Lecture 15: Local codes for distributed storage

Sivakanth Gopi

November 18, 2019

In this lecture, we will discuss local codes designed specifically for distributed stroage applications, called *Local Reconstruction Codes (LRCs).* In distributed storage, data is distributed among several servers (each with its own processor and a hard drive with few terabytes capacity). There are two main problems which come up.

About a billion terabytes of data is stored in "the cloud". This data is broken into small parts and stored in servers each with a capacity of a few terabytes. And millions of such servers form a gigantic data center. Adding all the costs of network, power and cooling infrastructure, it costs billions of dollars to build and maintain such a data center. Without erasure coding, a three way replication seems unimaginably wasteful!

1. Hard drives crash. Typically the life of a hard drive is about 3 years. And many of them can crash simultaneously. This leads to permanent loss of data which is unacceptable.
2. Sometimes a server becomes unresponsive. This can happen when it is busy serving another request or getting an update or rebooting. This leads to delays in serving user requests which is acceptable but undesirable

There are three stages in distributed storage on how people dealt with these problems.

1. In the early stages of distributed storage, people just replicated all the data 3 times. This is the "golden standard" in terms of reliability. It can tackle both the above problems quite well. But it is highly inefficient in terms of storage cost, it increases by 3 times!
2. The next generation used error correcting codes. They encode the data using (say) Reed-Solomon code (each server has one symbol of a codeword). For example a $(6, 9)$ Reed-Solomon code[1] gives about as much reliability as 3-way replication. But our storage cost is only 1.5x. Why can't we use longer codes and get more reliability? A $(12, 16)$ Reed-Solomon code provides similar reliability but with lower storage cost. The reason is Problem (2) (also Problem (1) to some extent). To respond to a user request when some server is not responding, we must access 12 other servers. This becomes extremely slow.
3. Ideally we want a code which can protect from a large number of erasures while having a fast local recovery algorithm to recover from a small number of erasures. And we want to minimize redundancy. Local Reconstruction Codes are precisely such codes and they form the third generation.

[1] A $(6, 9)$ code has 6 data servers and 3 parity servers computed using a systematic Reed-Solomon code. Such a code will allow us recover information in any 3 servers from the remaining 6 servers.

Preliminaries

A linear code (subset) $C \subseteq \mathbb{F}^n$ of dimension k can be described in two ways:

- **Generator matrix:** $C = \{Gx : x \in \mathbb{F}^k\}$ for some $n \times k$ matrix G, called the generator matrix. If the rows of G are $v_1, v_2, \ldots, v_n \in \mathbb{F}^n$, then
$$C(x) = (\langle v_1, x, , \rangle \ldots, \langle v_n, x \rangle).$$
The columns of G form a basis for C.
- **Parity check matrix:** $C = \{y \in \mathbb{F}^n : Hy = 0\}$ for some $(n-k) \times n$ matrix H, called the parity check matrix. Thw rows of H form a basis for $C^\perp$.

If C has minimum distance d, we can recover $C(x)$ even after erasing $d-1$ coordinates. If a subset $S \subseteq [n]$ of coordinates of a codeword $C(x)$ are erased, we can recover $C(x)$ iff the columns of H in S are linearly independent. This is because $Hy = 0 \Rightarrow H|_S y_S = -H|_{\bar{S}} y_{\bar{S}}$; so we can find y_S given $y_{\bar{S}}$ iff $H|_S$ has full column rank. In particular, minimum distance of C is d iff every for every subset $S \subset [n]$ of size $d-1$, $H|_S$ is linearly independent.

Local Reconstruction Codes (LRCs)

We argued that the idea of locality is extremely useful in the context of coding for distributed storage. For the past few lectures, we have studied local codes which have local decoding and correction algorithms that tolerate a constant fraction of corruptions. This is quite powerful, but unfortunately we do not have good constructions of such codes and we have also proved lower bounds which show that these codes must have vanishing rates. So these aren't very useful in practice. So we will relax our requirements by seperating the normal and worst cases.

"Handle normal and worst cases separately as a rule, because the requirements for the two are quite different" - Butler W. Lampson in 'Hints for Computer System Design'

- **Normal case:** Typically, only one (or a few) server crashes or becomes unresponsive at a time. We want to fast data recovery in this case.
- **Worst case:** Several servers crash or becomes unresponsive at once. This in an extremely rare event. We don't care too much about speed, we just want that data is protected in this case.

This motivates the definition of LRCs. Throughout this lecture, we will only talk about linear codes.

Definition 1. *An LRC with locality ℓ is an error correcting code $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ which allows recovery of any erased codeword symbol by reading at most ℓ other codeword symbols.*

In some cases, when the bottleneck is network bandwidth, it also makes sense to optimize the total communication cost to reconstruct a crashed server. Codes designed to minimize the communication cost (instead of number of servers accessed) are called *Regenerating codes*.

More generally, one could ask for local recovery even when some $a \geq 1$ codeword symbols are erased. Most of the theory carries over to this more general setting. The following proposition is a generalization of singleton bound for LRCs.

Proposition 2. *If $C : \mathbb{F}^k \to \mathbb{F}^n$ is an LRC with locality ℓ and minimum distance d, then*

$$n \geq k + \left\lceil \frac{k}{\ell} \right\rceil + d - 2. \tag{1}$$

Note that the above bound reduces to the singleton bound when $\ell = k$.

Pyramid codes

Suppose $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is a systematic code i.e. the message x is part of the codeword $C(x)$. If we only want local recovery of message coordinates, then there is a very simple way to achieve the bound (1).

First encode $x \in \mathbb{F}_q^k$ using a systematic Reed-Solomon code with distance d. This will produce $d-1$ parity symbols. WLOG, we can assume that the first parity symbol is $\sum_{i=1}^{k} x_i$. We will replace this symbol with $\lceil k/\ell \rceil$ parity symbols given by

$$\left(\sum_{i=t\ell}^{t\ell+\ell-1} x_i \right)_{t=0,1,\ldots,\lceil k/\ell \rceil - 1}.$$

Thus is our final encoding $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ where

$$n = k + \lceil k/\ell \rceil + d - 2$$

which matches the bound (1). If any message symbol is erased, we can recover it by reading at most ℓ other symbols. And the minimum distance of C is d. But parity symbols do not have local recovery.

Tamo-Barg construction

Tamo and Barg [TB14] gave a construciton of an LRC which has local recovery for every codeword coordinate and meets the bound (1) over fields of size $O(n)$. We will describe the parity check matrix of the code. Let $r = \ell + 1$. Let $q \geq n + 1$ be a prime power such that r divides $q - 1$. Let γ be a generator for $\mathbb{F}_q^*$. Let $\alpha = \gamma^{(q-1)/r}$. We will now make some simplifying assumptions. We will also assume that r divides n and r divides $d - 2$. These are not crucial for the

To recover a coordinate from ℓ other coordinates, all $\ell + 1$ of them should satisfy a parity check equation. We say that such a set of coordinates form a local group of size $r = \ell + 1$. In an LRC, every coordinate is part of some local group of size r. It can be shown that any LRC which matches the bound 1 should have disjoint local groups under some divisibility conditions.

construction, they will make the presentation easier. The code $C = \{y : H'y = 0\}$ where

$$H' = \left[\begin{array}{c|c|c|c} 1\,1\,\ldots\,1 & 0 & \cdots & 0 \\ \hline 0 & 1\,1\,\ldots\,1 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & 1\,1\,\ldots\,1 \\ \hline B_0 & B_1 & \cdots & B_{(n/r)-1} \end{array}\right]. \tag{2}$$

$B_0, B_1, \ldots, B_{(n/r)-1}$ are $(d-2) \times r$ matrices defined as follows. Let

$$\beta_{ij} = \gamma^i \alpha^j = \gamma^{i+j((q-1)/r)}.$$

Since $0 \leq i < (n/r) \leq ((q-1)/r)$, all the β_{ij} are distinct. Define

$$B_i = \begin{bmatrix} \beta_{i1} & \beta_{i2} & \ldots & \beta_{ir} \\ \beta_{i1}^2 & \beta_{i2}^2 & \ldots & \beta_{ir}^2 \\ \vdots & \vdots & & \vdots \\ \beta_{i1}^{d-2} & \beta_{i2}^{d-2} & \ldots & \beta_{ir}^{d-2} \end{bmatrix}.$$

Claim 3. *C has minimum distance d.*

Proof. This is equivalent to showing that every $d-1$ columns of H' are linearly independent. Since row operations doesn't change column rank, it is enough to show this for

$$H'' = \left[\begin{array}{c|c|c|c} 1\,1\,\ldots\,1 & 1\,1\,\ldots\,1 & \cdots & 1\,1\,\ldots\,1 \\ \hline B_0 & B_1 & \cdots & B_{(n/r)-1} \end{array}\right].$$

But every $d-1$ columns of H'' form a Vandermonde matrix which is full rank. □

Claim 4. *If $k = dim(C)$ then*

$$n - k = \frac{n}{r} + (d-2) - \frac{(d-2)}{r}.$$

Proof. $\dim(C) = n - \text{rank}(H')$. Note that H' has $\frac{n}{r} + (d-2)$ rows. We will show that there are $\frac{d-2}{r}$ rows of H' which are linearly dependent on other rows. This proves that $\text{rank}(H') = \frac{n}{r} + (d-2) - \frac{(d-2)}{r}$. If r divides t then

$$\beta_{ij}^t = \gamma^{it}\alpha^{jt} = \gamma^{it}(\gamma^{jt/r})^{q-1} = \gamma^{it}.$$

So whenever r divides t, for every i, the entries of the t^{th} row of B_i are constant. Thus whenever r divides t, the t^th row of $[B_0\ B_1\ \ldots\ B_{(n/r)-1}]$ is spanned by first (n/r) rows of H'. Therefore $(d-2)/r$ rows of H' are linearly dependent on the first (n/r) rows. □

Rewriting $n - k = \frac{n}{r} + (d-2) - \frac{(d-2)}{r}$, we get $n = k + k/r + d - 2$ which is the bound (1). In general we will get

$$n - k = \left\lceil \frac{n}{r} \right\rceil + (d-2) - \left\lfloor \frac{(d-2)}{r} \right\rfloor$$

which will be at most 1 off from optimal dimension k possible from bound (1).

Beyond minimum distance: Maximal recoverability

We have constructed LRCs with the optimal minimum distance (1) for a given dimension k, length n and locality ℓ. But are they "optimal"? Suppose C, C' are LRCs with the same parameters n, k, ℓ and same minimum distance d. Also assume that they have the same local groups. Are they equally good? No! Both of then can correct any pattern of at most $d-1$ erasures. But it is possible that C can recover from some erasure pattern of more than $d-1$ erasures, but C' cannot. And similarly C' can correct some patterns what C cannot. Then how do we compare different codes?

Luckily, it turns out there is an "optimal" LRC which corrects every erasure pattern that is correctable by any other LRC with the same parameters n, k, ℓ, d and the same local groups. Such an LRC is called a Maximally Recoverable LRC (MR LRC). Thus MR LRCs provide the strongest possible reliability guarantees given the locality constraints defining the shape of the parity check matrix. So why do MR LRCs exist? It is easy to see this from the parity check view.

Let us consider the more general setting where we want locality ℓ even when there are 'a' erasures, i.e., we can recover any 'a' erased coordinates by reading some ℓ coordinates. Let $r = a + \ell$. Suppose r divides n. In what follows we refer to subsets $\{r(i-1)+1, \ldots, ri\}$ of the set of code coordinates $[n]$ as local groups. There are $g = n/r$ local groups and each such group has size r. The parity check matrix H of a such an LRC has the following form

$$H = \left[\begin{array}{c|c|c|c} A_1 & 0 & \cdots & 0 \\ \hline 0 & A_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A_g \\ \hline B_1 & B_2 & \cdots & B_g \end{array}\right]. \tag{3}$$

Here $A_1, A_2, \cdots, A_g$ are $a \times r$ matrices over $\mathbb{F}_q$, $B_1, B_2, \cdots, B_g$ are $h \times r$ matrices over $\mathbb{F}_q$. The rest of the matrix is filled with zeros. Every matrix $\{A_i\}_{i \in [g]}$ is a parity check matrix of an $[r, r-a, a+1]$ MDS code; $A_1, A_2, \ldots, A_g$ are called the *local parity check matrices*.

This implies that we can recover from any a erasures in a local group by accessing $\ell = r - a$ symbols from the same local group. The bottom h rows of H, called *global parity checks*, increase the code co-dimension from ag to $ag + h$ and increase reliability when there are more erasures that the local parities cannot handle. The minimum distance of an LRC is at most $a + h + 1$ if $a + h < r$; this is because we cannot correct $a + h + 1$ erasures in a local group. But MR LRCs can correct many patterns way beyond minimum distance.

Proposition 5. *There exist MR LRCs over large enough fields.*

Proof. Imagine the entries of $A_1, A_2, \ldots, A_g$ and entries of $B_1, B_2, \ldots, B_g$ as distinct variables; say X represents a vector of all these variables. Now an erasure pattern S of size $ag + h$ is correctable (by some LRC as in (3)) iff $\det(H(X)|_S)$ is a non-zero polynomial in these variables. If $\det(H(X)|_S) = 0$, then no LRC with parity check matrix as in (3) can correct the erasure pattern S.

We will show that if we assign random values to the variables from a large enough finite field $\mathbb{F}_q$, then the resulting code will be maximally recoverable with high probability. If $\det(H(X)|_S)$ is a non-zero polynomial, then it will remain non-zero after substituting random $\mathbb{F}_q$ values with probability at least $1 - deg(\det(H(X)|_S))/q = 1 - (ag + h)/q$. There are atmost $\binom{n}{ag+h}$ such sets S. By union bound, if $q \gg (ag + h)\binom{n}{ag+h}$, then with high probability every erasure pattern that is correctable (by some code), will be correctable by our random code as well. So it is maximally recoverable. □

So what are the maximal erasure patterns that are correctable? These are precisely erasure patterns you can obtain by erasing 'a' coordinates in each local group and 'h' additional coordinates anywhere. So we can alternatively define MR LRCs as follows.

Definition 6. *Let C be an arbitrary (n, r, h, a, q)-LRC with parity check matrix of the form 3. We say that C is maximally recoverable if for any set $E \subseteq [n]$, $|E| = ga + h$, where E is obtained by selecting a coordinates from each of g local groups and then h more coordinates arbitrarily; E is correctable by the code C.*

We have seen that MR LRCs exists over large enough fields. But in practice, we require that field size is small. Finite field arithmetic over large fields makes both encoding and recovery extremely slow. Thus we want to construct MR LRCs over as small fields as possible. And fields of characteristic are preferred.

Open Problem 7. *What is the smallest field size q required for the existence of an (n, r, h, a, q)-MR LRC as in (3)? And can we construct them explicitly over such small fields?*

The random construction we have seen requires fields of size $q \gg n^{ag+h}$. This is too big. There are also multiple explicit constructions available which do much better than random [GHJY14, GYBS17, MPK19, GJX19, GGY17]. They bounds they acheive are uncomparable. Some bounds are better in some range of parameters over others. To simplify presentation, let us assume that a, h are constants and r is growing slowly with n, like say $r = n^{\varepsilon}$ for some fixed $0 < \varepsilon < 1$. The best construction in this regime is due to [GYBS17] require fields of size

$$q \lesssim \max\{O(n/r), O(r)^{a+h}\}^h.$$

The best lower bound on the field size is from [GGY17],

$$q \gtrsim_{a,h} n \cdot r^{\min\{a,h-2\}}.$$

A really interesting open problem is the following.

Open Problem 8. *If r, a, h are constant, are there MR LRCs over fields of linear size, i.e., $q \lesssim_{r,a,h} n$?*

Maximal recoverability can be defined in more general context. Given some linear constraints on the parity check matrix (called a code topology), we could define maximally recoverable codes with this topology. See [GHK+17] for more on different interesting topologies that are used in practice. This is a new area with several open questions. For example, we do not even what patterns are correctable by tensor codes!

References

[GGY17] Sivakanth Gopi, Venkatesan Guruswami, and Sergey Yekhanin. On maximally recoverable local reconstruction codes. *CoRR*, abs/1710.10322, 2017. Available at `http://arxiv.org/abs/1710.10322`.

[GHJY14] Parikshit Gopalan, Cheng Huang, Bob Jenkins, and Sergey Yekhanin. Explicit maximally recoverable codes with locality. *IEEE Transactions on Information Theory*, 60(9):5245–5256, 2014.

[GHK+17] Parikshit Gopalan, Guangda Hu, Swastik Kopparty, Shubhangi Saraf, Carol Wang, and Sergey Yekhanin. Maximally recoverable codes for grid-like topologies. In *28th Annual Symposium on Discrete Algorithms (SODA)*, pages 2092–2108, 2017.

[GJX19] Venkatesan Guruswami, Lingfei Jin, and Chaoping Xing. Constructions of maximally recoverable local reconstruction codes via function fields. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 68:1–68:14, 2019.

[GYBS17] Ryan Gabrys, Eitan Yaakobi, Mario Blaum, and Paul Siegel. Construction of partial MDS codes over small finite fields. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1–5, 2017.

[MPK19] Umberto Martínez-Peñas and Frank R Kschischang. Universal and dynamic locally repairable codes with maximal recoverability via sum-rank codes. *IEEE Transactions on Information Theory*, 2019.

[TB14] Itzhak Tamo and Alexander Barg. A family of optimal locally recoverable codes. *IEEE Transactions on Information Theory*, 60:4661–4676, 2014.