Local Recovery Properties of Capacity Achieving Codes

Arya Mazumdar†, Venkat Chandar* and Gregory W. Wornell§

***Abstract*—A code is called *locally recoverable* or *repairable* if any symbol of a codeword can be recovered by reading only a small (constant) number of other symbols. The notion of local recoverability is important in the area of distributed storage where a most frequent error-event is a single storage node failure. A common objective is to repair the node by downloading data from as few other storage node as possible.**

In this paper we study the basic error-correcting properties of a locally recoverable code. We provide tight upper and lower bound on the local-recoverability of a code that achieves capacity of a symmetric channel. In particular it is shown that, if the code-rate is ϵ less than the capacity then for the optimal codes, the maximum number of codeword symbols required to recover one lost symbol must scale as $\log \frac{1}{\epsilon}$.

I. Introduction

An *update-efficient* code is a mapping from messages to codewords such that for small a perturbation in a message the corresponding codeword changes only slightly. The term *update-efficiency* quantify this property. In the definitions below we use the following terminology. The *support* of a vector $\boldsymbol{x}$ (written as $\mathrm{supp}(\boldsymbol{x})$) is the set of coordinates where $\boldsymbol{x}$ has nonzero values. By *weight* of a vector we mean the size of support of the vector. It is denoted as $\mathrm{wt}(\cdot)$. The logarithms of this paper have base 2 unless otherwise mentioned.

Definition 1: A *code* $\mathcal{C} \in \mathbb{F}_2^n$ is a collection of binary n-vectors with a one-to-one encoding map $\phi : \mathbb{F}_2^k \to \mathcal{C}, k < n$. The *update-efficiency* of a code $(\mathcal{C}, \phi)$ is the maximum number of bits that needs to be changed in a codeword when 1 bit in the message is changed. A code has update-efficiency t if for all $\boldsymbol{x} \in \mathbb{F}_2^k$, and for all $\boldsymbol{e} \in \mathbb{F}_2^k : \mathrm{wt}(\boldsymbol{e}) = 1$, we have $\phi(\boldsymbol{x} + \boldsymbol{e}) = \phi(\boldsymbol{x}) + \boldsymbol{e}'$, for some $\boldsymbol{e}' \in \mathbb{F}_2^n : \mathrm{wt}(\boldsymbol{e}') \leq t$.

In our previous work [9], it was shown that the update-efficiency has to scale logarithmically with the block-length of the code if we are to to achieve any nontrivial rate with vanishing probability of error over binary symmetric as well as binary erasure channels. It was also shown that, there exists capacity-achieving codes with this scaling.

An informal *dual* property of the update-efficiency in codes is the *local recoverability*. Let us define this property for binary codes. However, this definition, as well as all other results of this paper can be easily generalized for non-binary codes.

† Department of ECE, University of Minnesota, Twin Cities, Minneapolis, MN 55455, email: arya@umn.edu.

* MIT Lincoln Lab, Lexington, MA 02421, email: vchandar@mit.edu.

§ Department of EECS, Massachusetts Institute of Technology, Cambridge, MA 02139, email: gww@mit.edu.

This work was supported in part by the US Air Force Office of Scientific Research under Grant No. FA9550-11-1-0183, and by the National Science Foundation under Grant No. CCF-1017772.

Definition 2: A code $\mathcal{C} \subset \mathbb{F}_2^n$ has *local recoverability* r, if for any $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{C}$ and for any $1 \leq i \leq n$, there exists a function $f_i : \mathbb{F}_2^r \to \mathbb{F}_2$ and indices $1 \leq i_1, \ldots, i_r \leq n, i_j \neq i, 1 \leq j \leq r$, such that $x_i = f_i(x_{i_1}, \ldots, x_{i_r})$.

It is evident that any codeword symbol of $\mathcal{C}$ can be recovered from at most r other symbols of the codewords. This property is desirable in distributed storage systems and was introduced in that context in [7].

In [7], as well as in [11], locally recoverable codes that also correct a number of adversarial errors, were considered. A trade-off between the local recoverability and error-correction was presented. In particular it was shown that, for a q-ary linear code, $q > 2$,

$$d \leq n - k - \left\lceil \frac{k}{r} \right\rceil + 2\,,$$

where d is the minimum distance, k is the dimension, and r is the local recoverability of the code. This can be generalized to nonlinear codes with all possible alphabet sizes. Indeed, it is shown in [3] that, for any q-ary code with size M, local recoverability r and minimum distance d,

$$\log M \leq \min_{1 \leq t \leq \left\lceil \frac{n}{r+1} \right\rceil} \Big[tr + \log A_q(n - t(r+1), d) \Big], \quad (1)$$

where $A_q(n, d)$ is the maximum size of a q-ary code with distance d.

However, so far we have not seen any work that considers capacity results for locally recoverable codes. But analogous results were presented for update-efficient codes in [2], [9]. In this paper, we fill that gap. Although, our results are derived for binary-input channels, as opposed to the large alphabet channel models usually considered for distributed storage, our proofs extend easily for large alphabet case.

The two main channels that we consider are the *binary symmetric channel* with error probability p, BSC(p), and the *binary erasure channel* with erasure probability p, BEC(p). Capacity of BSC(p) is $1 - \mathrm{h}(p)$, where $\mathrm{h}(p) = -p \log p - (1-p)\log(1-p)$ is the binary entropy function and capacity of BEC(p) is $1 - p$.

We show that it is possible to construct codes with rate ϵ less than the capacity of BEC (or BSC) that has local recoverability $O(\log \frac{1}{\epsilon})$ and simultaneously update-efficiency scaling logarithmically with block-length. Our main result is to show a converse result that the scaling $O(\log \frac{1}{\epsilon})$ for local recoverability of an ϵ-away-from-capacity code is optimal.

II. Main results

A. Existence of good codes

It is relatively easy to construct a good code with update efficiency $O(\log n)$, local recoverability $O(\log \frac{1}{\epsilon})$, and rate $C-\epsilon$, where C the capacity of the BSC or BEC. This construction is a little modification of the construction for update-efficient codes that appears in [9].

A *low density parity check* (LDPC) code is a linear code such that each row of the parity check matrix has a small (constant) number of nonzero values. It is known that LDPC codes achieve a positive error-exponent. That is for every $\epsilon > 0$ and any sufficiently large n, there exist an LDPC code of length n and rate $1-\mathrm{h}(p)-\epsilon$ that has check degree (number of 1s in a row of the parity-check matrix) at most $O(\log \frac{1}{\epsilon})$, and probability of incorrect decoding at most $2^{-E_L(p,\epsilon)n}$, for some $E_L(p,\epsilon) > 0$. We refer the reader to [6], [8] for more details of this result. Suppose we call this code $\hat{\mathcal{C}}$. Let $\hat{G}$ be the generator matrix of $\hat{\mathcal{C}}$.

Let $m = \frac{1+\alpha}{E_L(p,\epsilon)} \log n$, an integer, $\epsilon, \alpha > 0$. We avoid using ceiling and floor to have a clean presentation, unless it is not obvious from the context. Let G be the $nR \times n$ matrix that is the Kronecker product of $\hat{G}$ and the $n/m \times n/m$ identity matrix $I_{n/m}$, i.e.,

$$G = I_{n/m} \otimes \hat{G}.$$

Clearly a codeword of the code $\mathcal{C}$ with the generator matrix G is given by n/m codewords of the code $\hat{\mathcal{C}}$ concatenated side-by-side. The probability of error of $\mathcal{C}$ is therefore, by union bound, at most

$$\frac{n}{m} 2^{-E_L(p,\epsilon)m} = \frac{nE_L(p,\epsilon)}{(1+\alpha)n^{1+\alpha}\log n} = \frac{E_L(p,\epsilon)}{(1+\alpha)n^{\alpha}\log n}.$$

However, notice that the generator matrix has row weight bounded above by $m = \frac{1+\alpha}{E_L(p,\epsilon)} \log n$. Hence we have constructed a code with update efficiency $\frac{1+\alpha}{E(p,\epsilon)} \log n$, and rate $1-\mathrm{h}(p)-\epsilon$ that achieves a probability of error $< \frac{E(p,\epsilon)}{(1+\alpha)n^{\alpha}\log n}$ on a BSC(p).

Moreover the parity-check matrix of the resulting code will be block-diagonal with each block being the parity-check matrix of the code $\hat{\mathcal{C}}$. The parity-check matrix of the overall code has row-weight $O(\log \frac{1}{\epsilon})$. Hence, any codeword symbol can be recovered from at most $O(\log \frac{1}{\epsilon})$ other symbols by solving one linear equation. Therefore we have the following result.

Theorem 1: There exists a family of linear code $\mathcal{C}_n$ of length n and rate $1-\mathrm{h}(p)-\epsilon$, that have a probability of error over BSC(p) going to 0 as $n \to \infty$, and has update-efficiency $O(\log n/E_L(p,\epsilon))$ and local recoverability $O(\log \frac{1}{\epsilon})$.

Hence it is possible to simultaneously achieve local recovery and update-efficiency with a capacity-achieving code on BSC(p). Similar result follows for BEC(p).

B. Impossibility result for local recovery

In this section we concentrate on the converse results regarding local recovery properties of a code. Here, it can be noted that there are several possible definitions of local recovery. The simplest is perhaps the one in Defn. 2, to insist that for each codeword symbol, there is a set of at most r codeword positions that need to be queried to recover the given symbol with certainty. A weaker definition could allow adaptive queries, i.e., the choice of which r positions to query could depend on the values of previously queried symbols. Finally, one could ask that instead of obtaining the value of the codeword symbol with certainty, one obtains the value with some probability significantly higher than .5. For simplicity, we sketch the arguments here for the simplest definition, i.e., Defn. 2. The argument can easily be extended to the other definitions, except for some cases that will be explicitly mentioned later.

For the converse results, we prove our theorem for the binary erasure channel. We show that any code with a given local recoverability has to have rate bounded away from capacity to provide arbitrarily small probability of error, when used over the binary erasure channel.

In particular, we show below that, for any code, including non-linear codes, local recoverability at a gap of ϵ to capacity on the BEC must be at least $\Omega(\log \frac{1}{\epsilon})$, proving that the LDPC construction of the above section is simultaneously optimal to within constant factors for both update efficiency and local recovery.

The converse is based on an entropy argument. The idea is to show that if a code has local recovery complexity $c \log \frac{1}{\epsilon}$ for a suitable constant c, then, with overwhelming probability, the entropy of the output after a codeword is transmitted over a binary erasure channel with erasure probability p is less than $n(1-p-\epsilon)$. Thus, the rate of the code must be less than $(1-p-\epsilon)$, or the error probability will be non-vanishing, e.g., by Fano's inequality.

Theorem 2: For any code $\mathcal{C}$ of length n and rate $1-p-\epsilon$ that achieves probability of error less than δ for any $\delta > 0$ when used on a BEC(p), its local recoverability is at least $c \log \frac{1}{\epsilon}$, for some constant $c > 0$.

Proof: Let $\mathcal{C}$ be a code of length n and size 2^{nR} that has local recoverability r. Let T be the set of coordinates such that the number of query positions required to recover these coordinates appear before them. To show that such an ordering exists with $|T| \geq \frac{n}{r+1}$ we can randomly and uniformly permute the coordinates of $\mathcal{C}$; see that the expected number of such coordinates is $\frac{n}{r+1}$. Let us, without loss of generality, assume that $\mathcal{C}$ has such property, i.e., $|T| \geq \frac{n}{r+1}$.

Assume $I \subseteq \{1,\ldots,n\}$ be the set of coordinates erased by the BEC and $\bar{I} = \{1,\ldots,n\} \setminus I$. Let $\boldsymbol{x} \in \mathcal{C}$ be a randomly and uniformly chosen codeword. $\boldsymbol{x}_I$ and $\boldsymbol{x}_{\bar{I}}$ denote the projection of $\boldsymbol{x}$ on the respective coordinates. $H(\boldsymbol{x}_{\bar{I}})$ is the entropy of the un-erased coordinates and is a random-variable (with respect to the random choice of I by the BEC).

Suppose, the number of elements of T that has all their r recovery positions un-erased is u. Then, these elements do not contribute anything toward the entropy of $\boldsymbol{x}_{\bar{I}}$. Hence,

$$H(\boldsymbol{x}_{\bar{I}}) \leq |\bar{I}| - u.$$

But, $\mathbb{E}u \geq (1-p)^r |T|$. Therefore,

$$\mathbb{E}H(\boldsymbol{x}_{\bar{I}}) \leq n(1-p) - (1-p)^r \frac{n}{r+1}.$$

Now, because the entropy is a 1-Lipschitz functional of the independent random variables (erasures introduced by the channel), we can use Azuma's inequality [1] to see,

$$\Pr\left(H(\boldsymbol{x}_{\bar{I}}) > n(1-p) - (1-p)^r \frac{n}{r+1} + \alpha n\right) \leq e^{-\frac{\alpha^2 n}{2}}.$$

If we set $r = \frac{\log \frac{1}{(r+1)(\epsilon+\alpha)}}{\log \frac{1}{1-p}}$, then

$$\Pr\left(H(\boldsymbol{x}_{\bar{I}}) > n(1-p-\epsilon)\right) \leq e^{-\frac{\alpha^2 n}{2}}.$$

This indeed means that for a suitable constant c, if $r \leq c \log \frac{1}{\epsilon}$, then with very high probability $H(\boldsymbol{x}_{\bar{I}}) \leq n(1-p-\epsilon)$.

But, as $H(\boldsymbol{x}_{\bar{I}} \mid \boldsymbol{x}) = 0$, we have, $H(\boldsymbol{x} \mid \boldsymbol{x}_{\bar{I}}) = H(\boldsymbol{x}) - H(\boldsymbol{x}_{\bar{I}}) = nR - H(\boldsymbol{x}_{\bar{I}})$. Using Fano's inequality [5], the probability of error is bounded away from zero as long as $R \geq 1 - p - \epsilon$. This proves the claim.

■

Remark: This proof can be extended to the case when local recovery has to be guaranteed with certain probability, as opposed to being deterministic. However Fano's inequality shows the probability of error to be bounded away from 0, not to be close to 1. Note that, for the case of exact (deterministic) recovery, the above argument can be extended to show that the probability of error is not only bounded away from 0, but goes indeed to 1 (that is, an strengthening of the Fano's inequality argument is possible).

III. Rate-distortion

The dual problem of what were considering so far in this paper is the lossy source coding with update-efficiency and local recovery. Update-efficient codes with only lossless source compression has been considered before in the paper [10].

The rate-distortion function $R(D)$ of a source code expresses the optimal (smallest) rate achievable given a normalized distortion D. The formal descriptions can be found in any standard textbook of information theory (eg., [5]).

The main question, in the spirit of this paper, to be asked is, if we allow a rate slightly above the rate-distortion function, i.e., $R(D)+\epsilon$, then what is the local recoverability and update-efficiency (as defined in Defn.1 and 2) in terms of ϵ (possibly the length n as well) required to achieve the normalized distortion D.

It can be shown that local recoverability also grows as $\Omega(\log(\frac{1}{\epsilon}))$ in this case. This is a corollary of results for LDGM codes (Theorem 5.4.1 from [4]), and the proof already applies to non-linear codes. LDGM codes also show that $O(\log(\frac{1}{\epsilon}))$ recovery complexity is achievable.

Update efficiency for rate-distortion coding remains an open question. Update efficiency of $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$ can be achieved via random codes, but it is unclear that this is optimal. In particular, it is unclear that the update efficiency has to scale with ϵ at all.

Remark: For general rate-distortion problems, random coding would only achieve update efficiency $O(\frac{1}{\epsilon^2})$, but for the special case of a uniform source under Hamming distortion, the improved bound above can be achieved.

References

[1] N. Alon, J. Spencer, *The Probabilistic Method,* Wiley-Interscience, 2000.

[2] N. P. Anthapadmanabhan, E. Soljanin, S. Vishwanath, "Update-efficient codes for erasure correction," *48th Annual Allerton Conference on Communication, Control, and Computing,* pp. 376–382, October, 2010.

[3] V. Cadambe, A. Mazumdar, "Codes for distributed storage," *draft*, 2013.

[4] V. Chandar, *Sparse Graph Codes for Compression, Sensing, and Secrecy*, Ph.D. Thesis, MIT, 2010.

[5] T. Cover, J. Thomas, *Elements of Information Theory*, 2nd Ed., Wiley-Interscience, 2006.

[6] R. G. Gallager, *Low Density Parity Check Codes*, Monograph, M.I.T. Press, 1963.

[7] P. Gopalan, C. Huang, H. Simitci, S. Yekhanin, "On the locality of codeword symbols," Allerton, 2011.

[8] S. Litsyn, V. Shevelev, "On ensembles of low-density parity-check codes: asymptotic distance distributions," *IEEE Transactions on Information Theory*, vol. 48, no. 4, pp. 887–908, April 2002.

[9] A. Mazumdar, G. W. Wornell, V. Chandar, "Update efficient codes for error correction," *IEEE International Symposium on Information Theory (ISIT)*, 2012 , pp 1558–1562.

[10] A. Montanari, E. Mossel, "Smooth compression, Gallager bound and nonlinear sparse-graph codes," in *Proc. 2008 IEEE Intl. Symp. on Information Theory (ISIT '08)*, Piscataway, NJ: IEEE Press, 2008, pp. 2474-2478.

[11] D. Papailiopoulos, A. G. Dimakis, "Locally repairable codes," *IEEE International Symposium on Information Theory (ISIT)*, 2012, pp. 2771–2775.