Optimization-Based Approaches to Decoding Linear Codes

Mohammad H. Taghavi    Paul H. Siegel
(in collaboration with Rüdiger Urbanke)
mtaghavi@ucsd.edu    psiegel@ucsd.edu

University of California, San Diego

Jan, 2008
Motivation

- Linear programming (LP) decoding was introduced by Feldman, Wainwright, and Karger [2003] for decoding linear codes.
- Has a performance close to those of iterative message-passing (IMP) decoders.
  - There are theoretical connections between LP and IMP decoders.
- Advantages of LP decoding
  - More flexible for finite-length analysis due to the geometric structure (e.g., convex decision regions)
  - Potential for improvement
  - Detectable failures: ML certificate property
- Standard LP decoding is significantly more complex than IMP decoding.
  - Large problem size
  - Inefficiency of general-purpose LP solvers
Outline

1. Introduction: Linear Programming Decoding
2. Adaptive LP Decoding
3. A Message-Passing Solver for Adaptive LP Decoding
4. Conclusion
Outline

1. Introduction: Linear Programming Decoding
2. Adaptive LP Decoding
3. A Message-Passing Solver for Adaptive LP Decoding
4. Conclusion
Binary Linear Codes on Graphs

- A binary linear code is defined as \( C = \{ c \in \mathbb{F}_2^n | Hc = 0 \} \)

- \( H_{m \times n} \) is the parity-check matrix.
  - For low-density parity-check (LDPC) codes, \( H \) is sparse

- Tanner graph representation
  - Variable nodes \( \{ i = 1, \ldots, n \} \) and check nodes \( \{ j = 1, \ldots, m \} \)
  - Neighborhood \( N(j) \) and degree \( \text{deg}(j) \)

- Example

\[
H = \begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 0
\end{bmatrix}
\]

\( x_1 + x_3 + x_6 = 0 \mod 2 \)
Linear Relaxation of ML Decoding

- ML decoder finds the codeword $\hat{c}$ that maximizes the likelihood of the received vector, $\Pr[r|c]$. Equivalently:

  Minimize $\gamma^T c$

  Subject to $c \in C$

  where $\gamma_i = \log \left( \frac{\Pr(r_i|c_i = 0)}{\Pr(r_i|c_i = 1)} \right)$

- Linear Relaxation:
  - Replace the code space

    $\mathcal{C} = \left\{ x \in \{0, 1\}^n \left| \sum_{i \in N(j)} x_i = 0 \text{ mod } 2, \forall j = 1, \ldots, m \right. \right\}$

    by the Fundamental Polytope

    $\mathcal{P} = \left\{ x \in [0, 1]^n \left| \sum_{i \in V} (1 - x_i) + \sum_{i \in N(j) \setminus V} x_i \geq 1, \forall V \subseteq N(j) \text{ s.t. } |V| \text{ is odd, } \forall j = 1, \ldots, m \right. \right\}$
Linear Relaxation of ML Decoding

ML decoder finds the codeword \( \hat{c} \) that maximizes the likelihood of the received vector, \( \Pr[r|c] \). Equivalently:

\[
\begin{align*}
\text{Minimize} & \quad \gamma^T c \\
\text{Subject to} & \quad c \in C
\end{align*}
\]

where \( \gamma_i = \log \left( \frac{\Pr(r_i|c_i = 0)}{\Pr(r_i|c_i = 1)} \right) \)

Linear Relaxation:

- Replace the code space

\[
C = \left\{ x \in \{0, 1\}^n \mid \sum_{i \in N(j)} x_i = 0 \mod 2, \ \forall j = 1, \ldots, m \right\}
\]

by the Fundamental Polytope

\[
P = \left\{ x \in [0, 1]^n \mid \sum_{i \in V} (1 - x_i) + \sum_{i \in N(j) \setminus V} x_i \geq 1, \right. \\
\left. \forall V \subseteq N(j) \text{ s.t. } |V| \text{ is odd, } \forall j = 1, \ldots, m \right\}
\]
Linear Programming Decoding

- **LP Decoding:**
  
  \[
  \text{Minimize} \quad \gamma^T x \\
  \text{Subject to} \quad x \in \mathcal{P}
  \]

- The solution is one of the vertices of the polytope.

- Each check node of degree \(d\) is replaced by \(2^{d-1}\) linear constraints.
  - Problem size \(\propto m2^{d_{\text{max}}}\)

- Alternative representation by Chertkov-Stepanov [2007], and Yang-Wang-Feldman [2007]
  - Replace each check node by a tree of auxiliary variable nodes and degree-3 check nodes
  - Problem size \(\propto md_{\text{max}}\)
**Linear Programming Decoding**

- **LP Decoding:**
  
  \[ \text{Minimize } \gamma^T x \]
  
  \[ \text{Subject to } x \in P \]

  - The solution is one of the vertices of the polytope.
  - Each check node of degree \( d \) is replaced by \( 2^{d-1} \) linear constraints.
    - Problem size \( \propto m2^{d_{\max}} \)
  - Alternative representation by Chertkov-Stepanov [2007], and Yang-Wang-Feldman [2007]
    - Replace each check node by a tree of auxiliary variable nodes and degree-3 check nodes
    - Problem size \( \propto md_{\max} \)
Outline

1. Introduction: Linear Programming Decoding
2. Adaptive LP Decoding
3. A Message-Passing Solver for Adaptive LP Decoding
4. Conclusion
Do We Need All the Constraints to Decode?

Definition
A constraint $a^T x \leq b$ is **active** at point $x_0$ if $a^T x_0 = b$, and is a **cut** if $a^T x_0 > b$.

Theorem
At any given point $x \in [0, 1]^n$, at most one of the constraints introduced by each parity check can be a cut.

- We can find all the cuts in $O(md_{max})$ time.
Do We Need All the Constraints to Decode?

**Definition**

A constraint $a^T x \leq b$ is *active* at point $x_0$ if $a^T x_0 = b$, and is a *cut* if $a^T x_0 > b$.

**Theorem**

At any given point $x \in [0, 1]^n$, *at most one* of the constraints introduced by each parity check can be a cut.

- We can find all the cuts in $O(md_{max})$ time.
Adaptive LP Decoding

- Start with a minimal problem and add the constraints adaptively.

**Algorithm**

1. Set up the initial problem with only n simple constraints; \( k \leftarrow 0 \).
2. Run the LP solver and find the solution \( x^{(k)} \) to the current problem; \( k \leftarrow k + 1 \).
3. Find all constraints that generate cuts at \( x^{(k)} \) and add them to the problem.
4. If no cuts were found, \( x^{(k)} \) is the LP decoding output: Exit; otherwise, go to step 2.

**Theorem**

The above algorithm converges in at most n iterations.

- In practice, the maximum number of iterations is constant.
- Only a very small fraction of the original constraints will be used to obtain the solution to the LP decoding problem.
  - Less than two constraints per parity check.
Random regular LDPC codes of length 360 and rate $\frac{1}{2}$.

$SNR = -1$ dB $\leftarrow$ worst-case behavior
Low SNR Simulations: Decoding Time vs. Length

- Use *warm starts* to speed up the LP solver.

- \((3, 6)\)-regular and \((4, 8)\)-regular LDPC codes
- \(\text{SNR} = -1\) dB.
- Simplex algorithm, GNU Linear Programming Kit.
- Decoding time does not change significantly with the code density.
Simulations: Complexity vs. SNR

- Random (3, 6)-regular LDPC code of length 240, with 2400 trials for each point.
  - Solid lines: mean
  - Dashed lines: 95% confidence intervals
  - Solid marked lines: maximum and minimum
1 Introduction: Linear Programming Decoding
2 Adaptive LP Decoding
3 A Message-Passing Solver for Adaptive LP Decoding
4 Conclusion
A Message-Passing LP Decoder

- We have an LP with a set of (linear) constraints \( \kappa_1, \ldots, \kappa_s \).
- Reformulate the LP as

\[
\text{minimize } F(x) = \sum_{i=1}^{n} \gamma_i x_i - \sum_{j=1}^{s} \log(1_{\{\kappa_j\}})
\]

- Use the [functional] Min-Sum Algorithm (MSA) to solve this on a factor graph.
  - Assign a \textit{variable node} to each \( x_i \) and a \textit{constraint node} to each \( \kappa_j \).
  - The messages are continuous functions over \([0, 1]\).

**Lemma**

The messages at each iteration are \textit{linear functions} of \( x_i \).

- It is enough to only compute and exchange the slopes of these functions.
- Challenge: Multiple constraints derived from the same parity-check create many 4-cycles.
Solution: Revised Adaptive LP Decoding Scheme

Algorithm (Revised Adaptive LP Decoding)

1. Set up the initial problem with only \( n \) simple constraints; \( k \leftarrow 0 \).
2. Run the LP solver and find the solution \( x^{(k)} \) to the current problem; \( k \leftarrow k + 1 \).
3. Remove all the constraints that are not active at \( x^{(k)} \).
4. Find all constraints that generate cuts at \( x^{(k)} \) and add them to the problem.
5. If no cuts were found, \( x^{(k)} \) is the LP decoding output: Exit; otherwise, go to step 2.

Theorem

- At each iteration, the LP problem contains at most one linear constraint derived from each parity check.

Corollary: Each LP decoding pseudo-codeword has at most \( m \) fractional elements. (\( m \) is the number of check nodes)
Solution: Revised Adaptive LP Decoding Scheme

Algorithm (Revised Adaptive LP Decoding)

1. Set up the initial problem with only \( n \) simple constraints; \( k \leftarrow 0 \).
2. Run the LP solver and find the solution \( x^{(k)} \) to the current problem; \( k \leftarrow k + 1 \).
3. Remove all the constraints that are not active at \( x^{(k)} \).
4. Find all constraints that generate cuts at \( x^{(k)} \) and add them to the problem.
5. If no cuts were found, \( x^{(k)} \) is the LP decoding output: Exit; otherwise, go to step 2.

Theorem

- At each iteration, the LP problem contains at most one linear constraint derived from each parity check.

Corollary: Each LP decoding pseudo-codeword has at most \( m \) fractional elements. (\( m \) is the number of check nodes)
A (3,6)-regular LDPC code of length 120.
Outline

1. Introduction: Linear Programming Decoding
2. Adaptive LP Decoding
3. A Message-Passing Solver for Adaptive LP Decoding
4. Conclusion
Conclusion

- An adaptive technique for LP decoding.
  - Motivated by the properties of the LP relaxation.
  - Solves a hierarchy of much smaller problems.
  - The decoding time is reduced by orders of magnitude.

- Min-sum algorithm for linear programming
  - Fast and parallel implementation.
  - Density evolution can be used for deriving bounds for the asymptotic behavior.
Conclusion

- An adaptive technique for LP decoding.
  - Motivated by the properties of the LP relaxation.
  - Solves a hierarchy of much smaller problems.
  - The decoding time is reduced by orders of magnitude.

- Min-sum algorithm for linear programming
  - Fast and parallel implementation.
  - Density evolution can be used for deriving bounds for the asymptotic behavior.
Extensions and Related Works

- Some complexity gain by a primal-dual interior-point implementation.
- Some Performance gain by adaptively adding extra constraints based on redundant parity checks.
- A new graphical model for decoding in the presence of ISI based on linear relaxation.

Outlook

- Use the hierarchy of Adaptive LP subproblems to derive performance bounds for LP decoding.
- Further study the connection between LP and message-passing decoders.
- Improving interior-point implementations.
Conclusion, cont.

- Extensions and Related Works
  - Some complexity gain by a primal-dual interior-point implementation.
  - Some performance gain by adaptively adding extra constraints based on redundant parity checks.
  - A new graphical model for decoding in the presence of ISI based on linear relaxation.

- Outlook
  - Use the hierarchy of Adaptive LP subproblems to derive performance bounds for LP decoding.
  - Further study the connection between LP and message-passing decoders.
  - Improving interior-point implementations.