A Student’s View of Local Repair and Some Recent Works

Arya Mazumdar

University of Minnesota
Lightning Talk: An information storage graph

- Can store 1 bit in each vertex
- Content of any vertex can be determined by looking at the contents of its neighbors

\[
\begin{align*}
N(1) &= \{2, 3, 4, 5\} \\
N(2) &= \{1, 3\} \\
N(3) &= \{1, 2, 4\} \\
N(4) &= \{1, 3, 5\} \\
N(5) &= \{1, 4\}.
\end{align*}
\]

How many bits of information can be stored in this graph?
Locality of Codeword Symbols

Any symbol of a codeword must be reconstructed using at most \( r \) other symbols (Gopalan, Huang, Simitci, Yekhanin, 2011)

\[
d \leq n - k - \left\lfloor \frac{k}{r} \right\rfloor + 2.
\]

1. This bounds is not that difficult to get – nonlinear/linear etc. does not matter
2. $H(X_1 | X_{i1} X_{i2} \ldots X_{ir}) = 0$

Hmm... so this is exactly like **LDPC** codes! Is there any difference?
Erasure channel

- $x \in C$ is a randomly and uniformly chosen codeword.
- $I$ is the set of coordinates erased by BEC.
- Given $I$, $H(x_I | x) = 0$.
- $H(x | x_I) = H(x) - H(x_I) = nR - H(x_I)$.
- Using Fano’s inequality, the probability of error is bounded away from zero as long as $R \geq 1/n \cdot H(x_I)$. 
The output entropy

$T = \{\text{coordinates: the number of query positions required to recover these coordinates appear before them}\}$.

An ordering exists such that: $|T| \geq \frac{n}{r+1}$ (randomly permute the coordinates to see that).

Take that ordering of coordinates of our code.

Suppose the number of coordinates in $T$ that have all their recovery positions un-erased is $u$.

$H(x_i) \leq |\bar{I}| - u$. But, $\mathbb{E} u \geq (1 - p)^r |T|$.
Concentration of output entropy

- $\mathbb{E}H(x_\tilde{i}) < n(1 - p) - (1 - p)^r \frac{n}{r+1}$.
- $H(x_\tilde{i})$ is a 1-Lipschitz functional of the independent random variables (erasures introduced by the channel).
- Use Azuma’s inequality.

With high probability, $H(x_\tilde{i}) \leq n(1 - p) - (1 - p)^r \frac{n}{r+1}$. 
Bottom-line

• Hmm.. like LDPC codes

Rate = 1 - p - ε

local recoverability is at least $c \log \frac{1}{\epsilon}$

• So they might have a (sparse)graph representation?
How much storage is possible?

Storage graph: $G(V, E)$
$V = \{1, 2, \ldots, n\}$.

- $N(1) = \{2, 3, 4, 5\}$
- $N(2) = \{1, 3\}$
- $N(3) = \{1, 2, 4\}$
- $N(4) = \{1, 3, 5\}$
- $N(5) = \{1, 4\}$.

Recovery Sets
Formally ...

- $V = \{1, 2, \ldots, n\}$
- $X_1, X_2, \ldots, X_n$: content of the vertices, $X_i \in \mathbb{F}_q$, $i = 1, \ldots, n$
- RDSS code $C \subseteq \mathbb{F}_q^n$
- A set of deterministic recovery functions, $f_i : \mathbb{F}_q^{\lvert N(i) \rvert} \to \mathbb{F}_q$ for $i = 1, \ldots, n$
- For any codeword $(X_1, X_2, \ldots, X_n) \in \mathbb{F}_q^n$,

$$X_i = f_i(\{X_j : j \in N(i)\}), \quad i = 1, \ldots, n.$$ 

Capacity $= \log |C|$
1 bit

1 Bit of Storage
2 bits

One bit of data in each node (server)

Content of any server: Recoverable from the linked servers
2 bits continued ...

2 Bits of Storage
2.32193 bits ...
Reminder from lightning talk

Codewords
{00000, 01100, 00011, 11011, 11101}

\[ X_1 = X_2 \land X_5 \]
\[ X_2 = X_1 \lor X_3 \]
\[ X_3 = X_2 \land \bar{X}_4 \]
\[ X_4 = \bar{X}_3 \land X_5 \]
\[ X_5 = X_1 \lor X_4. \]
It’s quite practical (ask Google)

- All servers are not connected to each other
- Some links are easy to establish
- Consideration for
  - Physical Proximity
  - Architecture
  - Platform
  - Connections
Reduction to a dual problem! Index coding

- A set of n users \( \{1,2, \ldots , n\} \)
- Each want some information \( \{x_1, x_2, \ldots , x_n\} \)
- Each has some information \( \{S_1, S_2, \ldots , S_n\} \) (side information graph G)
  \[ S_i \subseteq \{x_1, x_2, \ldots , x_n\} \]

How many bits should be BROADCAST so that everyone gets what they need? INDEX(G)

Bar-Yossef, Birk, Jayram, Kol, ’06
Alon, Hasidim, Lubetzky, Stav, Weinstein, ’08
Lubetzky, Stav, ‘09
RDSS $\neq n - \text{INDEX}$

Codewords
\[ \{00000, 01100, 00011, 11011, 11101\} \]

\[ \text{CAP}_2(\hat{G}) = \log_2 5 \text{ and } \text{INDEX}_2(G) = 3 \text{ (achieved by linear functions!)} \]
RDSS and INDEX

- An index code of length $\ell \Rightarrow$ an RDSS code of dimension $n - \ell$
- Easy to see.
- An RDSS code of dimension $k \Rightarrow$ An index code of length $n - k + \log_q(\ln 2 + k \ln q)$ for the side information graph $G$.

\[ n - \text{CAP}_q(G) \leq \text{INDEX}_q(G) \leq n - \text{CAP}_q(G) + \log_q \left( \min\{n \ln q, 1 + \text{CAP}_q(G) \ln q\} \right). \]
A sketch
A sketch

$C_i = C + \chi_i$

$\bigcup_{i=1}^{m} C_i = F^n_q$
A sketch

Index code: Broadcaster just transmit the index of the translation
Indeed

- Random translation works –

- There exists a set of translations that forms a linear subspace!
- A random linear subspace also works as a good set of translations!
- There is no loss in dimensionality!

- Follows from the same argument: A linear code can be a good covering code (Goblick 1962; Delsarte, Piret, 1986; Cohen, 1983, Berger 1971)
Also: see Blinovskii1990, CHLL Covering Codes Book

- It is useful when decoding the index code.
Network information flow

- Index coding is the **hardest instance** of network coding [RouayhebEtAl2010, LangbergEtAl2012, BlasiakKleinbergLubetzky2011] Converse results regarding polynomial gap in approximation

- For general network coding, **only linear function capacity** is known to be NP Hard [LehmanRasala2004]

- RDSS is coding theoretic dual of INDEX : A Basic problem
Undirected graphs

For undirected graph:

\[
\text{CAP} \leq \text{Minimum Vertex Cover} \quad (\text{WHY?})
\]

Algorithm:

1. Construct a Max. Matching of the graph
2. Replicate data on both vertices of an edge in the matching

Max. Matching \(\leq\) Min. Vertex Cover \(\leq\) 2 * Max. Matching

Bipartite graphs (Optimal, König’s Theorem)

Better than this 2-approximation in poly-time: Unlikely – will imply better approximation of Min. Vertex cover –
Directed graphs

For directed graph:

\[ \text{CAP} \leq \text{Feedback Vertex Set (FVS)} \]

Main algorithm (Try to generalize Matching):

Construct a vertex disjoint cycle packing
Each cycle may store a symbol

Unclear: The cycle-packing number and FVS (Younger’s conjecture – Erdos-Posa – Reed-Robertson-Seymour-Thomas)

Seymour (1995): Fractional cycle packing
approximates FVS
Store a vector in each cycle
Directed graph

Seymour (1995): Fractional cycle packing

The fractional packing (exponential number of variables) can be solved (because of existence of a separation oracle for the dual problem) in polytime.

We can use a vector code to implement the fractional cycle packing. Every vertex stores a total of $p$ bits and divides this storage according to the fractional packing of cycles going through this vertex.
RDSS with minimum distance

• Consider an independence set $U$ such that size of neighborhood $N(U) < k$
• The code restricted to $U$ and $N(U)$ must have entropy $< k$
• Distance of the code is at most $n$ minus size of $U$ plus $N(U)$

$$d \leq |V| - k + 1 - \max_{u \in \mathcal{I}(G) : |N(u)| \leq k - 1} |U|$$
Say an $r$-regular graph ..

An independent set of size $(k-1)/r$ ..
Plug that in..

$$d \leq n - k - \left\lfloor \frac{k}{r} \right\rfloor + 2.$$
Acknowledgements

Collaborators in distributed storage: Venkat Chandar, Greg Wornell, Viveck Cadambe, Ankit Rawat, Sriram Vishwanath, Abhishek Agarwal

Apologies: Entire literature of Regenerative Code

Those who were not cited during the talk: (Dimakis, Shanmugam), (Chaudhary, Langberg, Sprintson), Tamo, Barg, Papailiopoulos