma_i f_i \quad \text{s.t.} \quad (f, w) \in Q \iff \min \sum_{i=1}^{n} \gamma_i f_i \quad \text{s.t.} \quad f \in \text{Proj}(Q)
\]
Properness and C-Symmetry
**Theorem.** $Proj(Q)$ is proper: $Proj(Q) \cap \{0, 1\}^n = C$.

So LP based on $Proj(Q)$ has the ML certificate property.
**Theorem.** \( \text{Proj}(Q) \) is proper: \( \text{Proj}(Q) \cap \{0, 1\}^n = C. \)

So LP based on \( \text{Proj}(Q) \) has the ML certificate property.

**Theorem.** \( \text{Proj}(Q) \) is \( C \)-symmetric.

So may assume all-zeros codeword.
**Theorem.** \( \text{Proj}(Q) \) is proper: \( \text{Proj}(Q) \cap \{0,1\}^n = C \).

So LP based on \( \text{Proj}(Q) \) has the ML certificate property.

**Theorem.** \( \text{Proj}(Q) \) is C-symmetric.

So may assume all-zeros codeword.

**Corollary.** Given that the all-zeros codeword was transmitted, the LP decoder using \( Q \) will fail iff \( \exists (f,w) \in Q \) with cost less than or equal to zero, where \( f \neq 0^n \).
Outline

1. Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2. LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3. LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Massage-Passing

4. Others
Performance Bound

**Theorem.**

- \( \text{Proj}(Q) \) is proper, so the LP decoder using \( \text{Proj}(Q) \) can correct up to \( \left\lfloor \frac{d_{\text{frac}}}{2} \right\rfloor - 1 \) errors in the BSC;
- \( \text{Proj}(Q) \) is C-symmetric, so

\[
d_{\text{frac}} = \min_{f \in (V(\text{Proj}(Q)) \setminus 0^n)} \sum_{i=1}^{n} f_i
\]
Algorithm: \[ \min \sum_{i=1}^{n} f_i \text{ where } f \in (V(Proj(Q)) \setminus 0^n). \]

1. Select \( x^0 \in (V(Proj(Q)) \setminus 0^n) \) and let \( F \subset Proj(Q) \) be the set of constraints that \( x^0 \) is not tight (we say a point is tight for a constraint if it meets the constraint with equality).
2. For each constraint in \( F \): define \( P' \) by making the constraint into an equality constraint.
3. \[ \min \sum_{i=1}^{n} f_i \text{ over } P' \text{ by LP solver (Simplex). Note that } x^0 \notin P'. \]
4. The minimum value obtained over all constraints in \( F \) is the minimum of \[ \sum_{i=1}^{n} f_i \text{ over all vertices other than } x^0. \]
Algorithm: \( \min \sum_{i=1}^{n} f_i \) where \( f \in (V(\text{Proj}(Q)) \setminus 0^n) \).

1. Select \( x^0 \in (V(\text{Proj}(Q)) \setminus 0^n) \) and let \( F \subset \text{Proj}(Q) \) be the set of constraints that \( x^0 \) is not tight (we say a point is tight for a constraint if it meets the constraint with equality).

2. For each constraint in \( F \): define \( P' \) by making the constraint into an equality constraint.

3. \( \min \sum_{i=1}^{n} f_i \) over \( P' \) by LP solver (Simplex). Note that \( x^0 \not\in P' \).

4. The minimum value obtained over all constraints in \( F \) is the minimum of \( \sum_{i=1}^{n} f_i \) over all vertices other than \( x^0 \).

Example on board.
Average Fractional Distance: Random (3,4) LDPC Code

- Upper, lower quartiles
- Average
- Least squares curve
- Max

Fractional Distance vs Code Length
Earlier result on classical distance (Tanner '81):

\[
D \geq \begin{cases} 
    d \left( \frac{(d-1)(m-1)}{(d-1)(m-1)-1} \right)^{\frac{g-2}{4}} - 1 + \frac{d}{m} \left[ (d - 1)(m - 1) \right]^{\frac{g-2}{4}}, & \text{if } \frac{g}{2} \text{ odd} \\
    d \left( \frac{(d-1)(m-1)}{4} \right)^{\frac{g}{4}} - 1, & \text{if } \frac{g}{2} \text{ even}
\end{cases}
\]
Lower Bound of $d_{frac}$

Earlier result on classical distance (Tanner '81):

$$D \geq \begin{cases} 
\frac{d [(d-1)(m-1)]^{\frac{g-2}{4}} - 1}{(d-1)(m-1) - 1} + \frac{d}{m} [(d - 1)(m - 1)]^{\frac{g-2}{4}}, & \text{if } \frac{g}{2} \text{ odd} \\
\frac{d [(d-1)(m-1)]^{\frac{g}{4}} - 1}{(d-1)(m-1) - 1}, & \text{if } \frac{g}{2} \text{ even}
\end{cases}$$

**Theorem.** Let $G$ be a factor graph with $deg_l^- \geq 3$ and $deg_r^- \geq 2$. Let $g$ be the girth and $g > 4$. Then $d_{frac}$ of $Proj(Q)$ is at least

$$d_{frac} \geq (deg_l^- - 1)^{\left\lceil \frac{g}{4} \right\rceil - 1}$$
Outline

1. Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2. LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3. LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Massage-Passing

4. Others
In the BEC, a set $S$ of variable nodes is a *stopping set* if all their corresponding bits are erased by the channel, and the checks in the neighborhood of $S$ all have degree at least two with respect to $S$.

In the BEC, massage-passing fails iff a stopping set exists. (Di, Proietti, Richardson, Telatar, Urbanke, ’02)
Binary Erasure Channel

In the BEC, a set $S$ of variable nodes is a *stopping set* if all their corresponding bits are erased by the channel, and the checks in the neighborhood of $S$ all have degree at least two with respect to $S$.

In the BEC, massage-passing fails iff a stopping set exists. (Di, Proietti, Richardson, Telatar, Urbanke, ’02)

**Theorem.** The performance of the LP decoder is equivalent to belief propagation on the BEC.
The sum-product algorithm performs very well in practice. However, it will not always converge; additionally it does not have the ML certificate property.

Note:

- In practice (for LDPC codes with large block length), when the sum-product decoder converges to a codeword, it is extremely rare for it not to be the ML codeword (Forney ’03).
- May make use of *LP duality* to give message-passing algorithms the ML certificate.
Comparing the computational complexity of LP decoding and message-passing decoding is still very much an open issue, both in theory and practice.

Theoretically, the LP decoder is far less efficient than the message-passing decoders, most of which run in linear time (for a fixed number of iterations). Intuitively, this is because that one would have to “pay” computationally for the ML certificate property.
LP vs min-sum

WER Comparison: Random Rate-1/2 (3,6) LDPC Code

Word Error Rate vs BSC Crossover Probability
min-sum & LP & sum-product & ML

WER Comparison: Random Rate-1/4 (3,4) LDPC Code

Word Error Rate vs. BSC Crossover Probability

Min-Sum Decoder
LP Decoder
Sum-Product Decoder
ML Decoder
Outline

1 Introduction
   - ML Decoding
   - Linear Programming
   - LP Relaxation

2 LP Decoding for Error-Correcting Codes
   - Fractional Distance
   - C-Symmetry

3 LP for LDPC
   - LP Polytope for LDPC
   - $d_{frac}$
   - Comparison with Message-Passing

4 Others
Tighter Relaxation - Redundant Parity Checks

Add new check nodes $k$: $N(k) = N(j_1) \oplus N(j_2)$ where $j_1, j_2 \in J, j_1 \neq j_2$.

Such $k$ is called second order parity checks.

If there are $|J| = n - k = m$ original check nodes (first order), then there are at most $\binom{m}{2}$ second order parity checks, and the polytope gets smaller but more complex.
Tighter Relaxation - Redundant Parity Checks

WER Comparison: Random Rate-1/4 (3,4) LDPC Code

- First-Order LP Decoder
- Second-Order LP Decoder

Shuai Zhang

Linear Programming Decoding for LDPC
Future Work

- Tighter relaxations.
- Strengthen the lower bound of $d_{frac}$.
- Deeper connections to sum-product.
- There may be more efficient methods for solving the coding LPs than using general-purpose LP algorithms.
- Irregular LDPC codes.
- LP for Turbo Codes: directed Trellis diagram (making use of Network flow), or keep using the undirected graph.
Main References

Jon Feldman. 

Changyan Di; Proietti D.; Telatar I.E.; Richardson T.J.; Urbanke R.L.

R. Tanner.
Thank you!