Analysis of Iterated Hard Decision Decoding of Product Codes with Reed-Solomon Component Codes

Justesen, Jørn; Høholdt, Tom

Published in:
IEEE Information Theory Workshop, 2007. ITW '07.

Link to article, DOI:
10.1109/ITW.2007.4313069

Publication date:
2007

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

Analysis of Iterated Hard Decision Decoding of Product Codes with Reed-Solomon Component Codes

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

Abstract— **Products of Reed-Solomon codes are important in applications because they offer a combination of large blocks, low decoding complexity, and good performance. A recent result on random graphs can be used to show that with high probability a large number of errors can be corrected by iterating minimum distance decoding. We present an analysis related to density evolution which gives the exact asymptotic value of the decoding threshold and also provides a closed form approximation to the distribution of errors in each step of the decoding of finite length codes.**

I. Introduction

Product codes are important in applications that require low redundancy, large block length, and high data rate. In an example we consider a code that is similar to the one used for video on DVDs. Other product codes are being considered for transmission on optical cables. In spite of their low minimum distance, these codes are known to provide good performance by correcting a much larger number of errors. The performance can be improved significantly if the decoding of row and column codes is iterated, and a recent result in graph theory is an important tool for the analysis of such algorithms. For products of Reed-Solomon (RS) codes we obtain an explicit asymptotic bound for the number of errors that can be corrected.

In Section II codes and error patterns are described as bipartite graphs. We then characterize the error patterns that cause decoding failure as cores in the error graph. In Section III a theorem about random graphs is used to relate the existence of such a core to the total number of errors. The product codes are decoded by iterating between decoding rows and columns. The distribution of the remaining errors is described as truncated Poisson distributions in Section IV, and it is proved that this description is consistent with the random graph result. In Section V we consider improved rates related to using two different codes for rows and columns, and Section VI extends the analysis to graph codes with RS component codes.

II. Error Patterns in Product Codes as Random Graphs

We describe the product of two RS codes of length N as a complete bipartite graph. The right vertices represent row codes and the left vertices column codes. The code symbols are labels on the branches, and symbols on the branches that connect to a given vertex, taken in some specified order, have to satisfy the parity checks of the corresponding code.

The row codes correct all error patterns of weight at most T_1, the column codes correct T_2 errors. Initially we let $T_1 = T_2 = T$. A total of W errors are assumed to occur at randomly chosen positions. Since the decoding is independent of the codeword and the error values, it is sufficient to consider the error graph, a bipartite graph with $N + N$ vertices and W randomly chosen branches.

In the present analysis we make the simplifying assumption that the decoder of the component code corrects T or fewer errors, and in other cases the symbols are left unchanged. If T is not too small, the probability of decoding errors when more than T errors occur, approximately $1/T!$ [1], is insignificant. The analysis clearly also applies exactly to the case of erasure correction.

III. Cores in Random Graphs

If the decoding of the component codes is repeated until a stable result is obtained, decoding failure is described by the following concept:

Definition: A k-core in a graph is a subgraph with the property that all vertices have degree at least k.

Lemma 1: Iterative decoding of the product code fails if and only if the error graph contains a $T + 1$ core.

The well-known procedure for finding a core in a graph consists in successively removing any vertex of degree less than k and all branches connected to it. In terms of the product code, this procedure clearly amounts to decoding component codes and correcting all error patterns of weight at most T.

The existence of cores in random graphs has been a subject of considerable interest in graph theory. In particularly the following result due to Pittel et al. is important [2]: Let G

1-4244-1564-0/07/$25.00 ©2007 IEEE

be a random graph with n vertices and w edges. With high probability a k connected core exists when $w > c_k n/2$, but not for smaller w. The core includes a large fraction of the vertices. Here c_k is defined in terms of the Poisson distribution

$$\sigma(j) = e^{-\lambda}\lambda^j/j!$$
$$\pi_k(\lambda) = \sum_{j \geq k-1} \sigma(j)$$
$$c_k = min_\lambda[\lambda/\pi_k(\lambda)], \lambda > 0$$

Thus $c_3 = 3.35$, $c_4 = 5.14$, $c_5 = 6.80$, $c_6 = 8.37$, $c_9 = 12.78$. Asymptotically $c_k \approx \sqrt{k + k \log k}$.

The result applies without change to random bipartite graphs. This can be proved by making a small modification in the simplified proof of the basic result, which was given in [3]. Here the initial distribution of the degrees of the vertices is first considered, for finite n the sum of the degrees is $2w$, but asymptotically the distribution is Poisson. For the bipartite graph, the sum of the degrees is W on each side for finite N. The degree of a vertex is initially interpreted as the number of half-edges associated with the vertex. These half-edges are later combined in pairs to make the actual edges of the graph. The following algorithm simultaneously specifies the random graph and removes edges connected to vertices of low degree:

- Remove a half-edge from a light vertex (degree $< k$)
- Remove a randomly selected half-edge (which becomes the other part of the complete edge)
- Repeat the process as long as there are light vertices

The proof in [3] goes on from here to analyze the evolution of the degree distribution as a stochastic process. For the complete bipartite graph, the only modification is that the steps are:

- Remove a half-edge from a light vertex on the right
- Remove a randomly selected half-edge from the left (to complete the edge)
- Repeat these steps with right and left reversed
- Repeat as long as there are light vertices

Clearly the distribution of the half-edges on the two sides is the same, and each side follows the same stochastic process as for the original graph.

The result in [2] is asymptotic, i.e. k is fixed while the size of the graph increases. Thus the result for product codes is exact for codes of increasing length and fixed error-correcting capacity. This approach gives a better approximation to the performance of practical codes than the fixed-rate analysis in [4], since the codes of interest usually have high rate and moderate values of T, while N is relatively large. Iterating the decoding an unlimited number of times is also quite realistic, since a number of errors close to the limit can be corrected with a moderate number of decoding steps.

IV. Alternating Between Decoding Rows and Columns

In the analysis of random graphs one branch is removed in each step. In this section we give a description that follows the actual decoding of product codes by removing all light vertices on one side in each step. Initially the number of errors in each row follows a Poisson distribution since N is large compared to T. The average number of errors that are decoded in a row when all row codes are decoded can then be found from this distribution as

$$\sum_{j \leq T} j e^{-m} m^j/j!$$

We now introduce the simplifying assumption that these decoded positions are randomly distributed in the columns. Clearly this is not exactly the case for finite N, since T errors in a particular row must be located in different columns. However, the effect vanishes asymptotically, and as long as the total number of errors is large, the approximation is extremely close. Thus the first decoding of the column codes operates on a Poisson distribution with a reduced mean value. We shall prove that after each decoding step the degree distribution follows a truncated Poisson distribution

$$P[\delta = j] = bm^j/j!, j > T$$

and 0 otherwise, where b is chosen to make the terms sum to 1. The parameter m, which we refer to as the Poisson parameter, is clearly no longer the mean value.

Lemma 2: If the degrees of the vertices on one side of the graph follow a truncated Poisson distribution, a randomly chosen subset of the branches are removed, and all resulting light vertices are removed, the degree distribution of the remaining vertices is again a truncated Poisson distribution.

Proof: If the degrees had followed a full Poisson distribution, and a certain fraction, d of the branches were removed at random, the result would be a Poisson distribution with a reduced mean value. From each vertex with degree $T + j$, an average of $(T + j)d$ branches are removed, and the result does not depend on the distribution in other vertices. These contributions would then be added to give the new Poisson distribution. However, since vertices that have degree more than T after the decoding of the rows are obtained from columns which had more than T errors before the decoding, it is sufficient that this part of the distribution was Poisson.

Thus when a fraction of the remaining errors are corrected in the rows, the part of the column distribution above T becomes a truncated column distribution with a reduced mean value, and some columns with T or fewer errors are created. These columns are then decoded in the next step.

The calculation of the mean value of a truncated Poisson distribution is facilitated by the following identity, which is a standard result in traffic theory.

Lemma 3: $\sum_{j \geq T} j e^{-m} m^j/j! = m\pi_{T+1}(m)$

Proof: The result follows when the summation index is changed from j to $j - 1$.

This lemma explains why the summation in the definition of π starts at $k-1$ rather than k. We omit the subscript in the rest of this section.

We can now describe the evolution of the degree distribution:

Theorem 1: If the total number of errors in initially $W = MN$, the number of errors in each row follows a Poisson

distribution with mean M. After the first decoding, the number of errors per column follows a Poisson distribution with mean

$$m(1) = M\pi(M)$$

The degree distribution after each of the following stages of decoding follow a truncated Poisson distribution with parameters

$$m(j) = M\pi(m(j-1))$$

Proof: The expected number of errors after the first decoding is

$$N\sum_{j\geq T} je^{-M}M^j/j! = NM\pi(M)$$

using Lemma 3. We make the approximation that these errors are randomly distributed in the columns, and starting from these initial values we prove the result by induction. At stage j the degree distribution of rows (columns) is a truncated Poisson distribution with parameter $m(j)$ after all rows with at most T errors have been decoded. Thus using Lemma 3, the average number of errors in each row or column is $m(j)\pi(m(j))$. Going back to the columns (rows), the distribution of columns with more than T errors was unchanged in the last decoding of columns, but using Lemma 3, the decoding of the rows reduced the parameter of the distribution by a factor

$$m(j)\pi(m(j))/m(j-1)\pi(m(j-1)) = M\pi(m(j))/m(j-1)$$

by the induction hypothesis. Thus we find the new parameter to be

$$m(j+1) = M\pi(m(j))$$

completing the proof.

If the initial value, M, is less than $min\{m/\pi(m)\}$, $m(j)$ converges to zero, while for M less than this threshold, m converges to the largest value such that $m' = M\pi(m')$.We then have

Theorem 2: In the limit of large N, a product of RS codes of length N correcting a fixed number of errors, T, can be decoded by iterated decoding of the component codes to correct $W = NM$ errors, when

$$M < min_m\{m/\pi(m)\}$$

The results of this analysis coincide with the properties of random graphs found in [2].

The iteration can be illustrated graphically as a sequence of points on the line $m = x$ and the graph of $M\pi(x)$. The graphic also indicates how the expected number of iterations increases as the number of errors approaches the threshold.

Example 1: For $N = 256$ and $T = 8$, we get from Theorem 2 that approximately $W = Nc_9 = 3270$ errors can be corrected with high probability. Simulations of decoding with $T = 8$ indicate that the number of errors that can be reliably corrected is about 3100. The simulations also confirm that the truncated Poisson distribution is a good approximation the actual values for most of the decoding process. A single RS code with the same length and rate, using 16 bit symbols but shortened to $N = 2^{15}$, would correct 1984 errors.

V. Different Rates of the Row and Column Codes

For small values of T, experiments indicate that the best performance (highest rate for a given fraction of corrected errors), is obtained with different values of T, T_1 and T_2 on the right and left respectively. For small values of T_1+T_2, the difference between the optimal rates is small, but it increases with the value of the sum. Thus the choice of 8 and 5 errors in the DVD code is good also for iterated decoding.

The original proof of cores in random graphs is not easily modified to work with different values of T in subsets of the vertices. However, in our analysis such a change is easily made. If the definition of the function π is modified to alternate between T_1 and T_2, the parameters are still updated by

$$m(j+1) = M\pi(m(j))$$

The errors are corrected if the initial number of errors is below a certain threshold, but for larger values the decoding process reaches a stationary point with a pair of parameters, $(m', m")$.

Example 2: For $N = 256$, $T_1 = 8$, and $T_2 = 5$, the maximal number of errors is about 2725. With 2560 errors, typically 9-10 decoding stages are required. Table I gives the number of errors corrected in each stage. The left column gives the average as computed from the expression above. The other columns provide the results of 5 simulations. Clearly there is some variation, in particular towards the end of the decoding. However, even for this small example, the asymptotic average gives a useful indication of the expected properties. Moreover, if similar numbers are computed for a total number of errors 50 greater or smaller (about the standard deviation), the differences between the averages are significantly greater than between the simulated cases for the same total number. Thus we can get a quite accurate picture of the expected decoding by considering the asymptotic results within the typical range of total errors.

TABLE I

Simulations of decoding of product code with $T_1 = 8, T_2 = 5$

Aver.	Sim.1	Sim.2	Sim.3	Sim.4	Sim. 5
564	584	576	608	558	563
223	236	260	227	183	234
268	291	311	301	205	239
167	198	181	174	139	156
262	297	314	227	199	262
239	302	310	241	150	221
403	470	407	391	282	392
331	176	193	340	328	332
103	6	8	51	381	161
0	0	0	0	135	0

The iteration can be illustrated graphically (in the form well-known from EXIT graphs) as a staircase line between the graph of $\pi(x)$ for T_1 and a reflected version of the graph of $\pi(x)$ for T_2. The graphic indicates that the decoding threshold is reached when these two curves touch.

VI. Graph Codes

For a more general graph, consider the bipartite graph derived from a projective plane [5]. Here a given vertex on the right is connected to $q+1$ vertices on the left, and these

have edges connecting to the remaining $q(q+1)$ right vertices. The connections from the right and left sides have the same properties. The initial distribution of degrees is the same as in the original graph with the modification that very large degrees do not occur. This difference is negligible. In this case the steps of the random process are

- Remove a light half-edge from the right
- Remove a random half-edge from the subset of q+1 vertices on the left (to complete the edge)
- Remove a light edge chosen from the same subset of vertices on the left
- Remove a random edge from the relevant subset on the right.
- Continue as long as there are light vertices

Since the process is the same in the two sides, we consider only vertices on the right. Here all edges have the same probability of being removed in the last step, and thus the distribution evolves as in the original graph. In this way the performance depends on the good expansion properties of the graph as one would intuitively expect.

VII. Asymptotics and Capacity

Formally the results of this paper are derived by keeping T fixed and letting N increase. As mentioned earlier, we are mostly interested in finite cases where T is fairly small, and N is a moderately large number typical of applications.

However, it would not be unrealistic to consider much larger values of T and for such a fixed value to let N become very large. RS codes with 16 or 32 bit symbols are long enough to be analyzed in this way. To get a good performance for such a code, T_1 should be close to the expected number of errors in a row, and T_2 should be much smaller, providing a construction similar to Forney's original concatenated codes. Clearly the rate approaches 1 for N going to infinity, but the codes have close to optimal rates in the sense that long RS codes correcting a fixed number of errors approach the Hamming bound.

References

[1] R. McEliece and L. Swanson: "On the error probability for Reed-Solomon codes", *IEEE Trans. Inform. Theory*, vol. 32, pp. 701-703, September 1986.

[2] B. Pittel, J. Spencer, and N. Wormald: "Sudden emergence of a giant k-core in a random graph", *J.Comb.Theory, Series B*, vol.67, pp. 11-151, 1996.

[3] S. Janson and M.J. Luczak: "A simple solution to the k-core problem", *Random Structures Algorithms*, vol. 30, no. 1-2, pp. 50-62, 2007.

[4] M. Schwartz, P.H. Siegel, and A. Vardy: "On the asymptotic performance of iterative decoders for product codes", *Proceedings ISIT 2005*, pp. 1758-62, Adelaide, Australia, September 2005.

[5] T. Hoeholdt and J. Justesen: "Graph codes with Reed-Solomon component codes", *Proceedings ISIT 2006*, pp. 2022-26, Seattle, Washington, July, 2006.