Iterative List Decoding

Justesen, Jørn; Høholdt, Tom; Hjaltason, Johan

Published in:
Proceedings of Information Theory Workshop on Coding and Complexity

Link to article, DOI:
10.1109/ITW.2005.1531863

Publication date:
2005

Document Version
Publisher's PDF, also known as Version of record

In the Proc. of IEEE ISOC ITW2005 on Coding and Complexity; editor M.J. Dinneen; co-chairs U. Speidel and D. Taylor; pages 90-93

Iterative List Decoding

Jørn Justesen
COM
Technical University of Denmark
DK-2800 Kgs Lyngby, Denmark
Email: jju@com.dtu.dk

Tom Høholdt
Department of Mathematics
Technical University of Denmark
DK-2800 Kgs Lyngby, Denmark
Email: T.Hoeholdt@mat.dtu.dk

Johann Hjaltason
Department of Mathematics
Technical University of Denmark
DK-2800 Kgs Lyngby, Denmark
Email: jh@emergens.dk

***Abstract*— We analyze the relation between iterative decoding and the properties of the extended parity check matrix. By considering a modified version of bit flipping, which produces a list of decoded words, we derive several relations between decodable error patterns and parameters of the code. By developing a tree of codewords at minimal distance form the received vector, we also obtain new information about the code.**

I. Introduction

We consider hard-decision iterative decoding of a binary (n, k, d) code. For a received vector, y, we calculate an extended syndrome $s = Hy'$, where H is a parity check matrix, but usually has more than $n-k$ rows. Our approach is based on one of the common versions of bit flipping (BF) [1], where the schedule is such that the syndrome is updated after each flip. In each step we flip a symbol chosen among those positions that reduce the weight of the extended syndrome, which we refer to briefly as the syndrome weight, u. A decoded word is reached when $u = 0$. In this paper we consider a variation of the common algorithm in the form of a tree-structured search. Whenever there is a choice between several bits, all possibilities are tried in succession. The result of the decoding algorithm is, in general, a list of codewords, obtained as leaves of the search tree. The bit flipping algorithm leads naturally to a solution in the form of a list of codewords at the same distance from y [2]. This list decoding concept is somewhat different from list decoding in the usual sense of all codewords within a certain distance from y. These concepts are useful both for designing decoding algorithms and for understanding the mechanisms of iterative decoding. By imposing a suitable set of limitation on the search we can still maintain a low complexity. Considering the decoding of the $(73, 45, 10)$ projective geometry code in detail, we find that a suitably modified BF algorithm can produce a maximum likelihood result or even a list of all codewords at minimum distance from the received vector.

II. Extended Parity Check Matrices and Syndromes

We consider a regular LDPC code given by a parity check matrix, possibly with more than $n - k$ rows, and the corresponding syndromes. All syndromes associated with single errors, i.e. the columns of the parity check matrix, have the same low weight, γ. For any pair of columns there is at most one row where both have a one. Thus an error pattern of weight w has syndrome weight u, where

$$w\gamma - w(w-1) \le u \le w\gamma \tag{1}$$

We assume that the rows of H satisfy a similar condition.

As the number of errors increases, so does the average weight of the syndrome, but the spread of possible values of u also increases, since it depends on the number of rows where multiple ones occur among the columns under consideration. It follows that the minimum distance of the code is lower bounded by

$$d \ge \gamma + 1 \tag{2}$$

A unique closest codeword is found by the BF algorithm if

$$w < (\gamma + 1)/2 \tag{3}$$

We refer to a coset as (u, w) when the weight of the coset leader is w, and the syndrome weight is u. These parameters serve as a first coarse characterization of the coset. A more detailed description relevant to the BF algorithm is based on the distribution of parity failures. Each position is checked by those rows of the parity check matrix that have a one in the corresponding column. We refer to the number of these rows that give a one in the syndrome, $f(j)$, as the number of parity failures for position j, and this is the information used for selecting which bit to flip. The vector $f(j)$ can be obtained by adding the rows corresponding to ones in the syndrome (using integer addition).

Lemma 1: If for a given syndrome and some position j, $f(j) > w$, any error pattern of weight w with that syndrome must include position j.

Proof: At most one of the rows that check a given position can have a one in any of the error positions. Since these are the only rows that can produce parity failures, a correct position has at most w parity failures.

A coset may be characterized by the parity failure distribution, f_i, which indicates the number of positions with i parity failures. Cosets with the same parity failure distribution may be expected to show similar properties when decoded by the BF algorithm.

A related set of parameters is the number of errors among the positions checked by row j in the parity check matrix. We let b_i indicate the number of rows that check i error positions.

0-7803-9481-X/05/$20.00

For a set of error positions, J, we have

$$\sum_{iodd} ib_i = \sum_{j \in J} f(j) \tag{4}$$

While f_i is a property of the coset and can be calculated from the received vector (or the syndrome), b_i is a property of a particular error pattern, and usually error patterns with different b_i exist in the same coset. However, it is clear that the syndrome weight u can be calculated as the sum of the b_i for odd i.

In each step, the basic BF algorithm changes the value of the received vector in a position that has $f(j) > \gamma/2$ making a transition to a coset with weight $w' = w - 1$ and $u' = u + \gamma - 2f(j)$. In this way we are able to characterize the possible transitions of the decoding algorithm. The vectors f_i and b_i are related to u and w by two sets of linear inequalities. However, we shall give the details only for the case of the $(73, 45)$ code, where the relations are equalities.

III. The List Decoding Algorithm

We first consider the basic version of the list decoding algorithm. Given the syndrome associated with a given coset, the vector of parity violations, $f(j)$, is computed from s. This state serves as the root of the decoding tree. Each node in the tree is characterized by the parent node, the extended syndrome, the syndrome weight u, and an ordered list of positions with $f(j) > \gamma/2$. From each node we can produce several offspring nodes by flipping a bit on the list and calculating the relevant characteristics. Eventually we reach a leaf, which is a decoded word when $u = 0$, or no solution if $u > 0$ and it is impossible to reduce the syndrome weight. The algorithm then back-tracks to the first node where a different bit can be flipped. It terminates when all nodes are explored.

The rationale for studying the list version of the BF algorithm is that we can describe the codewords that can be reached by the algorithm while avoiding an arbitrary selection of a coset leader.

Theorem 1: Let $F = \{J_i\}$ be a set of error patterns such that any subset of J_i is also in F. F is decoded by the list BF algorithm if and only if for every pattern, $J_i \in F$ there is at least one position such that $f(j) > \gamma/2$.

Proof: The condition is necessary, since for an error pattern not satisfying the condition the algorithm will either stop or flip a bit that is not in the error pattern. Sufficiency follows by induction on the weight of the error pattern: It is obvious for $w = 0$. Assume that error patterns in F of weight $w < w'$ are decoded. The condition ensures that in any error pattern of weight w' one bit will eventually be flipped, and the result is a pattern that is decoded.

The performance of the algorithm is analyzed by considering increasing values of u and w. For codes with a favorable distribution of syndrome weights, a proof of successful decoding can be based on the following result:

Lemma 2: For any error pattern of weight w in a coset with syndrome weight u, the list decoding algorithm will flip at least one error position if

$$u > w(\gamma - 1)/2$$

Proof: Since the syndrome is the sum of the w columns, one has at least u/w ones in rows where the syndrome is nonzero.

The goal of the iterative list decoding algorithm is to find the smallest distance to a codeword, and to find all codewords at this distance from the received word. The ideal situation is described in the following definition:

Definition 1: A Maximum-Likelihood List Decoding (MLLD) Algorithm takes an arbitrary received vector as input and produces as output a list of all error patterns in the same coset which have minimal weight within the coset.

For codes of moderate size, the lists can be rather short, and they could be useful in decoding an outer code in a concatenated scheme. If a codeword is found close to y, codewords at larger distances would have significantly lower conditional probability, and attempts to include them in the list could lead to long lists.

IV. Cosets

The analysis of the list decoding algorithm can provide information about the cosets of the code. For each (u, w) we would like to know the number of such cosets and the number of error patterns of weight w in each coset. It is useful to calculate the weight distribution of the space spanned by the columns of H. Such a calculation is feasible if $n - k$ is not too large or if H has a sufficiently regular structure.

Starting from low values of u we try to find the weights of the coset leaders. For a given u, we then consider error patterns of increasing weight w. For $w < d/2$ all error patterns are unique coset leaders, and we can find the number of (u, w) cosets directly. If not all cosets of weight less than $(\gamma + 1)/2$ are accounted for in this way, it follows from (1) that the weights of the remaining cosets must be relatively high, and they will not be decoded by bit flipping.

For each value of b_i we can count the corresponding error patterns, find $f(j)$ in the error positions, and decide the action of the BF algorithm. The (u, w) cosets can be classified in terms of their f_i vector, and this may allow us to determine the size of the list of solutions and thus the number of cosets. For a given weight, a large value of u indicates little overlap between the ones in the various columns. Thus the decision of the BF algorithm is the same as for majority decoding, and the algorithm just decodes the errors one at a time in an arbitrary order. A low syndrome weight occurs when a significant number of rows have two (or a larger even number of) ones. For error patterns of weight $w > d/2$ this will clearly be the case in some cosets where the weight of the coset leades is less than w. If w is the minimal weight in the coset, there may be multiple solutions in the form of codewords at the same distance, and we expect the number of solutions to increase with decreasing syndrome weight. Thus the analysis of BF provides at least partial information about the distribution of coset and syndrome weights.

V. Complexity

The complexity of the decoding algorithm can be measured by the number of nodes in the decoding tree. The basic algo-

rithm can lead to large search trees, and some modifications are introduced in order to limit the size. Clearly the complexity is at least of the order the list size multi-plied by the number of errors. However, two factors can make the number of nodes significantly larger: There may be branches that do not lead to error patterns, and the same error pattern may be found by flipping the bits in a different order.

A branch with no solutions occurs when a bit is flipped to go from a (u, w) coset to a $(u', w+1)$ coset where $u' < u$. In the example in Section VI we can prove that such nodes add little to the complexity. In order to control the second problem it may be useful to set a limit for w in the analysis. Thus a branch may be terminated if the syndrome weight indicates that no error pattern with w errors is possible. It is also clear that two error patterns in the same coset can share at most $w - d/2$ positions. Thus once a solution is found, a branch can be terminated if the bit flipped along the branch has too much overlap with solutions already found. For w significantly greater than $d/2$ we have to rely on properties of the code to ensure that the complexity remains polynomial.

Since the syndrome weight provides important information about the coset associated with a given node, it may be possible to create significant shortcuts by using known properties of the cosets. In the example we can simplify the decoding once the remaining number of errors is at most $d/2$. It is also possible to relax the requirement that u should decrease in every step to handle some special cosets.

VI. The (73,45,10) Projective Geometry Code

As an illustration of the algorithm discussed above, we consider the well-known difference set code [3].

A. Minimum weight codewords

The analysis is based on an extended 73 by 73 parity check matrix. It may be written in cyclic form, and it follows from the diffence set property that each row/column has weight $\gamma = 9$ and the product of any two rows is 1.

Fact [4]: A minimum weight codeword has ones in 10 positions such that the columns of the parity check matrix have two ones in 45 rows. The code has 32704 weight 10 codewords.

Proof: It follows from the difference set property that each of the 90 pairs must occur once, and the number of ones must be even. If any row has weight 4 or higher this is not possible. Using this property the codewords can be counted.

B. Error patterns and syndrome weights

From (1) we get a lower bound on u, but the upper bound can be tightened since the ones in two columns always share exactly one row. The lowest syndrome weights correspond to patterns that are part of minimum weight codewords. The highest syndrome weights occur when all columns share a single position. Between these extremes there are several cases.

The number of cosets with given weight of the extended syndrome can be counted. Since the all-ones vector is an extended syndrome, the distribution is symmetric. This distribution can be used to tighten the lower bound on u for certain values of w. For $w < 5$ the cosets are easily counted, since the coset leaders are unique. Enumerating the error patterns and the corresponding cosets is facilitated by noting that the parameters b_j must satisfy

$$\sum_j j b_j = 9w \quad (5)$$

$$\sum_{j\,odd} b_j = u \quad (6)$$

$$\sum_j j(j-1) b_j = w(w-1) \quad (7)$$

The parameters f_i satisfy a similar set of equations. Using these parameters, we have classified the cosets for increasing values of u and found the corresponding list sizes. In this way we have been able to conclude that with the exception of a single type of cosets, the condition of Lemma 2 is satisfied in all cases. Table I gives the number of cosets for some higher weights.

TABLE I
Cosets in the $(73, 45)$ Code

u	$w=5$	$w=6$	$w=7$	$w=8$
24		98112		
25	1373568		0	
28		6279168		0
29	8241408		2354688	
32		29360016		0
33	2649024		33652416	
36		24724224		24331776

C. List decoding

Even for this moderately complex code, it is possible to approach MLLD with an acceptable complexity. Even though the minimum distance is 10, which is excellent for a code with these parameters, most of the decodable error patterns have weight 7 and 8.

For $u = 24$ the coset leaders of weight 4 are unique, but there are some additional cosets, which have $w = 6$. These error patterns are the complete 6 arcs [4] in the projective plane, i.e. no three points are on a line, and any additional point will be on one of the lines that already has two points. In these cosets the list size is 21. Since an error position has only 4 parity failures, the error patterns cannot be corrected by the basic BF algorithm. However, we can extend the algorithm to allow the bits with 4 parity failures to be flipped the first time such a coset is reached in any branch of the decoding tree. Thus u is increased to 25 as we move to a $(25, 5)$ coset. The list decoding algorithm clearly needs to return to the same node to flip some of the other bits, but it is not allowed to loop between $(25, 5)$ and $(24, 6)$ cosets. The $(24, 6)$ cosets can also be reached from cosets with 7 or 8 errors, but there are no other cases where a bit with less than 5 parity failures has to be flipped. Other cosets with $w = 6$ have $u = 28, 32, 36, 40,$

or 48. For syndrome weight at least 36, the coset leader is unique. For $u = 28$, the list sizes are 7 and 9, for 32 there are 2 or 4 solutions. In $(29, 7)$ cosets the list size is 49. These cases with $w \leq 7$ accounts for all syndromes of weight up to 32.

REFERENCES

[1] Y. Kou, S. Lin and M. Fossorier: "Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results," *IEEE Trans.Inform.Theory* vol.47/7 pp. 2711-2736, Nov. 2001.

[2] J. Hjaltason: "List decoding of LDPC codes," MS Thesis, Technical University of Denmark, Jan, 2005.

[3] E.J. Weldon, Jr.: "Difference-set cyclic codes," *Bell Syst. Tech. J.*, vol. 45, pp. 1045-1055, Sept. 1966.

[4] J.W.P. Hirschfeld: *Projective Geometries over Finite Fields*, 2. ed., Oxford: Oxford Science Publications 1998.