MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Sparse Factor Graph Representations of Reed-Solomon and Related Codes

Jonathan S. Yedidia

TR2003-135 December 2003

Abstract

We present sparse factor graph representations of Reed-Solomon codes based on a fast Fourier transform. Similar "fast" transform factor graph representations are also presented for some extended Reed-Solomon codes. These representations can be used to create encoders, or message-passing decoders that use soft input information. We discuss various simplifications and transformations of the factor graphs that may be useful. Finally, we show that other interesting codes can be represented using sparse fast transform factor graphs.

1. First printing, December 2003. 2. Minor corrections, January 2004. Improved formatting, April 2004.

Sparse Factor Graph Representations of Reed-Solomon and Related Codes

Jonathan S. Yedidia

Abstract. We present sparse factor graph representations of Reed-Solomon codes based on a fast Fourier transform. Similar "fast" transform factor graph representations are also presented for some extended Reed-Solomon codes. These representations can be used to create encoders, or message-passing decoders that use soft input information. We discuss various simplifications and transformations of the factor graph representations that may be useful. Finally, we show that other interesting codes can be represented using sparse fast transform factor graphs.

1. Introduction

Sparse graphical representations of codes are of great current interest, because such representations can be used to construct efficient message-passing decoding algorithms. In Forney's paper that introduced "normal" factor graphs, which we hereafter call Forney factor graphs (FFG's), he also presented an interesting family of sparse FFG representations of Reed-Muller (RM) codes, based on a "fast" Hadamard transform [**1**]. In this paper, we present an analogous family of FFG representations for Reed-Solomon (RS) codes, based on the fast Fourier transform (FFT). We also discuss a number of potentially useful manipulations and generalizations of these representations.

Of course, RS codes have much better distance properties and are considerably more popular in practice than RM codes. Message-passing decoders based on the representations discussed here may be interesting competitors to the algebraic soft-decision decoders for RS codes proposed by Koetter and Vardy [**2**]. These message-passing decoders are designed to work on the q-ary level, as opposed to iterative decoders of RS codes that have very recently been proposed [**3, 4**] that take advantage of a bit-level representation.

2. Butterfly Factor Nodes

We begin by presenting a basic building block in our FFG representations, called a *butterfly* factor node. We refer the reader who is unfamiliar with FFG's to [**1**].

1991 *Mathematics Subject Classification.* 94B05.

Key words and phrases. Reed-Solomon Codes, Fast Fourier Transforms, Codes on Graphs.

In drawing a butterfly factor node, a careful distinction is made between "input" variables connected to its left side, and "output" variables connected to its right side. All variables in our FFG's are assumed to be q-ary variables, obeying the arithmetic laws of a Galois field $GF(q)$.

The n_I input variables connected to a butterfly factor node are denoted (starting with the variable at the top) $x_0, x_1, ..., x_{N_I-1}$, and the n_O output variables are similarly denoted $y_0, y_1, ..., y_{N_O-1}$. The input and output variables are related by a set of n_C linear constraints.

We will focus on FFG's that use butterfly factor nodes where $n_I = n_O = n_C = 2$, so that we have two constraints on two input and two output variables. In that case, the output variables are related to the input variables by equations of the form

$$\begin{aligned} y_0 &= Ax_0 + Bx_1 \\ y_1 &= Cx_0 + Dx_1. \end{aligned} \tag{2.1}$$

We draw (see figure 1) such a butterly factor node by placing the q-ary constants A, B, C, and D inside a square.

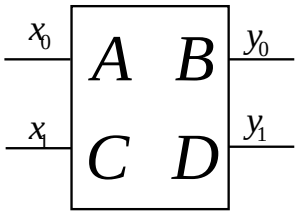

FIGURE 1. A butterfly factor node.

3. Reed-Solomon Codes and FFT's

We now review some properties of $[N, k, d]_q$ RS codes [**5**]. First, recall that $N = q - 1$, and $d = N - k + 1$. We let c_j be the codeword symbols, where j runs from 0 to $N - 1$; and let u_l be the information symbols, where l runs from 0 to $k-1$. We define α to be a primitive element of $GF(q)$ (i.e., the powers of α^n, where n runs from 1 to $q - 1$, are all different from each other). An RS code can then be defined by relating c_j to u_l according to

$$c_j = \sum_{l=0}^{k-1} u_l \alpha^{jl}. \tag{3.1}$$

This has the form of a discrete Fourier transform (DFT) over $GF(q)$, where the first k "frequency" components are given by the information symbols, and the other $N - k$ frequency components are fixed to zero [**5**].

Equation 3.1 can be re-written as a prescription for the generator matrix of a Reed-Solomon code. For example, an $[N = 8, k = 6, d = 3]_{q=9}$ Reed-Solomon code

would have a generator matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^1 & \alpha^4 & \alpha^7 & \alpha^2 & \alpha^5 \\ 1 & \alpha^4 & 1 & \alpha^4 & 1 & \alpha^4 & 1 & \alpha^4 \\ 1 & \alpha^5 & \alpha^2 & \alpha^7 & \alpha^4 & \alpha & \alpha^6 & \alpha^3 \end{pmatrix}. \tag{3.2}$$

where we have used the fact that $\alpha^8 = \alpha^0 = 1$ for $GF(9)$.

Because RS codewords can be obtained from a DFT, we can use an FFT construction for the DFT to represent RS codes, at least for convenient values of N and q. We will be particularly interested in FFT constructions when N, which is the number of components in the DFT, is a power of two. In such a case, the butterfly operations involved in the FFT take two inputs and outputs [**6**]. On the other hand, $N = 2^m$ for integer m implies that $q = 2^m + 1$, and recall that a Galois field only exists if q is a prime or a power of a prime. Thus, the most interesting candidate RS codes using this representation have $N = 4$, $q = 5$; $N = 8$, $q = 9$; $N = 16$, $q = 17$; or $N = 256$, $q = 257$.

Because our butterfly factor nodes can implement the butterfly operations in an FFT, we can use the well-developed theory of FFT's [**6**] to construct FFG representations of RS codes. Figure 2 shows an example.

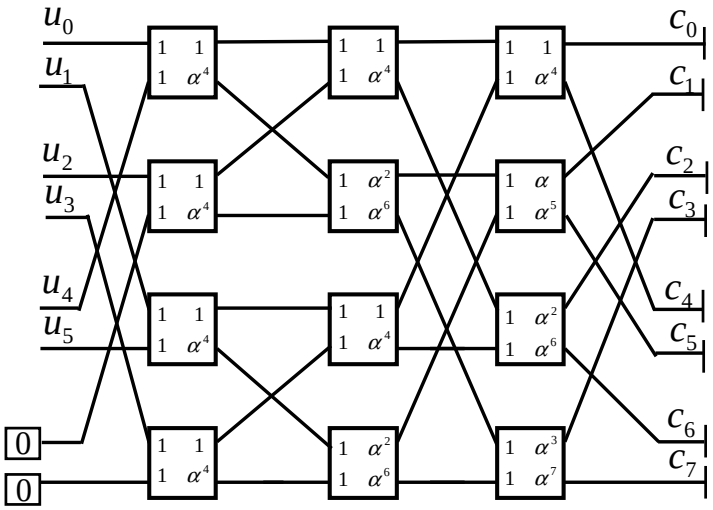

FIGURE 2. A representation of the $[N = 8, k = 6, d = 3]_{q=9}$ RS code with generator matrix given in equation(3.2). The six information symbols are denoted by u_i, and the eight code-word or transmitted symbols are denoted by c_i. That this graph represents the RS code as claimed can be verified by setting the information bits to values like 001000, and noting that one recovers the corresponding (third in this case) row of the generator matrix.

Althought the restrictions placed on q in these representations are somewhat inconvenient, it is still possible to imagine using them even on data that has an underlying binary format. For example, the RS code could be the outer code in a concatenated code construction, where the inner code was a non-linear binary code that had 17 or 257 codewords.

4. Extended RS Codes

It would clearly be worthwhile to have a similar representation for extended RS codes, such that $N = q = 2^m$, where m is an integer. The generator matrix for an extended RS code can be obtained from that of an RS code by adding a single column with a one in the first row, and a zero in all other rows. For example, the generator matrix for an $[N = 8, k = 4, d = 5]_{q=8}$ extended Reed-Solomon code is

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 0 & 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \\ 0 & 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha & \alpha^4 \end{pmatrix}. \tag{4.1}$$

Unfortunately, the codeword symbols of an extended RS code are no longer related by a DFT to the information symbols. Nevertheless, we expect that similar "fast" transform factor graph representations of extended RS codes can be constructed for general $N = q = 2^m$, although so far we have only been able to verify the correctness of such representations for small codes. For example, figure 3 shows such a representation for the code whose generator matrix was given above, although it is easier to see that this representation is correct by comparing with the equivalent generator matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 0 & 0 & \alpha^4 & \alpha & \alpha^4 & \alpha^2 & \alpha^2 & \alpha \\ 0 & 1 & \alpha^6 & \alpha^5 & \alpha & \alpha^3 & \alpha^4 & \alpha^2 \end{pmatrix}. \tag{4.2}$$

(Our Galois Field $GF(8)$ is constructed using the primitive polynomial $p(z) = z^3 + z + 1$, and the third row of the above matrix is obtained from the sum of the second and third rows of the previous one, while the fourth row is obtained from the sum of the second, third and fourth rows of the previous one.)

5. Simplified and Redundant Representations

In his representations of RM codes, Forney used equality and parity factor nodes. By using instead butterfly factor nodes that implement the Hadamard transform $y_0 = x_0 + x_1$, $y_1 = x_1$, we obtain a representation of RM codes that has an identical form (except for the different constants inside the butterfly factors) to our representation of RS codes.

Forney also presented a set of reduction rules that simplify such FFG representations. Similar rules apply to our representations as well. Using such rules, one can, for example, create the reduced representation of the $[N = 8, k = 4, d = 5]_{q=9}$ RS code shown in figure 4.

The potential advantages of *redundant* representations of codes for message-passing decoders were emphasized in [**7**]. It is easy to make redundant versions of these representations, by exploiting the symmetries of the codes, or the fact that alternative FFT constructions exist for a given DFT.

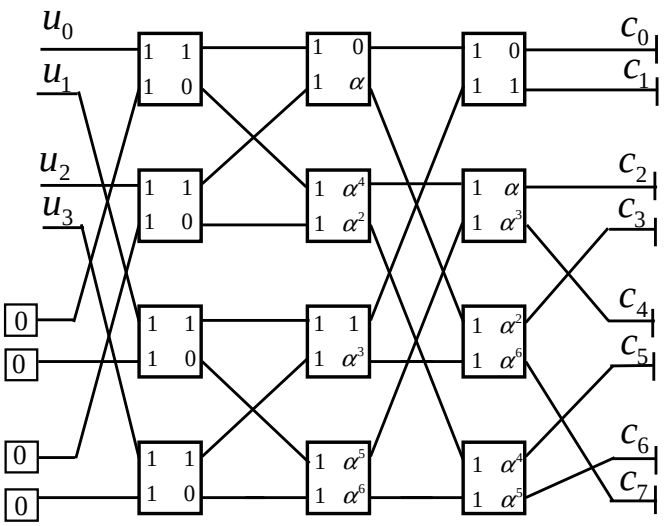

FIGURE 3. A representation of the $[N = 8, k = 4, d = 5]_{q=8}$ RS code with generator matrix given in equation(4.2).

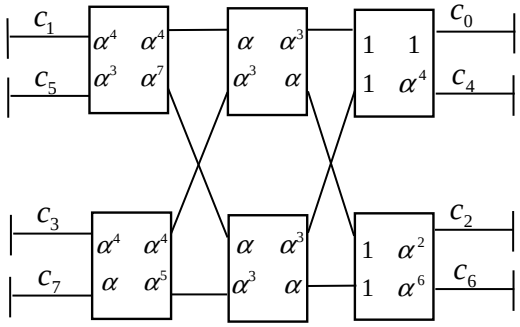

FIGURE 4. A reduced representation of an $[N = 8, k = 4, d = 5]_{q=9}$ RS code.

6. Other Codes

One can represent a wide variety of interesting codes using transforms constructed from butterfly factor nodes. As one small example, the $[N = 12, k = 6, d = 6]_{q=3}$ extended ternary Golay code can be represented as shown in figure 5. A single butterfly factor node in this representation can be identified with the $[N = 4, k = 2, d = 3]_{q=3}$ "tetra-code."

Carlach and Otmani [**8**] have shown that a variety of excellent self-dual binary codes, including the binary extended Golay code, can be constructed in a similar manner, using factor nodes that can be identified with the $[N = 8, k = 4, d = 4]_{q=2}$ extended Hamming code.

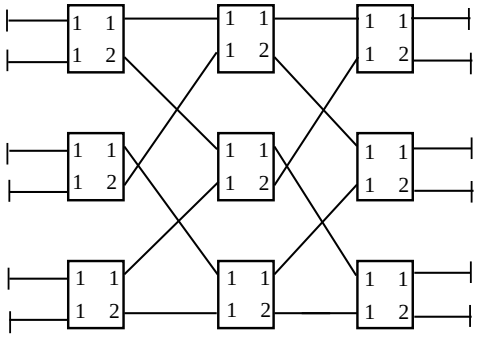

FIGURE 5. A representation of the extended ternary Golay code.

One potential problem with the representations considered above is that the internal variables do not receive any channel evidence. This may cause problems for message-passing decoders. It may be worthwhile to help the decoders, by using codes obtained from lengthening the above codes by connecting channel evidence to the internal variables. Such codes would still have reasonably good guaranteed minimum distance.

7. Encoders and Decoders

The representations discussed here can clearly be used to make efficient encoders, following Forney [**1**]. They can also be used as the basis of message-passing decoders. Such approaches are now quite standard, so we only make a couple comments.

The butterfly factor nodes that we have focused on enforce two constraints on four variables. It is obviously worthwhile to use a form of belief propagation that simultaneously enforces both constraints. For example, suppose that we have a butterfly factor node F that enforces the constraints $y_0 = Ax_0 + Bx_1$ and $y_1 = Cx_0 + Dx_1$, and we receive messages from the variables x_0, x_1, and y_0, denoted $n_F(x_0)$, $n_F(x_1)$, $n_F(y_0)$, and we want to compute the message $m_F(y_1)$ that F sends to y_1. We should use a message-update rule of the form

(7.1)
$$m_F(y_1) \propto \sum_{x_0} \sum_{x_1} \sum_{y_0} n_F(x_0) n_F(x_1) n_F(y_0) \delta(y_0 - Ax_0 - Bx_1) \delta(y_1 - Cx_0 - Dx_1)$$

if we want to use a "sum-product" algorithm. These message-update rules actually have a complexity that scales like q^2, because when x_0 and x_1 are given, then y_0 is determined.

One can consider such message-update rules a simple form of generalized belief propagation (GBP) [**9**]. In the context of an application that uses an ordinary real-valued FFT, Storkey discusses the advantages of GBP compared to a BP approach that treats the two constraints in each butterfly separately [**10**].

There are many interesting theoretical questions that this work raises, but from the practical point of view, decoding simulations are also needed to test its worth. We hope to report on the results of such simulations soon.

References

[1] G.D. Forney, Jr., "Codes on Graphs: Normal Realizations," *IEEE Trans. on Information Theory*, vol. 47, pp. 520-548, Feb. 2001.

[2] R. Koetter and A. Vardy, "Algebraic Soft-Decision Decoding of Reed-Solomon Codes," *IEEE Trans. on Information Theory*, vol. 49, pp. 2809-2825, Nov. 2003.

[3] J. Jiang and K.R. Narayanan, "Iterative Soft-Decision Decoding of Reed-Solomon Codes," to appear in *IEEE Communications Letters*, 2004.

[4] G. Ungerboeck, "Iterative Soft Decoding of Reed-Solomon Codes," Proceedings of the 3rd International Symposium on Turbo Codes and Related Codes, 2003.

[5] For background on RS codes, see, e.g., R. Blahut, *Algebraic Codes for Data Transmission*, Cambridge University Press, 2003.

[6] For background on FFT's, see, e.g., T. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*, MIT Press, 1990, Chapter 32.

[7] J.S. Yedidia, J. Chen, and M. Fossorier, "Generating Code Representations Suitable for Belief Propagation Decoding," *Proceedings of the 40th Annual Allerton Conference on Communications, Control, and Computing*, 2002.

[8] J.-C. Carlach and A. Otmani, "A Systematic Construction of Self-Dual Codes," *IEEE Trans. on Information Theory*, vol. 49, pp. 3005-3009, Nov. 2003.

[9] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms," MERL Technical Report TR2002-35, 2002, available online at http://www.merl.com/papers/TR2002-35/.

[10] A. Storkey, "Generalised Propagation for Fast Fourier Transforms with Partial or Missing Data," to appear in Advances in Neural Information Processing Systems, vol. 16, 2004.

Mitsubishi Electric Research Labs (MERL), 201 Broadway, 8th Floor, Cambridge, MA 02139

E-mail address: yedidia@merl.com