LP Decoding

Jon Feldman*
Industrial Engineering and Operations Research
Columbia University, New York, NY, 10027
jonfeld@ieor.columbia.edu

David R. Karger†
Laboratory for Computer Science
MIT, Cambridge, MA, 02139
karger@theory.lcs.mit.edu

Martin J. Wainwright
Electrical Engineering and Computer Science
UC Berkeley, CA, 94720
wainwrig@eecs.berkeley.edu

Abstract. Linear programming (LP) relaxation is a common technique used to find good solutions to complex optimization problems. We present the method of "LP decoding": applying LP relaxation to the problem of maximum-likelihood (ML) decoding. An arbitrary binary-input memoryless channel is considered. This treatment of the LP decoding method places our previous work on turbo codes [6] and low-density parity-check (LDPC) codes [8] into a generic framework. We define the notion of a *proper* relaxation, and show that any LP decoder that uses a proper relaxation exhibits many useful properties. We describe the notion of *pseudocodewords* under LP decoding, unifying many known characterizations for specific codes and channels. The *fractional distance* of an LP decoder is defined, and it is shown that LP decoders correct a number of errors equal to half the fractional distance. We also discuss the application of LP decoding to binary linear codes. We define the notion of a relaxation being *symmetric* for a binary linear code. We show that if a relaxation is symmetric, one may assume that the all-zeros codeword is transmitted.

1 Introduction

The problem of maximum-likelihood (ML) decoding is to find the codeword most likely to have been transmitted, given a corrupted codeword from a noisy channel. Linear programming is the problem of finding an optimal solution to a system of linear inequalities under a linear objective function [2]. In this paper, we consider linear programming (LP) formulations of the ML decoding problem on binary codes. We use LP variables to represent code bits, and the LP objective function is defined by the channel likelihood ratios.

Previous work on LP decoding [6, 4, 7, 8, 5] has focused on two specific cases: *turbo codes* [1] and *low-density parity-check codes* [11]. These two families of codes have received a lot of attention recently due to their excellent performance. Performance bounds for LP decoding in these cases are for specific LP formulations, code constructions, and/or channel models.

In this paper we consider LP decoders for arbitrary binary codes, under an arbitrary binary-input memoryless channel. We provide a framework for designing LP decoders, and general techniques for analyzing them. Central to every LP decoder is its associated *polytope*: the set of points that satisfy the constraints of the LP. A decoding polytope should contain every codeword, and should also exclude every binary word that is not a codeword. We define such polytopes as *proper*. We show that LP decoders that use proper polytopes have the *ML certificate* property: whenever they output a codeword, it is guaranteed to be the ML codeword.

In general, for any sub-optimal decoder, the *pseudocodewords* correspond to the set of possible results of the decoder. This set includes the codewords, but also some non-codewords

*Research supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.
†Research supported by NSF contract CCR-9624239 and a David and Lucille Packard Foundation Fellowship.

that can "fool" the algorithm. In this paper, we provide a general characterization of the pseudocodewords for any LP decoder, under any binary-input memoryless channel. This characterization allows us to give an exact expression for the word error rate (WER) of the decoder: the probability that the transmitted codeword is not the "most likely" pseudocodeword. When applied to specific codes, polytopes and channels, LP pseudocodewords are equivalent to other sets of pseudocodewords that have been studied in previous work. For example, in the binary erasure channel (BEC), when the polytope from [8] is used, the set of LP pseudocodewords is equivalent to the *stopping sets* of the code, as defined by Di et. al [3]. For tail-biting trellises, when the polytope from [5] is used, the set of LP pseudocodewords is equivalent to the pseudocodewords defined by Forney et. al [9].

The *fractional distance* of an LP decoder is defined as an analog to the (classical) distance. It is shown that under the binary symmetric channel (BSC), LP decoders correct a number of errors equal to half the fractional distance of the code. We also discuss the application of LP decoding to binary linear codes. We define the notion of a polytope being *symmetric* for a binary linear code. We show that if a polytope is symmetric, one may assume that the all-zeros codeword is transmitted, which greatly simplifies analysis for this rich class of codes.

1.1 Channel Model. In this paper we assume an arbitrary binary-input memoryless channel; i.e., the data is transmitted as discrete symbols from $\{\mathtt{0}, \mathtt{1}\}$, and each transmitted symbol is affected by the noise in the channel independently. Let $y \in \mathcal{C}$ denote the transmitted codeword. We will use $\tilde{y} = (\tilde{y}_1, \ldots, \tilde{y}_n)$ to denote the received (corrupted) word. Each $\tilde{y}_i$ is a symbol from some space Σ that depends on the channel model. For example, in the binary symmetric channel (BSC), we have $\Sigma = \{0, 1\}$; in the AWGN channel, we have $\Sigma = \mathbb{R}$.

In our analysis of binary linear codes in Section 3, we also assume a symmetric channel; i.e., the noise affects 0s and 1s in the same way. Formally, symmetry tells us that Σ can be partitioned into pairs $(\mathtt{a}, \mathtt{a}')$ such that

$$\begin{aligned} \Pr[\ \tilde{y}_i = \mathtt{a} \mid y_i = 0\ ] &= \Pr[\ \tilde{y}_i = \mathtt{a}' \mid y_i = 1\ ], \text{ and} \quad (1) \\ \Pr[\ \tilde{y}_i = \mathtt{a} \mid y_i = 1\ ] &= \Pr[\ \tilde{y}_i = \mathtt{a}' \mid y_i = 0\ ]. \quad (2) \end{aligned}$$

2 The Method of LP Decoding

2.1 Linear Programming Relaxation. A *linear program* (LP) consists of a set of linear inequalities (constraints) and a linear objective function over a set of variables. Solving the linear program means finding a setting of the variables that satisfies the inequalities, and optimizes the objective function. Linear programs can be solved efficiently using the simplex algorithm [19], which runs efficiently in practice, or the ellipsoid algorithm [12], which has worst-case run-time guarantees.

Although many important problems can be solved as LPs, not all problems are directly amenable to this treatment. One issue is that LP solutions can be real-valued, whereas the variables (in certain problems) may only be meaningful as integers (e.g., number of seats in an airplane). If we add the restriction that all variables must be integers, we obtain an *integer linear programming* (ILP) problem, which (unfortunately) is NP-hard in general.

A natural strategy for finding an *approximate* solution to an ILP, then, is to remove the integer constraints, solve the resulting LP, and then transform the solution into a meaningful one. (For example, rounding techniques, often randomized, are one method of transforming an LP solution into a decent solution to the ILP of interest.) This generic technique is referred to as *linear programming relaxation*, and many successful approximation algorithms to NP-hard optimization problems are based on it [14].

2.2 An LP Relaxation of ML Decoding. Suppose we wish to decode a binary code $\mathcal{C} \subseteq \{0, 1\}^n$ under some binary-input memoryless channel. Let $y \in \mathcal{C}$ denote the transmitted codeword, and let $\tilde{y}$ denote the received codeword. Let γ_i be the *log-likelihood ratio* of the ith code bit:

$$\gamma_i = \ln\left(\frac{\Pr[\tilde{y}_i \mid y_i = 0]}{\Pr[\tilde{y}_i \mid y_i = 1]}\right). \tag{3}$$

The sign of the log-likelihood ratio γ_i determines whether transmitted bit y_i is more likely to be a 0 or a 1. (In particular, if y_i is more likely to be a 1, then γ_i will be negative, whereas if y_i is more likely to be a 0, then γ_i will be positive.) We will refer to γ_i as the *cost* of code bit y_i, where γ_i represents the cost incurred by setting a particular bit y_i to 1, and to the sum $\sum_i \gamma_i y_i$ as the cost of a particular codeword y. With these definitions, the ML codeword is exactly the codeword of minimum cost [5].

Our LP relaxations for decoding will have LP variables f_i for each code bit, where $i \in \{1, \ldots, n\}$. Suppose we were able to solve the following problem:

$$\text{minimize} \sum_{i=1}^{n} \gamma_i f_i \;\; \text{s.t.} \;\; f \in \mathcal{C}.$$

Any optimal solution f to this system is an ML codeword. However, optimizing over $\mathcal{C}$ is too complex in general. Therefore, we optimize instead over a less complex *polytope* $\mathcal{P} \subseteq [0,1]^n$, defined by a set of linear constraints on the variables f_i. The particular nature of the constraints will depend on the underlying code. In previous work [6, 8], we have defined polytopes for turbo codes, LDPC codes, and arbitrary binary linear codes. In each of these cases, our polytopes contain a linear (in n) number of constraints, and are therefore solvable efficiently.

Since we are looking for codewords, it should be the case that our polytope includes all the codewords, and does not include any non-codewords.

Definition 1. *A polytope $\mathcal{P}$ is* proper *for code $\mathcal{C}$ if the integral points in $\mathcal{P}$ are exactly the codewords of $\mathcal{C}$; i.e., $\mathcal{P}$ is proper if $\mathcal{P} \cap \{0,1\}^n = \mathcal{C}$.*

Given a proper polytope $\mathcal{P}$, our LP decoder solves the following linear program:

$$\text{minimize} \sum_{i=1}^{n} \gamma_i f_i \quad \text{s.t.} \quad f \in \mathcal{P} \tag{4}$$

Define the *cost* of a point $f \in \mathcal{P}$ as $\sum_{i=1}^{n} \gamma_i f_i$. The LP in equation (4) will find the point in $\mathcal{P}$ with minimum cost. If the LP solution is integral (i.e., all f_i are either 0 or 1), then the LP decoder outputs the codeword f. In contrast, if the LP solution is fractional (i.e., some f_i is non-integral), then the decoder outputs "error."

Theorem 2. *An LP decoder using a proper polytope has the* ML certificate *property: if the decoder outputs a codeword, it is guaranteed to be an ML codeword.*

Proof. If the LP decoder outputs a codeword $f \in \mathcal{C}$, then the cost of the point $f \in \mathcal{P}$ is at most the cost of any point in $\mathcal{P}$. Since $\mathcal{P}$ is proper, we have $\mathcal{P} \supseteq \mathcal{C}$, and so f has cost at most the cost of any codeword $y \in \mathcal{C}$. We conclude that f is the ML codeword. □

Example. Suppose we have the linear code $\mathcal{C} = \{0000, 1101, 1011, 0110\}$. This code can be characterized by the parity check equations $(y_1 \oplus y_2 \oplus y_3) = 0$ and $(y_2 \oplus y_3 \oplus y_4) = 0$. We define a polytope $\mathcal{R}$ on four variables $\{f_1, f_2, f_3, f_4\}$ as the set of points that satisfy the following linear inequalities:

(A)	(B)	(C)
$f_1 \le f_2 + f_3$	$f_2 \le f_3 + f_4$	$0 \le f_1 \le 1$
$f_2 \le f_1 + f_3$	$f_3 \le f_2 + f_4$	$0 \le f_2 \le 1$
$f_3 \le f_1 + f_2$	$f_4 \le f_2 + f_3$	$0 \le f_3 \le 1$
$f_1 + f_2 + f_3 \le 2$	$f_2 + f_3 + f_4 \le 2$	$0 \le f_4 \le 1$

The (C) constraints ensure that all f_i take on values between zero and one. The (A) and (B) constraints ensure that the polytope $\mathcal{R}$ is proper; i.e., the set of binary words of length four that satisfy the above constraints are exactly the set of codewords of $\mathcal{C}$. To see this, consider the (A) constraints; the binary words that satisfy these constraints are exactly the words that satisfy the parity check equation $(y_1 \oplus y_2 \oplus y_3) = 0$. Similarly, the (B) constraints correspond to the parity check equation $(y_2 \oplus y_3 \oplus y_4) = 0$. This polytope is a special case of a general-purpose polytope for binary linear codes and LDPC codes [8, 5].

2.3 Success Conditions for LP Decoding. Overall, the LP decoder succeeds if the transmitted codeword is the unique optimal solution to the LP. The decoder fails if the transmitted codeword is not an optimal solution to the LP. In the case of multiple LP optima (which for many noise models has zero probability), we will be conservative and assume that the LP decoder fails. Therefore, we have the following theorem.

Theorem 3. *For any binary-input memoryless channel, an LP decoder using polytope $\mathcal{P}$ will fail if and only if there is some point in $\mathcal{P}$ other than the transmitted codeword y with cost less than or equal to the cost of y.*

We use WER_y to denote the word error rate (WER) of the LP decoder, given a particular transmitted codeword y. By Theorem 3, we have:

$$\mathrm{WER}_y = \Pr\left[\exists f \in \mathcal{P}, f \neq y \,:\, \sum_i \gamma_i f_i \leq \sum_i \gamma_i y_i\right] \tag{5}$$

2.4 Vertices, Codewords and Pseudocodewords. An *extreme point*, or equivalently a *vertex* of a polytope is a point that cannot be expressed as the convex combination of other points in the polytope. Let $\mathcal{V}(\mathcal{P})$ be the set of vertices of the polytope $\mathcal{P}$. A fundamental fact of linear programming is that the optimal solution to an LP can always be found at a vertex of the polytope associated with the LP [19]. Therefore, the LP decoder will always find the *lowest cost* vertex of the polytope $\mathcal{P}$.

Theorem 4. *For any polytope $\mathcal{P} \subseteq [0,1]^n$ that is proper for $\mathcal{C}$, every codeword $y \in \mathcal{C}$ is a vertex of $\mathcal{P}$.*

Proof. In the unit hypercube $[0,1]^n$, binary words of length n cannot be expressed as the convex combination of other points in the hypercube. Since $\mathcal{P}$ is contained within the hypercube $[0,1]^n$, we have that all points in $\mathcal{P} \cap \{0,1\}^n$ are vertices of $\mathcal{P}$. Since $\mathcal{P}$ is proper, we have $\mathcal{P} \cap \{0,1\}^n = \mathcal{C}$, and the theorem follows. □

It is important to note that the converse statement (i.e., every polytope vertex is a codeword) may *not* hold, however, since the polytope could have *fractional* (non-integral) vertices. So, in general we have

$$\mathcal{C} \;\subseteq\; \mathcal{V}(\mathcal{P}) \;\subseteq\; \mathcal{P} \;\subseteq\; [0,1]^n.$$

In LP decoding, vertices take on the role of *pseudocodewords*: the set of possible results that a sub-optimal decoder may produce. Pseudocodewords are a superset of the codewords, and may contain "false" codewords that "fool" the algorithm. While the set of codewords is a function of the code itself, the set of pseudocodewords is a function of the sub-optimal decoding algorithm being used.

Understanding the pseudocodewords of a sub-optimal algorithm allows a thorough analysis of its WER. For example, Di et. al [3] exploit the structure of "stopping sets" to analyze the word error rate of iterative decoding on the binary erasure channel. Even and Halabi [4] derive combinatorial theorems about "promenades" (the pseudocodewords of an LP relaxation for rate-1/2 repeat-accumulate codes [6]), and use them to show tight upper and lower bounds on the WER of LP decoding.

Example. Consider the polytope $\mathcal{R}$ designed earlier for the code $\mathcal{C} = \{\texttt{0000}, \texttt{1101}, \texttt{1011}, \texttt{0110}\}$. The vertices of this polytope include the codewords, as well as the fractional vertices $(1, \frac{1}{2}, \frac{1}{2}, 0)$ and $(0, \frac{1}{2}, \frac{1}{2}, 1)$. Note that neither of the fractional vertices can be expressed as convex combinations of codewords. We have $\mathcal{V}(\mathcal{R}) = \big\{(0,0,0,0),(1,1,0,1),(1,0,1,1),(1,1,1,1),(1,\frac{1}{2},\frac{1}{2},0),(0,\frac{1}{2},\frac{1}{2},1)\big\}$. This is the set of *pseudocodewords* for the LP decoder using $\mathcal{R}$ on this code.

Remarks: Pseudocodewords have been studied by Wiberg [21], Forney et. al [9] and Frey et. al [10] as codewords of an iterative decoder computation tree, under min-sum decoding. They have also been studied for tail-biting trellises [9], LDPC codes in the binary erasure channel [3], and Tanner graph covers [15]. In many of these specific cases, LP pseudocodewords are equivalent to (or very related to) previously studied pseudocodeword sets (see [8, 5] for details). It would be interesting to see how the known techniques for analyzing the weights of pseudocodewords could be used to analyze the weights of LP pseudocodewords for other codes and channels.

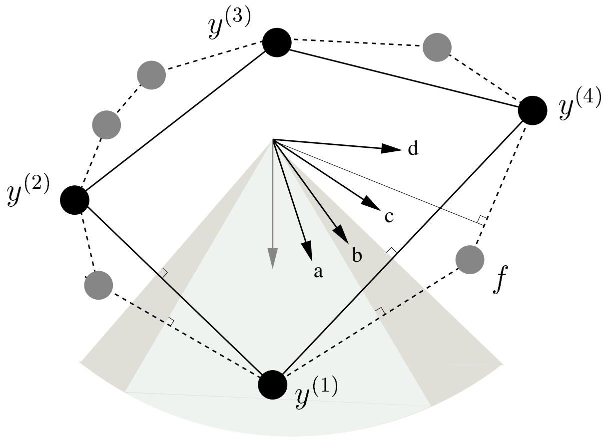

Figure 1. A decoding polytope $\mathcal{P}$ (dotted line) and the convex hull $\mathcal{P}_{\text{ML}}$ (solid line) of the codewords $y^{(1)}$ through $y^{(4)}$. Also shown are the four possible cases (a–d) for the objective function, and the normal cones to both $\mathcal{P}$ and $\mathcal{P}_{\text{ML}}$.

2.5 Geometric Perspective. Figure 1 provides a geometric perspective of LP decoding, and its relation to exact ML decoding. The inner solid line encloses the *convex hull* of the codewords (i.e., the set of points that are convex combinations of codewords), denoted by $\mathcal{P}_{\text{ML}}$. The dotted line in the figure represents the relaxed LP decoding polytope $\mathcal{P}$, and the circles represent pseudocodewords (vertices of $\mathcal{P}$). The black circles in the figure represent codewords (also members of $\mathcal{P}_{\text{ML}}$), whereas the gray circles represent fractional vertices that are not codewords.

The objective function is the only element of the LP that depends on channel noise. An LP objective function can be seen as a *direction* inside the polytope; solving the LP amounts to finding the point in the polytope that is furthest in that direction. If there is no noise in the channel, then the transmitted codeword will be the ML codeword, and thus the lowest cost codeword. The gray arrow in Figure 1 represents the objective function without noise, and it points directly toward the transmitted codeword $y^{(1)}$. Noise in the channel appears in the LP as a perturbation of the objective function away from the "no noise" direction. If the perturbation is small, then $y^{(1)}$ will remain the optimal point of the LP. If the perturbation is large (i.e., high channel noise), then $y^{(1)}$ will no longer be optimal.

Both exact ML and relaxed LP decoding can be seen as minimizing the LP objective specified by the channel, but over different constraint sets. In exact ML decoding, the constraint set is the convex hull $\mathcal{P}_{\mathrm{ML}}$ of codewords, whereas relaxed LP decoding uses the larger polytope $\mathcal{P}$. As a concrete illustration, consider again the set-up of Figure 1, in which codeword $y^{(1)}$ was transmitted. The four arrows labeled (a)–(d) correspond to different "noisy" versions of the LP objective function. (a) If there is very little noise, then both ML decoding and LP decoding succeed, since both have the transmitted codeword $y^{(1)}$ as the optimal point. (b) If more noise is introduced, then ML decoding succeeds, but LP decoding fails, since the fractional vertex f is optimal for the relaxation. (c) With still more noise, ML decoding fails, since $y^{(4)}$ is now optimal; LP decoding still has a fractional optimum f, so this error is detected. (d) Finally, with a lot of noise, both ML decoding and LP decoding have $y^{(4)}$ as the optimum, and so both methods fail and the error is undetected. Note that in the last two cases (c,d), when ML decoding fails, the failure of the LP decoder is in some sense the fault of the code itself, as opposed to the decoder.

2.6 Normal Cones. The behavior of relaxed LP decoding and exact ML decoding can be distinguished in terms of the *normal cones* [13] associated with the LP and ML polytopes at a given codeword $y \in \mathcal{C}$. The (negative) normal cones are defined as follows:

$$\begin{aligned} N_y(\mathcal{P}) &= \Big\{\gamma \in \mathbb{R}^n : \sum_i \gamma_i (f_i - y_i) \geq 0 \ \text{ for all } \ f \in \mathcal{P}\Big\}, \\ N_y(\mathcal{P}_{\mathrm{ML}}) &= \Big\{\gamma \in \mathbb{R}^n : \sum_i \gamma_i (f_i - y_i) \geq 0 \ \text{ for all } \ f \in \mathcal{P}_{\mathrm{ML}}\Big\}. \end{aligned}$$

Note that $N_y(\mathcal{P})$ corresponds to the set of cost vectors γ such that y is an optimal solution to the LP defined by polytope $\mathcal{P}$, and the objective function $\sum_i \gamma_i f_i$. The set $N_y(\mathcal{P}_{\mathrm{ML}})$ has a similar interpretation as the set of cost vectors γ for which y is an ML codeword. Since $\mathcal{P}_{\mathrm{ML}} \subset \mathcal{P}$, it is immediate from the definition that $N_y(\mathcal{P}_{\mathrm{ML}}) \supset N_y(\mathcal{P})$ for all $y \in C$. For example, in Figure 1, cost vector (a) belongs to both $N_{y^{(1)}}(\mathcal{P}_{\mathrm{ML}})$ and $N_{y^{(1)}}(\mathcal{P})$. In contrast, the vector (b) belongs to $N_{y^{(1)}}(\mathcal{P}_{\mathrm{ML}})$, but not to $N_{y^{(1)}}(\mathcal{P})$.

If codeword y is transmitted, the success probability of an LP decoder is equal to the total probability mass of $N_y(\mathcal{P})$, under the distribution on cost vectors defined by the channel. The success probability of ML decoding is similarly related to the probability mass in the normal cone $N_y(\mathcal{P}_{\mathrm{ML}})$. Thus, the discrepancy between the normal cones of $\mathcal{P}$ and $\mathcal{P}_{\mathrm{ML}}$ is a measure of the gap between exact ML and relaxed LP decoding.

The cone $N_y(\mathcal{P})$ can be seen as a "signal-space" characterization of the LP pseudocodewords. Such characterizations have been given by Frey et. al [10] in the case of iterative decoding and by Koetter and Vontobel [15] using the notion of graph covers. In particular, the "fundamental cone" studied by Koetter and Vontobel [15] for graph covers is polar [13] to the normal cone $N_{0^n}(\mathcal{Q})$ associated with the polytope $\mathcal{Q}$ defined in [8] for LDPC codes.

2.7 The Fractional Distance. We motivate the notion of fractional distance by providing an alternative definition for the (classical) distance in terms of a proper polytope $\mathcal{P}$. Recall that any proper polytope $\mathcal{P}$ is characterized by a one-to-one correspondence between codewords and integral vertices of $\mathcal{P}$; i.e., $\mathcal{C} = \mathcal{P} \cap \{0,1\}^n$. The Hamming distance between two points in the discrete space $\{0,1\}^n$ is equivalent to the l_1 distance between the points in the space $[0,1]^n$. Therefore, given a proper polytope $\mathcal{P}$, we may define the distance of a code as the minimum l_1 distance between two integral vertices, i.e.,

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

The LP polytope $\mathcal{P}$ may have additional *non-integral* vertices, as illustrated in Figure 1. Accordingly, we define the *fractional distance* d_{frac} of a polytope $\mathcal{P}$ as the minimum l_1 distance between an integral vertex (codeword) and any *any other* vertex (pseudocodeword) of $\mathcal{P}$; i.e.,

$$d_{frac} \quad = \quad \min_{\substack{y \in \mathcal{C} \\ f \in \mathcal{V}(\mathcal{P}) \\ f \neq y}} \sum_{i=1}^{n} |y_i - f_i|.$$

Note that this fractional distance is always a lower bound on the classical distance of the code, since every codeword is a polytope vertex (in the set $\mathcal{V}(\mathcal{P})$). Moreover, the performance of LP decoding is tied to this fractional distance. The proof of the following theorem is essentially the same as the proof in [8], and so it is omitted. We refer the reader to the thesis [5] for details.

Theorem 5. *Let $\mathcal{C}$ be a binary code and $\mathcal{P}$ a proper polytope in an LP relaxation for $\mathcal{C}$. If the fractional distance of $\mathcal{P}$ is d_{frac}, then the LP decoder using $\mathcal{P}$ is successful if at most $\lceil d_{frac}/2 \rceil - 1$ bits are flipped by the binary symmetric channel.*

3 Symmetric Polytopes for Binary Linear Codes

Binary linear codes have some special algebraic structure that can be exploited in the analysis of decoding algorithms. For example, for most message-passing decoders, one may assume without loss of generality that the all-zeros codeword 0^n, which is always a codeword of a binary linear code, was transmitted. This *all-zeros assumption* greatly simplifies analysis as well as notation. Furthermore, the distance of a binary linear code is equal to the lowest *weight* of any non-zero codeword, where the weight of a codeword y is defined as $\sum_i y_i$.

In this section we discuss the application of LP decoding to binary linear codes. We first define the notion of a polytope being $\mathcal{C}$*-symmetric* for a particular binary linear code $\mathcal{C}$. We then prove that if a decoder uses a $\mathcal{C}$-symmetric polytope, then the all-zeros assumption is valid, and the fractional distance is equal to the lowest weight of any non-zero polytope vertex. This result not only simplifies analysis, but it also allows us to *compute efficiently* the fractional distance of a polytope.

3.1 $\mathcal{C}$-symmetry of a Polytope. For a point $f \in [0,1]^n$, we define its *relative point* $f^{[y]} \in [0,1]^n$ with respect to codeword y as follows: for all $i \in \{1,\ldots,n\}$, let $f_i^{[y]} = |f_i - y_i|$. Note that this operation is its own inverse; i.e., we have $\left(f^{[y]}\right)^{[y]} = f$ for all $f \in [0,1]^n$. Intuitively, the point $f^{[y]}$ is the point that has the same spatial relation to the point 0^n as f has to the codeword y (and vice-versa).

Definition 6. *A proper polytope $\mathcal{P}$ for the binary code $\mathcal{C}$ is $\mathcal{C}$*-**symmetric** *if, for all points f in the polytope $\mathcal{P}$ and codewords y in the code $\mathcal{C}$, the relative point $f^{[y]}$ is also contained in $\mathcal{P}$.*

3.2 All-Zeros Assumption. The validity of the all-zeros assumption is not immediately clear in the context of LP decoding. In this section, we prove that one *can* make the all-zeros assumption when analyzing LP decoders, as long as the polytope used in the decoder is $\mathcal{C}$-symmetric.

Theorem 7. *For any LP decoder using a $\mathcal{C}$-symmetric polytope to decode $\mathcal{C}$ under a binary-input memoryless symmetric channel, the probability that the LP decoder fails is independent of the codeword that is transmitted.*

Proof. For an arbitrary transmitted word y, we need to show that $\mathrm{WER}_y = \mathrm{WER}_{0^n}$. Define $\mathrm{BAD}(y) \subseteq \Sigma^n$ to be the set of received words $\tilde{y}$ that cause decoding failure, assuming y is transmitted:

$$\mathrm{BAD}(y) = \left\{ \tilde{y} \in \Sigma^n : \exists f \in \mathcal{P}, f \neq y, \text{ where } \sum_i \gamma_i f_i \leq \sum_i \gamma_i y_i \right\},$$

where the cost vector $\gamma = \gamma(\tilde{y})$ is a function of the received word $\tilde{y}$. (Note that this definition is conservative, in that it includes the case of multiple LP optima as decoding failure.) Rewriting equation (5), we have that for all codewords y,

$$\mathrm{WER}_y = \sum_{\tilde{y} \in \mathrm{BAD}(y)} \Pr[\ \tilde{y} \mid y\ ]. \tag{6}$$

As a particular case, for the codeword 0^n, we have $\mathrm{WER}_{0^n} = \sum_{\tilde{y} \in \mathrm{BAD}(0^n)} \Pr[\ \tilde{y} \mid 0^n\ ]$.

We now show that the space Σ^n of possible received vectors can be partitioned into pairs $(\tilde{y}, \tilde{y}^0)$ such that $\Pr[\ \tilde{y} \mid y\ ] = \Pr[\ \tilde{y}^0 \mid 0^n\ ]$, and $\tilde{y} \in \mathrm{BAD}(y)$ if and only if $\tilde{y}^0 \in \mathrm{BAD}(0^n)$. This partition, along with equation (6), gives $\mathrm{WER}_y = \mathrm{WER}_{0^n}$. The partition is performed according to the symmetry of the channel. Fix some received vector $\tilde{y}$. Define $\tilde{y}^0$ as follows: let $\tilde{y}^0_i = \tilde{y}_i$ if $y_i = \mathtt{0}$, and $\tilde{y}^0_i = \tilde{y}'_i$ if $y = \mathtt{1}$, where $\tilde{y}'_i$ is the symbol symmetric to $\tilde{y}_i$ in the channel. (See Section 1.1 for details on symmetry.) Note that this operation is its own inverse and therefore gives a valid partition of Σ^n into pairs.

First,we show that $\Pr[\ \tilde{y} \mid y\ ] = \Pr[\ \tilde{y}^0 \mid 0^n\ ]$. From the channel being memoryless, we have

$$\begin{aligned}
\Pr[\ \tilde{y} \mid y\ ] = \prod_{i=1}^{n} \Pr[\tilde{y}_i \mid y_i] &= \prod_{i:y_i=\mathtt{0}} \Pr[\tilde{y}^0_i \mid 0] \prod_{i:y_i=\mathtt{1}} \Pr[\tilde{y}_i \mid 1] && (7a)\\
&= \prod_{i:y_i=\mathtt{0}} \Pr[\tilde{y}^0_i \mid 0] \prod_{i:y_i=\mathtt{1}} \Pr[\tilde{y}'_i \mid 0] && (7b)\\
&= \prod_{i:y_i=\mathtt{0}} \Pr[\tilde{y}^0_i \mid 0] \prod_{i:y_i=\mathtt{1}} \Pr[\tilde{y}^0_i \mid 0] && (7c)\\
&= \Pr[\ \tilde{y}^0 \mid 0^n\ ]
\end{aligned}$$

Equations (7a) and (7c) follow from the definition of $\tilde{y}^0$, whereas equation (7b) follows from the symmetry of the channel (equations (1) and (2)). Now it remains to show that $\tilde{y} \in \mathrm{BAD}(y)$ if and only if $\tilde{y}^0 \in \mathrm{BAD}(0^n)$. Let γ be the cost vector when $\tilde{y}$ is received, and let γ^0 be the cost vector when $\tilde{y}^0$ is received, as defined in equation (3). Suppose $y_i = 0$. Then, $\tilde{y}_i = \tilde{y}^0_i$, and so $\gamma_i = \gamma^0_i$. Now suppose $y_i = 1$; then $\tilde{y}^0_i = \tilde{y}'_i$, and so

$$\gamma^0_i = \log\left(\frac{\Pr[\tilde{y}'_i \mid y_i = 0]}{\Pr[\tilde{y}'_i \mid y_i = 1]}\right) = \log\left(\frac{\Pr[\tilde{y}_i \mid y_i = 1]}{\Pr[\tilde{y}_i \mid y_i = 0]}\right) = -\gamma_i.$$

This follows from the symmetry of the channel (equations (1) and (2)). We conclude that

$$\gamma_i = \gamma^0_i \text{ if } y_i = 0, \text{ and } \gamma_i = -\gamma^0_i \text{ if } y_i = 1. \tag{8}$$

Fix some point $f \in \mathcal{P}$ and consider the relative point $f^{[y]}$. We claim that the difference in cost between f and y is the same as the difference in cost between $f^{[y]}$ and 0^n. In particular, we reason as follows:

$$\begin{aligned}
\sum_i \gamma_i f_i - \sum_i \gamma_i y_i &= \sum_{i:y_i=0} \gamma_i f^{[y]}_i - \sum_{i:y_i=1} \gamma_i f^{[y]}_i && (9a)\\
&= \sum_{i:y_i=0} \gamma^0_i f^{[y]}_i + \sum_{i:y_i=1} \gamma^0_i f^{[y]}_i && (9b)\\
&= \sum_i \gamma^0_i f^{[y]}_i - \sum_i \gamma^0_i 0_i. && (9c)
\end{aligned}$$

Equation (9a) follows from the definition of $f^{[y]}$, and equation (9b) follows from equation (8).

Now suppose $\tilde{y} \in \mathrm{BAD}(y)$, and so by the definition of BAD there is some $f \in \mathcal{P}$, where $f \neq y$, such that $\sum_i \gamma_i f_i - \sum_i \gamma_i y_i \leq 0$. By equation (9c), we have that $\sum_i \gamma_i^0 f_i^{[y]} - \sum_i \gamma_i^0 0_i^n \leq 0$. Because $\mathcal{P}$ is $\mathcal{C}$-symmetric, $f^{[y]} \in \mathcal{P}$, and by the fact that $f \neq y$, we have that $f^{[y]} \neq 0^n$. Therefore $\tilde{y}^0 \in \mathrm{BAD}(0^n)$. A symmetric argument shows that if $\tilde{y}^0 \in \mathrm{BAD}(0^n)$ then $\tilde{y} \in \mathrm{BAD}(y)$. □

Since the all-zeros codeword has zero cost, the all-zeros assumption gives the following corollary to Theorem 3:

Corollary 8. *Under the all-zeros assumption, for any binary linear code $\mathcal{C}$ over any binary-input memoryless symmetric channel, the LP decoder using the $\mathcal{C}$-symmetric polytope $\mathcal{P}$ will fail if and only if there is some non-zero point in $\mathcal{P}$ with cost less than or equal to zero.*

3.3 Fractional Distance of Symmetric Polytopes. The (classical) distance of a binary linear code is equal to the minimum weight of a non-zero codeword. This fact is very important when analyzing the distance of linear codes. It turns out that we can make a similar simplifying assumption when we analyze fractional distance. We refer the reader to [5] for a proof of the following:

Theorem 9. *The fractional distance of a $\mathcal{C}$-symmetric polytope $\mathcal{P}$ for a binary linear code $\mathcal{C}$ is equal to the minimum weight of a non-zero vertex of $\mathcal{P}$.*

In contrast to the classical distance, the fractional distance of a $\mathcal{C}$-symmetric polytope $\mathcal{P}$ for a binary linear code $\mathcal{C}$ can be computed efficiently. This can be used to bound the worst-case performance of LP decoding for a particular code and polytope. Since the fractional distance is a lower bound on the real distance, we thus have an efficient algorithm to give a non-trivial lower bound on the distance of a binary linear code. We refer the reader to [8, 5] for the details of this algorithm.

4 Conclusion

In this paper we outlined the basic technique of LP decoding. We derived general success conditions for an LP decoder, and showed that any decoder using a proper polytope has the ML certificate property. The fractional distance of a polytope was defined in this general setting, and it was shown that LP decoders correct a number of errors up to half the fractional distance. Furthermore, for binary linear codes, we established symmetry conditions for the polytope that allow for the all-zeros assumption, and regarding fractional distance as fractional weight. It is our hope that this paper will be a good starting point for the design and analysis of LP decoders.

There are many open questions in the area of LP decoding (for a full discussion, see [5]). Most importantly, we would like to see code constructions that take advantage of the simple characterization of pseudocodewords offered by the LP decoder. Analytic performance bounds using LP decoders have been proved for rate-1/2 repeat-accumulate codes [6, 4] and LDPC codes [8]. It is important to prove bounds for more complex turbo codes, as well as improve the bounds for LDPC codes (which are not yet close to the observed performance [5] of LP decoders in this case).

Given a specific proper polytope for an LP decoder, we can employ a number of different known techniques to tighten the polytope [18, 20, 16, 17], thereby obtaining an improved decoder (at the expense of additional computation). We discussed this idea somewhat in [5], but have yet to use it to strengthen our analytic performance bounds.

Finally, it would be interesting to look at codes over non-binary alphabets, and over non-memoryless channels. We could model a non-binary code in an LP by using several $0 - 1$ variables as indicators of a symbol taking on a particular value. Alternatively, we could map the code to a binary code, and use an LP relaxation for the binary code. It would be interesting to see if anything is gained by by representing the larger alphabet explicitly.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: turbo-codes. *Proc. IEEE International Conference on Communication (ICC), Geneva, Switzerland*, pages 1064–1070, May 1993.

[2] D. Bertsimas and J. Tsitsiklis. *Introduction to linear optimization*. Athena Scientific, Belmont, MA, 1997.

[3] C. Di, D. Proietti, T. Richardson, E. Telatar, and R. Urbanke. Finite length analysis of low-density parity check codes. *IEEE Transactions on Information Theory*, 48(6), 2002.

[4] G. Even and N. Halabi. Improved bounds on the word error probability of RA(2) codes with linear programming based decoding. In *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, October 2003.

[5] J. Feldman. *Decoding Error-Correcting Codes via Linear Programming*. PhD thesis, Massachusetts Institute of Technology, 2003.

[6] J. Feldman and D. R. Karger. Decoding turbo-like codes via linear programming. *Proc. 43rd annual IEEE Symposium on Foundations of Computer Science (FOCS)*, November 2002. To appear in *Journal of Computer and System Sciences*.

[7] J. Feldman, D. R. Karger, and M. J. Wainwright. Linear programming-based decoding of turbo-like codes and its relation to iterative approaches. In *Proc. 40th Annual Allerton Conference on Communication, Control, and Computing*, October 2002.

[8] J. Feldman, M. J. Wainwright, and D. R. Karger. Using linear programming to decode linear codes. *37th annual Conference on Information Sciences and Systems (CISS '03)*, March 2003. Submitted to *IEEE Transactions on Information Theory*, May, 2003.

[9] G. D. Forney, R. Koetter, F. R. Kschischang, and A. Reznik. On the effective weights of pseudocodewords for codes defined on graphs with cycles. In *Codes, systems and graphical models*, pages 101–112. Springer, 2001.

[10] B. Frey, R. Koetter, and A. Vardy. Signal-space characterization of iterative decoding. *IEEE Transactions on Information Theory*, 47(2):766–781, 2001.

[11] R. Gallager. Low-density parity-check codes. *IRE Trans. Inform. Theory*, IT-8:21–28, Jan. 1962.

[12] M. Grotschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[13] J. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms*, volume 1. Springer-Verlag, New York, 1993.

[14] D. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing, 1995.

[15] R. Koetter and P. O. Vontobel. Graph-covers and iterative decoding of finite length codes. In *Proc. 3rd International Symposium on Turbo Codes*, September 2003.

[16] L. B. Lasserre. An explicit exact SDP relaxation for nonlinear $0 - 1$ programs. *K. Aardal and A.M.H. Gerards, eds.,* Lecture Notes in Computer Science, 2081:293–303, 2001.

[17] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for $0 - 1$ programming. Technical Report PNA–R0108, Centrum voor Wiskunde en Informatica, CWI, Amsterdam, The Netherlands, 2001.

[18] L. Lovász and A. Schrijver. Cones of matrices and set-functions and $0 - 1$ optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[19] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley, 1987.

[20] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Optimization*, 3:411–430, 1990.

[21] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linkoping University, Sweden, 1996.