Complete Erasure Recovery of Irregular Repeat-Accumulate Codes

Saejoon KIM

Department of Computer Science
Sogang University
Seoul 121-742, Korea
Email: saejoon@sogang.ac.kr

Abstract

We consider the problem of complete erasure recovery of irregular repeat-accumulate codes used over the binary erasure channel. Borrowing a technique from [1, 3], we give an extended construction of irregular repeat-accumulate codes that complete the recovery of all erasures and as a result, achieve the channel capacity.

1. INTRODUCTION

Irregular repeat-accumulate (IRA) codes are systematic codes defined by bipartite graphs of degree sequence functions $\rho(x) = \sum_i \rho_i x^{i-1}$ and $\lambda(x) = \sum_i \lambda_i x^{i-1}$ where $\rho_i(\lambda_i)$ is the fraction of edges whose left (right) node degree is i. They are linear-time encodable, have decoding algorithms that are also linear-time, and moreover, almost achieve the channel capacity on the binary erasure channel. Specifically, if an IRA code of k information bits is used over the binary erasure channel with erasure probability δ, an erasure recovery algorithm can recover all but at most ηk, $\eta > 0$, information bits with exponentially high probability for all code rates less than the channel capacity $C = 1 - \delta$ [4]. For the code rate of $C(1-\epsilon)$, $\epsilon > 0$, only $O(k \log \frac{1}{\epsilon})$ time is required by the erasure recovery algorithm to recover this many bits [9]. Hence if we ignore the ηk information bits, IRA codes achieve the capacity of binary erasure channel with only logarithmic sacrifice in decoding complexity. So ideally, we would like to finish recovery of the remaining ηk information bits.

Similar problem arises in low-density parity-check codes as well. Specifically, for a family of degree sequence functions satisfying some constraint [8], an erasure recovery algorithm for low-density parity-check codes can recover all but at most some small number of bits with exponentially high probability. Luby *et al.* [5, 6] have shown that for low-density parity-check codes, complete erasure recovery is made possible by restricting the graph associated with the low-density parity-check code to be an expander which will ensure the erasure recovery algorithm to continue until all bits are recovered. Unfortunately, the same argument cannot be directly applied to IRA codes due to their systematic nature.

To this end, we give an extended construction of IRA codes such that all bits are recovered with exponentially high probability. This extended construction uses a pre-code that will recover all bits from almost all bits recovered by the IRA code. Attractive feature of this pre-code, whose underlying technique was originally presented in [1, 3], is that it is of rate near-1, and linear-time encodable and decodable enabling the extended construction to maintain the desirable properties of IRA codes but recover slightly more losses. In other words, the extended construction is linear-time encodable and decodable, and capacity-achieving over the binary erasure channel. The main result of this article is the following theorem.

Theorem 1 *For the binary erasure channel of parameter δ, there exist linear-time encodable and decodable codes of rate $(1-\delta)(1-\epsilon)$ that can recover, with high probability, all δ-fraction of erasures, $\epsilon > 0$. More precisely, the encoding complexity per bit is $\Theta(-1/((\log \epsilon)\epsilon)) + O(\log \frac{1}{\epsilon})$ and the decoding complexity per bit is $O(\log \frac{1}{\epsilon}) + \Theta(-1/((\log \epsilon)\epsilon^2))$.*

Using a pre-code to clean up the remaining losses or errors is not a new idea and has been presented before, for example, in [1, 3, 10]. Furthermore, the use of a pre-code is not limited to IRA codes and can be applied to *any* code to clean up the losses or errors. In particular, Raptor codes [10] use irregular low-density parity-check codes [5, 6] as pre-code to finish the recovery of remaining erasures in LT codes [7]. While the same code can work equally well as pre-code in our code construction to finish the recovery of remaining losses in IRA codes, our approach presents yet another

scheme that achieves the same goal.

The outline of this article is as follows. In the next section, we give a high-level structure of our code construction and in the following section, we show that our construction yields a family of capacity-achieving codes and prove Theorem 1. We conclude with a summary at the end.

2. CODE CONSTRUCTION

The main additional ingredient in our extended construction of IRA code is the rate near-1 pre-code. The extended construction consists of the pre-code cascaded with IRA code of degree sequence functions $\rho(x)$ and $\lambda(x)$ as shown in Figure 1. While the pre-code is constructed explicitly, the IRA code is chosen randomly in the ensemble defined by the degree sequence functions.

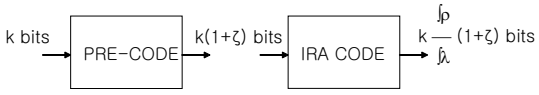

Figure 1: Extended Construction of Irregular Repeat-Accumulate Code

For encoding, use the pre-code to encode k information bits to produce $k(1+\zeta)$ bits and next, use the IRA code to encode the $k(1+\zeta)$ bits from the pre-code to produce $k\frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}(1+\zeta)$ bits, $\zeta > 0$. We shall call the encoded bits from the pre-code, intermediate bits. Codeword of the overall code is the codeword of the IRA code, *i.e.*, intermediate bits concatenated with check bits from IRA code. As a result, rate of the code is $((1+\frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx})(1+\zeta))^{-1}$. Note that only ζ-fraction loss is incurred in the code rate compared to IRA code alone. For the IRA code, we shall use the rate $C(1-\epsilon')$ code in [9] that can recover almost all losses with decoding complexity $O(\log\frac{1}{\epsilon'})$ per bit for channel capacity C. In terms of the gap from channel capacity ϵ, rate of the overall code is $C(1-\epsilon)$ where $\epsilon = \frac{\epsilon'+\zeta}{1+\zeta}$ for $\epsilon > 0, \epsilon' > 0, \zeta > 0$.

Decoding is clear. First, use the IRA code to decode the received bits and erasures, which will recover almost all erasures. Next, use the pre-code to finish off recovery of all remaining erasures. If the pre-code has code rate near-1, *i.e.*, ζ is very small, then the extended construction can recover all erasures at rates close to the capacity.

We will now describe the pre-code which will comprise of two codes $\mathcal{C}_1$ and $\mathcal{C}_2$. As the pre-code is defined over the field $\mathbb{F}_q$ for $q \gg 2$, an explanation is in order. Before k information bits are encoded by the pre-code, they are partitioned into $k' = \frac{k}{l}$ blocks each of length l and each block is represented by a symbol in the field $\mathbb{F}_q$ for $q = 2^l$ where l is some positive integer. Now the k information symbols are ready for encoding by the pre-code. Code $\mathcal{C}_1$ is a systematic code that encodes k' information symbols and outputs $k'\zeta'$ check symbols. $\mathcal{C}_1$ has the property that given all of the check symbols, up to γ fraction of the erasures in information symbols can be recovered for some γ to be determined later. Code $\mathcal{C}_2$ encodes the $k'\zeta'$ check symbols from $\mathcal{C}_1$ to a string of symbols of length $k'\zeta$, $\zeta = 4\zeta'$, and it has the property that up to $\gamma k'$ erasures can be recovered. For $\mathcal{C}_2$, we may use the linear-time encodable and decodable code by Spielman [11] that can recover a small but positive constant fraction of erasures. Hence the pre-code encodes k' information symbols to $k'(1+\zeta)$ symbols comprising of k' information symbols and $k'\zeta$ check symbols, and can recover up to $\gamma k'$ erasures. Finally, the encoded symbols are stretched to $k(1+\zeta)$ bits before IRA code encodes them.

Decoding of the pre-code can be done in reverse direction. $k(1+\zeta)$ intermediate bits and erasures obtained from decoding IRA code are partitioned into $k'(1+\zeta) = \frac{k(1+\zeta)}{l}$ blocks each of length l and each block is represented by a symbol in the field $\mathbb{F}_q$ or an erasure if the block contains an erasure. Code $\mathcal{C}_2$ decodes to recover all erasures in its $k'\zeta$ symbols and code $\mathcal{C}_1$ decodes to recover all of its k' information symbols given all $k'\zeta$ check symbols. Finally k' symbols are stretched back to k bits. It suffices to describe the code $\mathcal{C}_1$ in the remaining of this article.

Consider a $\triangle$-regular Ramanujan bipartite graph $G = (V, E)$ where $V = V_1 \cup V_2$ and $|V_1| = |V_2| = n$. We shall assume that its second largest eigenvalue in absolute value is λ and an ordering on E from 1 to $k' = n\triangle$. Let $E(v)$ be the set of edges incident on the vertex v, and for a string $\mathbf{x} = (x_e)_{e\in E}$ of length k' whose entries are indexed by the ordering on E, let $(\mathbf{x})_v$ be the part of $\mathbf{x}$ that is indexed by $E(v)$.

Encoding of $\mathcal{C}_1$ works as follows. For k' information symbols $\mathbf{x}$, associate them with the edges in the graph G and use a systematic Reed-Solomon code of dimension $\triangle$ and length $(1+\frac{\zeta'}{2})\triangle$ to encode each $(\mathbf{x})_v$ for $v \in V_1 \cup V_2$. This gives a lower bound on the size of the field $\mathbb{F}_q$, that is $2^l \geq (1+\frac{\zeta'}{2})\triangle$. As will be shown in the next section, we will require $\triangle = \Theta(\frac{1}{\zeta^2})$ which along with the lower bound on field size deduces to a lower bound on l in terms of the gap from channel capacity ϵ, *viz.*,

$$l \geq \Theta(\log\frac{1}{\epsilon}). \tag{1}$$

Finally, $\frac{\zeta'}{2}\triangle$ check symbols for each $(\mathbf{x})_v$, $v \in V_1 \cup V_2$,

will be associated with the vertex v, and there is a total of $n\zeta'\triangle$ check symbols as required.

Lemma 2 *The pre-code is linear-time encodable. More precisely, the encoding time is* $\Theta(\frac{-k}{(\log\epsilon)\epsilon})$.

Proof: Since $\triangle$ is a constant, it takes a constant amount of time to compute $\frac{\zeta'}{2}\triangle$ check bits for each $(\mathbf{x})_v$, $v \in V_1 \cup V_2$, and hence a linear time to encode $\mathcal{C}_1$. Specifically, it takes time proportional to $\triangle \cdot \frac{\zeta'}{2}\triangle$ to encode for each $(\mathbf{x})_v$, $v \in V_1 \cup V_2$, and hence $O(\frac{k'}{\zeta})$ time overall. Since $\mathcal{C}_2$ can be encoded in $O(k')$ time, the total encoding time of pre-code is $O(\frac{k'}{\zeta})$. This together with the inequality of (1) proves the Lemma. □

In the next section, we will show that code $\mathcal{C}_1$ can recover up to $\gamma k'$ erasures in the information symbols for $\gamma = O(\zeta^2)$. As a result, the pre-code will be able to recover up to any $O(\zeta^2)$ fraction of erasures which should be at least η, the fraction of intermediate bits left unrecovered by the IRA code, in order for complete erasure recovery.

3. COMPLETE ERASURE RECOVERY

Let Dec denote erasure recovery algorithm for the systematic Reed-Solomon code of dimension $\triangle$ and length $(1 + \frac{\zeta'}{2})\triangle$ that outputs the recovered symbols corresponding to the systematic part of the codeword for erasures less than $\lfloor\frac{\zeta'}{2}\triangle\rfloor$. For higher number of erasures, assume Dec just outputs the received symbols and erasures of the systematic part, *i.e.*, no erasure recovery. Denote the systematic part of the received symbols and erasures by $\mathbf{y}$, and the received check symbols by $\mathbf{z}$ which were completely recovered by $\mathcal{C}_2$. Furthermore denote the received check symbols associated with vertex v by $(\mathbf{z})_v$. Next shows the *Erasure Recovery Algorithm* that will be applied to code $\mathcal{C}_1$, with m to be determined later.

Erasure Recovery Algorithm

For $i = 1$ to m do {
  Let W stand for V_1 if i odd, and for V_2 otherwise.
  Iteration i: for every $v \in W$, let $(\mathbf{y})_v = \mathsf{Dec}((\mathbf{y})_v \circ (\mathbf{z})_v)$
    where $\circ$ denotes concatenation.
}

Now let $d_0 = \lfloor\frac{\zeta'}{2}\triangle\rfloor$ and δ_0 denote the minimum distance and the relative minimum distance of the Reed-Solomon code, respectively.

Theorem 3 *For* $0 < \alpha < 1$ *and* $d_0 \geq \frac{3}{2}\lambda$, *Erasure Recovery Algorithm can recover* $O(\zeta^2)$*-fraction of erasures in* $\mathcal{C}_1$ *in linear time.*

Proof: Let $Y^{(i)}$ be the set of erasures in the information symbols identified by the edges of the graph G after iteration i of the *Erasure Recovery Algorithm.* Suppose

$$|Y^{(0)}| \leq \alpha\delta_0(\delta_0 - \frac{\lambda}{\triangle})k'(1+\zeta')$$

where $|Y^{(0)}|$ is the number of erasures in the received information symbols before *Erasure Recovery Algorithm* begins. Let $W = V_1$ for i odd and $W = V_2$ for i even, and

$$A^{(i)} = \{v \in W : E(v) \cap Y^{(i)} \neq \emptyset\}$$

for $i = 1, 2, \cdots$. So $A^{(i)}$ is the set of vertices whose symbols associated with the incident edges were not completely recovered after iteration i of the *Erasure Recovery Algorithm.* Then for $v \in A^{(1)}$, $|E(v) \cap Y^{(0)}| \geq d_0$, and for $v \in A^{(2)}$, $|E(v) \cap Y^{(1)}| \geq d_0$. The assumption on $|Y^{(0)}|$ along with the fact that $|Y^{(0)}| \geq |A^{(1)}|d_0$ gives

$$|A^{(1)}| \leq \alpha(\delta_0 - \frac{\lambda}{\triangle})n(1+\frac{\zeta'}{2}). \tag{2}$$

To get a bound on $|A^{(2)}|$, we use Lemma 4 in [12] and obtain

$$2|A^{(2)}|d_0 \leq \lambda(|A^{(1)}| + |A^{(2)}|) + 2|A^{(1)}||A^{(2)}|\frac{\triangle}{n}$$

which along with Equation (1) translates to

$$|A^{(2)}| \leq \frac{\lambda}{d_0(1-\alpha) + \lambda(2\alpha - 1)}|A^{(1)}|.$$

Hence if $d_0 \geq \frac{3}{2}\lambda$, $|A^{(2)}| \leq \beta|A^{(1)}|$ with $\beta < 1$. As in [12], this argument easily extends to all $A^{(i)}$'s, $i = 1, 2, \cdots$. Specifically, $|A^{(i+1)}| \leq \beta|A^{(i)}|$, until $A^{(i)} = \emptyset$ when which the decoding is complete. Therefore, after

$$\log_{\frac{1}{\beta}}|A^{(1)}| + 2$$

iterations, erasure recovery is complete, and this logarithmic depth decoding can be implemented to run in linear time using the argument in [2]. More precisely, since it takes $O(\triangle^2)$ time to recover each erasure and there are at most $k'\gamma = O(k'\zeta^2)$ erasures, the total decoding time is $O(\frac{k'}{\zeta^2})$ which reformulates to $\Theta(\frac{-k}{(\log\epsilon)\epsilon^2})$ using the inequality of (1). Since $\delta_0 \approx \frac{\zeta'}{2}$,

$O(\zeta^2)$-fraction of erasures can be recovered, and moreover since Ramanujan graphs have the property of $\lambda \approx 2\sqrt{\triangle}$, we require $\triangle = \Theta(\frac{1}{\zeta^2})$. □

Linear-time decodability of $\mathcal{C}_1$ together with that of $\mathcal{C}_2$ gives the next Lemma. ($\mathcal{C}_2$ has $O(k')$ decoding time.)

Lemma 4 *The pre-code is linear-time decodable. More precisely, the decoding time is* $\Theta(\frac{-k}{(\log \epsilon)\epsilon^2})$.

Since IRA code of rate $(1-\delta)(1-\epsilon')$ can correct any δ fraction but a small constant number of erasures, the described extended IRA code of rate $(1-\delta)(1-\epsilon')/(1+\zeta)$ can correct any δ fraction of erasures for sufficiently large block length which completes the proof of Theorem 1 in Section I. The encoding complexity per bit of the extended construction is that of the pre-code which is $\Theta(-1/((\log \epsilon)\epsilon))$ plus that of the IRA code which is $O(\log \frac{1}{\epsilon})$. Similarly, the decoding complexity per bit is that of the IRA code which is $O(\log \frac{1}{\epsilon})$ plus that of the pre-code which is $\Theta(-1/((\log \epsilon)\epsilon^2))$.

4. CONCLUSION

In this article, an extended construction of IRA codes using a pre-code has been presented. We have shown that the inclusion of pre-code enables IRA codes to finish recovery of all erasures at code rates very close to the channel capacity. An important feature of this construction is that it only requires linear amount of additional encoding and decoding times to the IRA codes in [4, 9].

References

[1] N. Alon and M. Luby, "A Linear Time Erasure-Resilient Code with Nearly Optimal Recovery," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1732-1736, Nov. 1996.

[2] A. Barg and G. Zémor, "Error exponents of expander codes," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1725-1729, June 2002.

[3] V. Guruswami and P. Indyk, "Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets," *Proc. Symp. on Theory of Computing '02*, Canada, 2002.

[4] H. Jin, A. Khandekar and R. McEliece, "Irregular Repeat-Accumulate Codes," *Proc. 2nd. International Conf. Turbo Codes*, Brest, France, pp. 1-8, Sept. 2000.

[5] M. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman and V. Stemann, "Practical Loss-Resilient Codes," *Proc. 29th Annual ACM Symp. on Theory of Computing*, pp. 150-159, 1997.

[6] M. Luby, M. Mitzenmacher, M.A. Shokrollahi and D.A. Spielman, "Efficient Erasure Correcting Codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 569-584, Feb. 2001.

[7] M. Luby, "LT-codes," *Proc. ACM Symposium on Foundations of Computer Science*, 2002.

[8] P. Oswald and M.A. Shokrollahi, "Capacity-Achieving Sequences for the Erasure Channel," *IEEE Trans. Inform. Theory*, vol. 48, no. 12 pp. 3017-3028, Dec. 2002.

[9] I. Sason and R. Urbanke, "Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, pp. 1247-1256, June 2004.

[10] M.A. Shokrollahi, "Raptor Codes," *IEEE Int'l Symp. on Inform. Theory '04*, Chicago, USA, 2004.

[11] D.A. Spielman, "Linear-Time Encodable and Decodable Error-Correcting Codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1723-1731, Nov. 1996.

[12] G. Zémor, "On Expander Codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 835-837, Feb. 2001.