v. 3, n. 12, 2023

Journal of
Engineering
Research

INDEX CODING FROM REED-SOLOMON CODES

Valéria G. P. Alencar
FEEC/UNICAMP, Campinas, SP

Max H. M. Costa
FEEC/UNICAMP, Campinas, SP

The original version of this paper (in Portuguese) was presented and published in the Proceedings of the XL Brazilian National Congress of Applied and Computational Mathematics (CNMAC), in September/2021.

Abstract: The problem of index coding subject to transmission errors was initially considered by Dau et al. [5]. In this work we establish a connection between index coding and error correcting codes, through the tree construction for nested cyclic codes proposed in [3]. We implemented the tree construction algorithm in Matlab language, which helped to solve some implementation problems found in [3]. We verified that for cyclic codes there will not always be an increase in the error correction capability between the levels of the tree. This is why we restricted this study, initially, to Reed-Solomon codes, since they are MDS codes, which guarantees an increase in Hamming distance at each level of the tree. This means that, under certain conditions, the knowledge of side information will be interpreted as an increase in the error correction capability of the decoder.

Keywords. Index Coding, Side Information, Error Correcting Codes, Finite Fields.

INTRODUCTION

The classic noise-free index coding problem consists of a sender with k independent messages $w_1,...,w_k$ and a broadcast channel with multiple receivers, where each receiver demands a subset of messages, while knowing the values of a different subset of messages as side information. Let $R_1,...,R_n$ be n receivers and suppose that S_i represents lateral information and D_i the demand of receiver R_i, where S_i, D_i $\subset \{1,...,k\}$. The goal is to find a coding scheme, called index coding, that satisfies the demand of all receivers and uses a minimum number of transmissions.

We considered the specific case of index coding for noisy discrete broadcast channels, where all receivers demand all messages from the source, i.e., $S_i \cup D_i = \{1,...,k\}$. Given this model, the possibility arises of designing error correcting codes whose mapping of messages into codewords is such that the decoder can increase the Hamming distance in a receiver that has prior knowledge of the values of some subset of the messages as side information.

We are assuming that the sender is unaware of the subset of messages already known to the receiver and performs encoding so that any possible side information can be efficiently used in the decoder. The notion of multiple interpretation was introduced in [9], showing that the greater the amount of lateral information available in the receiver, the greater the error correction capacity in decoding. Constructed codes must also be error correcting codes for index coding when the receiver has no side information, i.e., when D_i=ø.

The index coding technique presented here is given by the tree construction of nested cyclic codes proposed in [3]. We restrict ourselves to the Reed-Solomon codes because they are MDS (maximum separation distance) codes, which guarantees an increase in the Hamming distance at each level of the tree. This means that, under certain conditions, knowledge of lateral information will be interpreted as an increase in the error correction capability of the decoder.

PRELIMINARIES

INDEX CODING WITH SIDE INFORMATION

The goal of the index coding is to perform a joint encoding of the messages of all users, in order to simultaneously meet the demands of all receivers, while transmitting the resulting message at the highest possible rate. Please see [2] for an in-depth look at index coding.

Below we present the model through an example. Consider the wireless communication system shown in Figure 1. Receiver R_i is requesting the message w_i, $i \in \{1,2,3\}$, and knows other messages as side information; In particular, receiver 1 knows w_3 as side information, receiver 2 knows w_1 and w_3, and receiver 3 knows w_1 and w_2. The server wants to send messages to receivers using as few transmissions as possible.

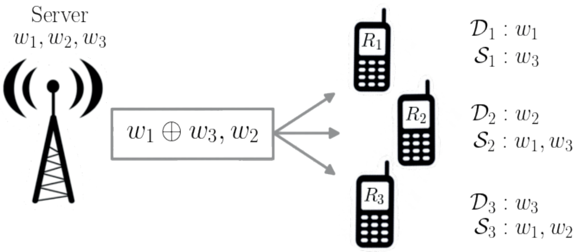

Figure 1: Index coding with three receivers.

Assuming a noiseless broadcast channel, the server would communicate all messages by sending one at a time, in three transmissions. Alternatively, when transmitting the two coded messages: $w_1 \oplus w_3$ and w_2, each receiver can retrieve their demands message using the received coded messages and available side information as seen below:

$$\begin{aligned}
&\text{Receiver 1: } (w1 \oplus w3) \oplus w3 = w1\\
&\text{Receiver 2: } (w1 \oplus w3) \oplus w2 \oplus (w1 \oplus w3) = w2\\
&\text{Receiver 3: } (w1 \oplus w3) \oplus w2 \oplus (w1 \oplus w2) = w3
\end{aligned}$$

A TREE CONSTRUCTION WITH NESTED CYCLIC CODES

The tree-based algebraic construction of nested cyclic codes, proposed by Barbosa and Costa [3], aims to:

i) Encode, independently, different data packets, providing protection against channel errors;

ii) Encode different data packets producing codewords that are added resulting in the packet C_0;

iii) Correct the errors on C_0 and, finally, recover the data in the receiver by polynomial operations.

NESTED CYCLIC CODES

A nested code is characterized by a global code where each element is given by a sum of codewords, each belonging to a different subcode. That is,

$$c = i_1 G_1 \oplus i_2 G_2 \oplus \cdots \oplus i_N G_N$$

where ⊕ represents an XOR operation. For an information vector i_ℓ,$1 \leq \ell \leq N$, the codeword $i_\ell G_\ell$ belongs to a subcode C_ℓ of code C and c ∈ C.

$$\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2 + \cdots + \mathcal{C}_N$$

Nested cyclic codes, whose subcodes are generated by generator polynomials, were originally proposed by Heegard [6], were originally called under the term of partitioned linear block codes and can be defined as follows:

Let $C=\{C(x) \in F_q[x]; g(x)|C(x)\}$ be a t-error correcting cyclic code having g(x) as the generator polynomial. Note that $C=\langle g(x)\rangle$ is an ideal of the ring $R_n=\mathbb{F}_q[x]/(x^n-1)$, but is also a vector subspace of $\mathbb{F}_q^n$, so we can write:

$$C(x) = p_1(x)g_1(x) + p_2(x)g_2(x) + \cdots + p_N(x)g_N(x)$$

where $C_\ell(x)=p_\ell(x)g_\ell(x), 1 \leq \ell \leq N$, is an encoded packet belonging to the t_ℓ-error correcting subcode

$$\mathcal{C}_\ell = \{C_\ell(x) \in F_q[x]; g_\ell(x)|C_\ell(x)\}$$

Generated by $g_\ell(x)$ and satisfying the conditions:

1) $g_\ell(x)|g_{\ell+1}(x)$;
2) $deg[C_\ell(x)] < deg[g_{\ell+1}(x)]$.

THE TREE CONSTRUCTION METHOD

Consider a tree in which root node is associated with the vector subspace of an encompassing error correcting code. Set the root node of the tree to be the code C such that:

$$\mathcal{C}_{i0} = \langle g_{i0}(x)\rangle = \{C_{i0}(x) \in F_q[x]; g_{i0}(x)|C_{i0}(x)\}.$$

This subspace corresponds to a t_0-cyclic error correcting code $C_{i0}(n, k_{i0})$, generated by the polynomial $g_{i0}(x)$.

A tree of nested cyclic codes is a finite tree such that satisfies the conditions:

1)Each inner node (including the root node) can be subdivided into another inner node and a terminal node;

2)The jth inner node is associated with a linear subspace $C_{ij} \subset F_q^n$ of dimension k_{ij} and can be subdivided into the subspaces as follows:

$$\mathcal{C}_{ij} = \mathcal{C}_{i(j+1)} + \mathcal{C}_{t(j+1)}$$

such that $\mathcal{C}_{i(j+1)} \cap \mathcal{C}_{t(j+1)} = \{0\}$ and $k_{ij} = k_{i(j+1)} + k_{t(j+1)}$ where $k_{i(j+1)} = dim\mathcal{C}_{i(j+1)}$ and $k_{t(j+1)} = dim\mathcal{C}_{t(j+1)}$.

3)All subspaces associated with the inner nodes must be cyclic linear block codes defined by a generator polynomial;

4)If $\mathcal{C}_{ij} = \langle g_{ij}(x) \rangle$ e $\mathcal{C}_{i(j+1)} = \langle g_{i(j+1)}(x) \rangle$. Então $g_{ij}(x) | g_{i(j+1)}(x)$. Furthermore, $g_{ij}(x) | x^n - 1$ to any $g_{ij}(x)$;

5)To conclude, the last inner node will have no ramifications.

If $p_j(x)$ the data packet associated with the terminal node, for 1≤j≤T. The encoding is given by:

$$C_j(x) = p_j(x) g_{i(j-1)}(x).$$

Then, the encoded packets are summed up and the resulting codeword is sent out by the transmitter

$$C_0(x) = C_1(x) + C_2(x) + \cdots + C_T(x).$$

TREE CONSTRUCTION: ALGORITHM AND CONSIDERATIONS

We describe a few algorithms in Matlab and considerations for fitting to the model of Tree Construction, which can be found at [1], allowing to perform the calculations on finite fields by making the appropriate transformations from integer representation to powers of α, based on Table 1. Below, we exemplify the main idea of the algorithm.

Exemple 1. *For T=3 is C_{i0}(7,5) be a Reed-Solomon code in* GF(8) *and $k_{t1}=k_{t2}=2$ the dimensions of subspaces $C_{t1}=C_{t2}$, respectively. The last node associated with C_{i2} of dimension $k_{i2}=1$. The packages $p_1(x)=x+\alpha^2$, $p_2(x)=\alpha^3x+\alpha$ are associated with terminal nodes and both have length 2, the package $p_3(x)=\alpha^5$* is associated with the last node and has length 1.

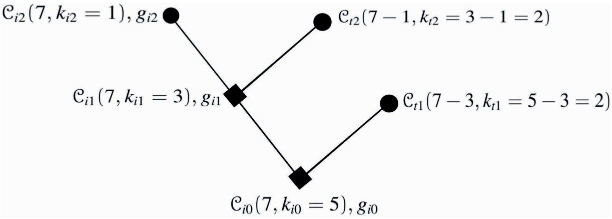
$\mathcal{C}_{i2}(7, k_{i2} = 1), g_{i2}$
$\mathcal{C}_{t2}(7-1, k_{t2} = 3-1 = 2)$
$\mathcal{C}_{i1}(7, k_{i1} = 3), g_{i1}$
$\mathcal{C}_{t1}(7-3, k_{t1} = 5-3 = 2)$
$\mathcal{C}_{i0}(7, k_{i0} = 5), g_{i0}$

Figure 2: Tree Construction.

Let α be the primitive element of GF(8), then the generator polynomials are:

1) $deg(g_{i0}(x)) = n - k_{i0} = 2 \Rightarrow g_{i0}(x) = \prod_{j=1}^{2}(x - \alpha^j) = x^2 + \alpha^4 x + \alpha^3.$

2) $deg(g_{i1}(x)) = n - k_{i1} = 4 \Rightarrow g_{i1}(x) = \prod_{j=1}^{4}(x - \alpha^j) = x^4 + \alpha^3 x^3 + x^2 + \alpha x + \alpha^3.$

3) $deg(g_{i1}(x)) = n - k_{i1} = 4 \Rightarrow g_{i1}(x) = \prod_{j=1}^{4}(x - \alpha^j) = x^4 + \alpha^3 x^3 + x^2 + \alpha x + \alpha^3.$

Consider the packages $p_1(x) = x + \alpha^2$, $p_2(x) = \alpha^3 x + \alpha$, $p_3(x) = \alpha^5$, i.e, $p_1 = [1,4]$, $p_2 = [3,2]$ and $p_3 = [7]$, according to Table 1. Coding the packets, we have:

$$\begin{aligned} C_1(x) &= p_1(x) g_{i0}(x) \\ &= x^3 + \alpha x^2 + \alpha^4 x + \alpha^5; \\ C_2(x) &= p_2(x) g_{i1}(x) \\ &= \alpha^3 x^5 + \alpha^5 x^4 + \alpha^6 x^3 + \alpha^2 x^2 + x + \alpha^4; \\ C_3(x) &= p_3(x) g_{i2}(x) \\ &= \alpha^5 x^6 + \alpha^5 x^5 + \alpha^5 x^4 + \alpha^5 x^3 + \alpha^5 x^2 + \\ & \alpha^5 x + \alpha^5. \end{aligned}$$

Then, the codeword to be transmitted is given by:

$$
\begin{aligned}
C_0(x) &= C_1(x) + C_2(x) + C_3(x) \\
&= \alpha^5 x^6 + \alpha^2 x^5 + 0x^4 + \alpha^3 x^3 + 1x^2 + 0x + \alpha^4 .
\end{aligned}
$$

Power of α	GF (8) Element	Binary	Integer
0	0	000	0
1	1	001	1
α	x	010	2
α^2	x^2	100	4
α^3	x+1	011	3
α^4	x^2+x	110	6
α^5	x^2+x-1	111	7
α^6	x^2+1	101	5

Table 1: Construction of a Galoi field: GF(8)

Considering tree construction based on Reed-Solomon codes and assuming that the receiver has side information available, when will there be an increase in error correction capability?

Proposition 1. *Due to the nesting structure, the variable error correctability characteristic can only be observed if there is a sequential removal of the packets associated with the nodes from the root to the top of the tree.*

Demonstration. Suposing that $C_\ell(x), 1 \leq \ell \leq T$, is the first coded packet known at the receiver, then

$$
\begin{aligned}
&C_0(x) = p_1(x)g_{i0}(x) + \cdots + p_{(\ell-1)}(x)g_{i(\ell-2)}(x) + \\
&p_{(\ell+1)}(x)g_{i\ell}(x) + \cdots + p_T(x)g_{i(T-1)}(x) \\
&= [p_1(x) + \cdots + p_{(\ell-1)}(x)q_{(\ell-1)}(x) + p_{(\ell+1)}(x)q_{(\ell+1)}(x) \\
&+ \cdots + p_T(x)q_T(x)]g_{i0}(x)
\end{aligned}
$$

therefore, $C_0(x) \in \mathcal{C}_{i0}(n, k_{io})$, whose error correction capability is t_0. Note that even though the receiver knows about other packages $C_j(x), \ell < j \leq T$, the result does not change. On the other hand, if all packages $C_j(x); 1 \leq j < \ell$, are known to the receiver, we can write:

$$
\begin{aligned}
C_0(x) &= p_{(\ell+1)}(x)g_{i\ell}(x) + \cdots + p_T(x)g_{i(T-1)}(x) \\
&= [p_{(\ell+1)}(x)\overline{q}_{(\ell+1)}(x) + \cdots + p_T(x)\overline{q}_T(x)]g_{i\ell}(x)
\end{aligned}
$$

thus, $C_0(x) \in \mathcal{C}_{i\ell}(n, k_{i\ell})$, whose error correction capability is is $t_\ell \geq t_0$, equality occurs only when $d_{min}(C_\ell) - d_{min}(C_0) < 2$.

We analyze two cases of tree construction of nested cyclic codes, with the same parameters at each level. In one of them we observe no increase in the error correction capability from the second to last internal node of the tree. This is due to the variety of possibilities of generating polynomials for a cyclic code of parameters (n, k). As a result, we demonstrate in Proposition 2, that for Reed-Solomon codes this feature of increasing capacity will be guaranteed provided that:

$$k_{ij} - k_{i(j+1)} \geq 2, \forall j = 0, \ldots, T-1.$$

Exemple 2. *Let $C_{i0}(15, 10)$ be a cyclic code in GF(2) and $k_{t1}=4$, $k_{t2}=2$ be the dimensions of the subspaces C_{t1}, C_{t2} respectively. The last node is associated with C_{i2} with dimension $k_{i2}=4$.*

We consider the factorization:

$$x^{15} - 1 = (1 + x)(1 + x^3 + x^4)(1 + x + x^2 + x^3 + x^4)$$
$$(1 + x + x^2)(1 + x + x^4)$$

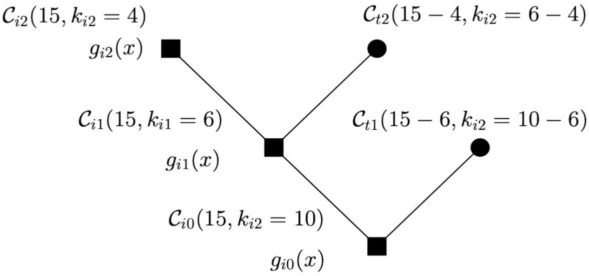
$\mathcal{C}_{i2}(15, k_{i2} = 4)$
$g_{i2}(x)$
$\mathcal{C}_{t2}(15 - 4, k_{i2} = 6 - 4)$
$\mathcal{C}_{i1}(15, k_{i1} = 6)$
$g_{i1}(x)$
$\mathcal{C}_{t1}(15 - 6, k_{i2} = 10 - 6)$
$\mathcal{C}_{i0}(15, k_{i2} = 10)$
$g_{i0}(x)$

Figura 3: Tree construction.

Case 1. Consider the generator polynomials:

- $g_{i0}(x) = (1+x)(1+x^3+x^4)$

$\Rightarrow d_{min}(C_{i0}) = 4 \text{ and } t_0 = 1;$

- $g_{i1}(x) = g_{i0}(x)(1+x+x^2+x^3+x^4)$

$\Rightarrow d_{min}(C_{i1}) = 6 \text{ and } t_1 = 2;$

- $g_{i2}(x) = g_{i1}(x)(1+x+x^2)$

$\Rightarrow d_{min}(C_{i2}) = 8 \text{ and } t_2 = 3.$

Note that there was an increase in the error correction capacity at each level of the tree, which does not occur in the following case.

Case 2. Now consider the following generator polynomials:

- $g_{i0}(x) = (1+x)(1+x+x^4)$

$\Rightarrow d_{min}(C_{i0}) = 4 \text{ and } t_0 = 1;$

- $g_{i1}(x) = g_{i0}(x)(1+x^3+x^4)$

$\Rightarrow d_{min}(C_{i1}) = 6 \text{ and } t_1 = 2;$

- $g_{i2}(x) = g_{i1}(x)(1+x+x^2)$

$\Rightarrow d_{min}(C_{i2}) = 6 \text{ and } t_2 = 2;$

Proposition 2. *Given a (n, k) Reed-Solomon code, which has minimum distance d=n-k+1, it is possible to guarantee an increase in error correction capability at each level of the tree as long as* $k_{ij} - k_{i(j+1)} \geq 2, \forall j = 0, \ldots, T-1.$

Demonstration: We must prove that $t_{i(j+1)} \geq t_{ij} + 1, \forall j = 0, \ldots, T-1.$ Without loss of generality, set j=0. If $k_{i0} - k_{i1} \geq 2$, then, we can write:

$$(-d_{i0} + n + 1) + d_{i1} - n - 1 \geq 2$$

$$d_{i1} - 1 \geq d_{i0} - 1 + 2$$

$$\left\lfloor \frac{d_{i1}-1}{2} \right\rfloor \geq \left\lfloor \frac{d_{i0}-1}{2} \right\rfloor + 1$$

$$t_{i1} \geq t_{i0} + 1$$

CONCLUSIONS

This work considers index coding from the construction of nested cyclic codes. After the error correction phase the jth packet pj(x) is decoded by the operations:

$$p_j(x) = \begin{cases} \dfrac{[C_0(x) \mod g_{ij}(x)]}{g_{i(j-1)}(x)}, & \text{if } 1 \leq j \leq T \\ \dfrac{C_0(x)}{g_{i(T-1)}(x)}, & \text{if } j = T \end{cases}$$

The information will be contained in the remainder of the division of $C_0(x)$ by $g_{ij}(x)$, since the modulo operation eliminates the influence of all messages related to polynomials of degree equal to or greater than the degree of $g_{ij}(x)$. Thus, the quotient of the final division operation provides the desired information, since all other messages have degree less than the degree of the divisor polynomial. Therefore, in the case of the last package, only the division operation is required. In summary, the modulo operation removes the branches above the node of interest and the division operation removes the branches below. Therefore, no side information is needed at the receiver in order to recover the data packets.

The verification that for cyclic codes there will not always be an increase in the error-correction capacity between the levels of the tree, as considered in [3], leads us to search for answers on how to correctly choose the generating polynomials for a parameter code (n, k) and its subcodes, in order to guarantee subcodes with larger Hamming distance, to the point of observing an increase in the error-correction capacity between the levels of the tree. A method for construction of chains of some linear block codes while maintaining the minimum distances (of the generated subcodes) as large as possible is presented in [8] and may be the answer to this quest.

REFERENCES

1. Alencar, V. G. P, Construcão-de-Arvore em MatLab no GitHub (2021). https://github.com/valeriaurca/Construção-de-Arvore.git.

2. Arbabjolfaei, F. and Kim, Y. H. (2018), Fundamentals of Index Coding, *Foundations and Trends@ in Communications and Information Theory*, 14:163–346, 2018. DOI: 10.1561/0100000094.

3. Barbosa, F. C. and Costa, M. H. M. A tree construction method of nested cyclic codes, *IEEE Information Theory Workshop*, 302-305, 2011. DOI: 10.1109/ITW.2011.6089441.

4. Blahut, R.E. *Algebraic Codes for Data Transmission.* Cambridge University Press, New York, 2003.

5. Dau, S. H., Skachek, V. and Chee, Y. M. Error Correction for Index Coding with Side Information, *IEEE Transactions on Information Theory,* 59:1517–1531, 2013. DOI: 10.1109/TIT.2012.2227674.

6. Heegard, C. Partitioned linear block codes for computer memory with "stuck-at" defects, *IEEE Transactions on Information Theory,* 29:831–842, 1983. DOI:10.1109/TIT.1983.1056763.

7. Hefez, A., Villela, M. L. T. *Códigos Corretores de Erros, 2a. edição.* IMPA, Rio de Janeiro, 2017.

8. Vinck, A. J. H., Luo, Y. Optimum distance profiles of linear block codes, *IEEE International Symposium on Information Theory*, 1958–1962, 2008. DOI: 10.1109/ISIT.2008.4595331.

9. Xiao, L., T. Fuja, J. Kliewer and D. J. Jr. Costello. Nested codes with multiple interpretations, *2006 40th Annual Conference on Information Sciences and Systems*, 851-856, 2006. DOI: 10.1109/CISS.2006.286586.