Codes, Graphs, and Service Capacity

Austin Allen [1] Dr. Emina Soljanin [2]

[1]Carnegie Mellon University

[2]Rutgers University

July 12th, 2019

Motivation

There is always demand for storage systems that are reliable and available (latency and service capacity) to handle large amounts of data:

- Amazon (could lose sales)
- Google (page views can drop)
- Gaming (ping could be really high)
- Virtual Reality (not realistic enough)

Why coding?

Most people associate Coding Theory with **error correction**, however for us coding is applied to **distributed storage systems**

Problem formulation

Service Capacity problem

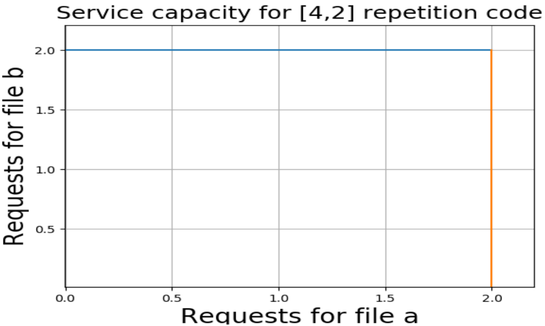
System model:

- We want to store k files $f_1, f_2, ..., f_k$ redundantly across n servers.
- Each file is of the same size and each server stores the same amount of data.
- Requests to download file f_i arrive at a rate of λ_i.
- Each server has a capacity of 1.

Objective: Given some redundancy scheme applied to the servers in the system, what is the set of rates $\lambda_1, ..., \lambda_k$ that can be supported by the system.

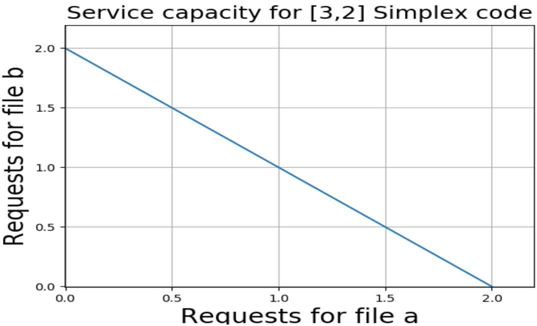
Repetition scheme

Suppose we wanted to store 2 files redundantly across 4 servers. We can use the [4,2] Repetition code $(a, b) \rightarrow (a, a, b, b)$.



Simplex scheme

Suppose we wanted to store 2 files redundantly across 3 servers. We can use the [3,2] Simplex code $(a, b) \to (a, b, a + b)$.



Research objective

- Service capacity has been considered in the continuous setting.
- Slicing of node capacity used.
- Coding Theory is discrete in nature.
- What if we considered service capacity in the discrete setting?
- No slicing, only taking whole capacity.
- Graph theory useful.
- What are connections to graph-theoretic tools and fundamental questions in load-balancing?

Recovery groups

Definition (Recovery group)

Suppose we have a linear code $\mathcal{C}$ and a generator matrix G for $\mathcal{C}$. If any symbol can be recovered by some linear combination of the columns of G, then those columns form a **recovery group** for some symbol.

For our purposes we are interested in recovery of **information symbols**.

Graph construction (AABS., 2019)

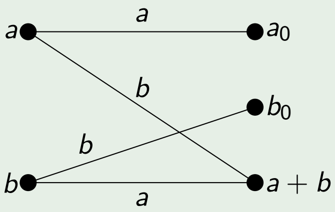
Graph requirements

- Linear $[n, k]_q$ code $\mathcal{C}$ with *generator matrix* $\mathbf{G}$.
- Vertices labeled by columns of $\mathbf{G}$.
- Vertices connected if they form a recovery group for information symbol.
- Add dummy vertices to connect to the systematic vertices.
- Edges are "labeled" by information symbols.
- May result is multiple edges between vertices.
- May result in a hypergraph.

[3,2] Simplex Code

Example

Suppose we label the columns of the generator matrix a, b and $a + b$. Then we construct the graph formed by the [3, 2] Simplex Code as:



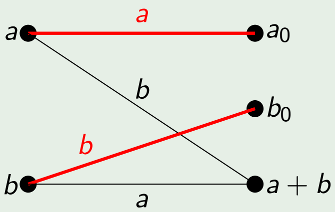
Matching

Definition (Matching)

Given a graph $G = (V, E)$, a matching is a set of edges $e \in E(G)$ such that no two edges share the same vertex.

Example

We construct the graph formed by the [3, 2] Simplex Code as

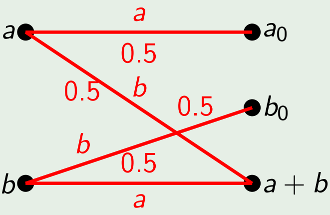


Definition (Matching)

Given a graph $G = (V, E)$, a matching is a set of edges $e \in E(G)$ such that $\forall v \in V(G)$, the sum of the edge weights for each of its incident edges is less than or equal to 1.

Example

We construct the graph formed by the $[3, 2]$ Simplex Code as



Matching LP

Maximize $\sum_{e \in E(G)} x_e$ subject to:

$$\sum_{e \perp v} x_e \leq 1 \tag{1}$$

$$x_e \in \mathbb{N}, \forall e \in E(G) \tag{2}$$

Fractional LP

Maximize $\sum_{e \in E(G)} x_e$ subject to:

$$\sum_{e \perp v} x_e \leq 1 \tag{3}$$

$$x_e \geq 0, \forall e \in E(G) \tag{4}$$

Whenever G is bipartite, the matching number equals the fractional

Linear Programming

Service Capacity as LP (Soljanin,2019)

We denote by $R_{i_1}, \ldots, R_{t_i}$ the t_i disjoint recovery groups of file f_i and by λ_i the portion of requests for file f_i that are assigned to the recovery group $R_{i,j}, j = 1, \ldots, t_i$. Then the achievable service rate region of such as system is a set of vectors $\lambda = (\lambda_1, \ldots, \lambda_k)$ for which there exist $\lambda_{i,j}$ satisfying the following constraints:

$$\sum_{j=1}^{t_i} \lambda_{i,j} = \lambda_i, 1 \leq i \leq k \tag{5}$$

$$\sum_{j=1}^{k} \sum_{1 \leq j \leq t_i} \lambda_{i,j} \leq 1_\ell, 1 \leq \ell \in R_{i,j} \leq n \tag{6}$$

Equivalent optimization (AABS., 2019)

Theorem (Frac LP = Service LP)

Suppose we have a linear $[n, k]_2$ code $\mathcal{C}$ with generator matrix $\boldsymbol{G}$, and consider the graph created by the columns of $\boldsymbol{G}$. The service capacity linear program for $\mathcal{C}$ is the equivalent to the fractional matching linear program for its associated graph.

- This means that fractional matchings in the graph gives us achievable points in the service capacity region.
- This means integral matchings in the graph tells us about load-balancing properties.

Important connection (A**A**BS.,2019)

Lemma (Simplex code graphs)

Connecting two vertices if they form a recovery group for an information symbol is the same as having a bipartite graph where L has the column vectors with an odd number of 1's and R has all of the column vectors with an even number of 1's and the dummy vertices.

What does this mean?

- Simplex codes form bipartite graph.
- fractional matching = integral matching.
- Promising for codes that admit bipartite graphs.
- Deeper connection with batch codes for load-balancing.

Future work

- Continue looking at connections using discrete tools.
- Look at batch properties of codes that can be used in distributed storage: Simplex (we can say something), MDS (Reed-Solomon), first-order Reed-Muller, Switch.
- Overall goal is to show that designing a coding scheme is equivalent to using a batch code.

References

Aktas et. al.
On the service capacity region of accessing erasure coded content.
page 8, 2017.
55th Annual Allerton Conference on Communication, Control, and Computing.

Baumbaugh et. al.
Batch codes from hamming and reed-muller codes.
Journal of Algebra Combinatorics Discrete Structures and Applications, page 13, 2017.

Anderson et.al.
Service rate region of content access from erasure coded storage.
In *Proceedings of the 2018 Information Theory Workshop*, page 5, 2018.

Frederic Havet.
Fractional relaxation.

F.J. MacWilliams and N.J.A. Sloane.
The Theory of Error Correcting Codes, volume 16.
North-Holland Publishing Company, 1981.

Vitaly Skachek.
Batch and PIR codes and their connections to locally repairable codes.
Survey, 2017.

Emina Soljanin.
Service rates of codes and vertex covers of graphs.
Joint Mathematics Meetings, 2019.

Acknowledgements

A special thanks to the National Science Foundation for supporting this research through the NSF grant CCF-1852215, as well as my mentors Dr. Emina Soljanin, Mehmet Aktas, and Amir Behrouzi-Far, and DIMACS for hosting.

DIMACS
Center for Discrete Mathematics and Theoretical Computer Science
Founded as a National Science Foundation Science and Technology Center