On a Duality Between Recoverable Distributed Storage and Index Coding

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

Abstract—In this paper, we introduce a model of a single-failure locally recoverable distributed storage system. This model appears to give rise to a problem seemingly dual of the well-studied index coding problem. Although the relation between the dimensions of an optimal index code and optimal distributed storage code of our model can be established based on previous works, we give a completely coding theoretic proof of this apparent duality result.

I. INTRODUCTION

Recently, local repair property of error-correcting codes is the center of a lot of research activity. In a distributed storage system, a single server failure is the most common error-event, and in that case, the aim is to reconstruct the content of the server from as few other servers as possible (or by downloading minimal amount of data from other servers). The study of such regenerative storage systems was initiated in [6] and then followed up in several recent works. In [8], a particularly neat characterization of a local repair property is provided. It is assumed that, each symbol of an encoded message is stored at a different node in the network (since the symbol alphabet is unconstrained, a symbol could represent a packet or block of bits of arbitrary size). Accordingly, [8] investigates codes allowing any single symbol of any codeword to be recovered from at most a constant number of other symbols of the codeword, i.e., from a number of symbols that does not grow with the length of the code.

The work of [8] is then further generalized to several directions and a number of impossibility results regarding, as well as construction of, locally repairable codes were presented (see, for example, [3], [9], [13], [14], [16]), culminating in very recent construction of [15].

However, the topology of the network of distributed storage system is missing from the above definition of local repairability. Namely, all servers are treated equally irrespective of their physical positions, proximities, and connections. Here we take a step to include that into consideration. We study the case when the topology of the storage system is fixed and the network of storage is given by a graph. In our model, the servers are represented by the vertices of a graph, and two servers are connected by an edge if it is easier to establish up-or-down link between them, for reasons such as physical locations of the servers, architecture of the distributed system or homogeneity of softwares, etc. It turns out that, our model is closely related to the following index coding problem on a side information graph. In this paper, we formalize this relation.

A. Index Coding

A very natural “source coding” problem on a network, called the index coding, was introduced in [2], and since then is a subject of extensive research. In the index coding problem a side information graph \( G(V, E) \) is given. Each vertex \( v \in V \) represent a receiver that is interested in knowing an uniform random variable \( Y_v \in \mathbb{F}_q \). For any \( v \in V \), define \( N(v) = \{ u \in V : (v, u) \in E \} \) to be the neighborhood of \( v \). The receiver at \( v \) knows the values of the variables \( Y_u, u \in N(v) \). How much information should a broadcaster transmit, such that every receiver knows the value of its desired random variable? Let us give the formal definition from [2], adapted for \( q \)-ary alphabet here.

**Definition 1:** An index code \( \mathcal{C} \) for \( \mathbb{F}_q^n \) with side information graph \( G(V, E), V = \{1, 2, \ldots, n\} \), is a set of codewords in \( \mathbb{F}_q^n \) together with:

1) An encoding function \( f \) mapping inputs in \( \mathbb{F}_q^n \) to codewords, and

2) A set of deterministic decoding functions \( g_1, \ldots, g_n \) such that \( g_i(f(Y_1, \ldots, Y_n), \{Y_j : j \in N(i)\}) = Y_i \) for every \( i = 1, \ldots, n \).

The encoding and decoding functions depend on \( G \). The integer \( \ell \) is called the length of \( \mathcal{C} \), or \( \text{len}(\mathcal{C}) \). Given a graph \( G \) the minimum possible length of an index code is denoted by \( \text{INDEX}_q(G) \).

In [2], a connection has been made with the length of an index code to a quantity called the minrank of the graph. Suppose, \( A = (a_{ij}) \) be an \( n \times n \) matrix over \( \mathbb{F}_q \). It is said that \( A \) fits \( G(V, E) \) over \( \mathbb{F}_q \) if \( a_{ij} \neq 0 \) for all \( i \) and \( a_{ij} = 0 \) whenever \( (i, j) \notin E \) and \( i \neq j \).

**Definition 2:** The minrank of a graph \( G(V, E) \) over \( \mathbb{F}_q \) is defined to be,

\[
\text{minrank}_q(G) = \min\{\text{rank}_{\mathbb{F}_q}(A) : A \text{ fits } G\}.
\]  

(1)

It was shown in [2], that,

\[
\text{INDEX}_q(G) \leq \text{minrank}_q(G),
\]  

(2)

and indeed, \( \text{minrank}_q(G) \) is the minimum length of an index code on \( G \) when the decoding functions are all linear. The above inequality can be strict in many cases [1], [11].
In [1], the problem of index coding is further generalized. We only describe here what is important for our context. Only for this part, assume $q = 2$. To characterize the optimal size of an index code, [1] introduces the notion of a confusion graph.

Two input strings, $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{F}_2^n$ are called confusable if there exists some $i \in \{1, \ldots, n\}$, such that $x_i \neq y_i$, but $x_j = y_j$, for all $j \in N(i)$. In the confusion graph of $G$, total number of vertices are $2^n$, and each vertex represents a different $(0, 1)$-string of length $n$. There exists an edge between two vertices if and only if the corresponding two strings are confusable with respect to the graph $G$. The maximum size of an independent set of the confusion graph is denoted by $\gamma(G)$.

However, the confusion graph and $\gamma(G)$ in [1] were used as tools to characterize the the rate of index coding; they were not used to model any immediate practical problem. In this paper, we show that, this notion of confusable strings fits perfectly to the situation of local recovery of a distributed storage system. Namely, $\gamma(G)$, in our problem becomes the largest possible size of a locally recoverable code for a system with topology given by $G$.

B. Organization

The paper is organized in the following way. In Section II, we introduce formally the model of a recoverable distributed storage system inspired by [1]. The notion of an optimal recoverable distributed storage code given a graph and its relation to the optimal index code is also described here. In Section III, we provide a completely coding theoretic proof of the main duality relation of the index code and distributed storage code. Our proof is based on a covering argument of the Hamming space, and rely on the fact that for any given subset of the Hamming space there exists a translation of the set, that has very small overlap with the original subset. We end with some concluding remarks.

II. RECOVERABLE DISTRIBUTED STORAGE SYSTEMS

Consider the network of distributed storage, for example, one of Fig. 1. As mentioned in the introduction, the property of two servers connected by an edge is based on the ease of establishing a link between the server. It is also possible (and sensible, perhaps) to model this as a directed graph (especially when uplink and downlink constructions have varying difficulties). However, for the sake of clear presentation, here we restrict ourselves only to the case of simple undirected graphs.

If the data of any one server is lost, we want to recover it from the nearby servers, i.e., the ones with which it is easy to establish a link. This notion is formalized below.

Suppose, the graph $G(V, E)$ represents the network of storage. Each element of $V$ represents a server, and in the case of a server failure (say, $v \in V$ is the failed server) one must be able to reconstruct its content from its neighborhood $N(v)$.

Given, this constraint what is the maximum amount of information one can store in the system? Without loss of generality, assume $V = \{1, 2, \ldots, n\}$ and the variables $X_1, X_2, \ldots, X_n$ respectively denote the content of the vertices, where, $X_i \in \mathbb{F}_q, i = 1, \ldots, n$.

Definition 3: A recoverable distributed storage system (RDSS) code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with storage recovery graph $G(V, E), V = \{1, 2, \ldots, n\}$, is a set of vectors in $\mathbb{F}_q^n$ together with:

- A set of deterministic recovery functions, $f_i : \mathbb{F}_q^{N(i)} \rightarrow \mathbb{F}_q$ for $i = 1, \ldots, n$ such that for any codeword $(X_1, X_2, \ldots, X_n) \in \mathbb{F}_q^n$,

$$X_i = f_i((X_j : j \in N(i))), \quad i = 1, \ldots, n.$$  \hspace{1cm} (3)

Again, the decoding functions depend on $G$. The log-size of the code, $\log_q |\mathcal{C}|$, is called the dimension of $\mathcal{C}$, or dim$(\mathcal{C})$. Given a graph $G$ the maximum possible dimension of an RDSS code is denoted by $\text{RDSS}_q(G)$.

For example, consider the graph of Fig. 1 again. Here, $V = \{1, 2, 3, 4, 5\}$. The recovery sets of each vertex (or storage nodes) are given by:

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

Suppose, the contents of the nodes 1, 2, \ldots, 5 are $X_1, X_2, \ldots, X_5$ respectively, where, $X_i \in \mathbb{F}_q, i = 1, \ldots, 5$. Moreover, $X_1 = f_1(X_2, X_3, X_4, X_5), X_2 = f_2(X_1, X_3), X_3 = f_3(X_1, X_2, X_4), X_4 = f_4(X_1, X_3, X_5), X_5 = f_5(X_1, X_4)$.

Assume, the functions $f_1, i = 1, \ldots, 5$, in this example are linear. That is, for $\alpha_{ij} \in \mathbb{F}_q, 1 \leq i, j \leq 5$,

$$\begin{align*}
X_1 &= \alpha_{12} X_2 + \alpha_{13} X_3 + \alpha_{14} X_4 + \alpha_{15} X_5 \\
X_2 &= \alpha_{21} X_1 + \alpha_{22} X_3 \\
X_3 &= \alpha_{31} X_1 + \alpha_{32} X_2 + \alpha_{34} X_4
\end{align*}$$

Fig. 1. Example of a distributed storage graph
The dimension of the null-space of D is n minus the rank of D. Hence, it is evident that the dimension of the RDSS code is \( n - \text{minrank}_q(G) \).

From the above discussion, we have,

\[
\text{RDSS}_q(G) \geq n - \text{minrank}_q(G),
\]

and, \( n - \text{minrank}_q(G) \) is the maximum possible dimension of an RDSS code when the recovery functions are all linear. At this point, it is tempting to make the assertion \( \text{RDSS}_q(G) = n - \text{INDEX}_q(G) \), however, that would be wrong. This is shown in the following example.

This example is present in [1], and the distributed storage graph, a pentagon, is shown in Fig. 2. For this graph, a maximum-sized binary RDSS code consists of the codewords \{00000, 01100, 00011, 11011, 11011, 11101\}. The recovery functions are given by,

\[
\begin{align*}
X_1 &= X_2 \wedge X_5 \\
X_2 &= X_1 \vee X_3 \\
X_3 &= X_2 \wedge \bar{X}_4 \\
X_4 &= \bar{X}_3 \wedge X_5 \\
X_5 &= X_1 \vee X_4.
\end{align*}
\]

If all the recovery functions are linear, we could not have an RDSS code with so many codewords. Here \( \text{RDSS}_2(G) = \log_2 5 \). On the other hand, the minimum length of an index code for this graph is 3, i.e., \( \text{INDEX}_2(G) = 3 \), and this is achieved by the following linear mappings. The broadcaster transmit \( Y_1 = X_2 + X_3, Y_2 = X_4 + X_5 \) and \( Y_3 = X_1 + X_2 + X_3 + X_4 + X_5 \). The decoding functions are, \( X_1 = Y_1 + Y_2 + Y_3; X_2 = Y_1 + X_3; X_3 = Y_1 + X_2; X_4 = Y_2 + X_3; X_5 = Y_2 + X_4 \).

Although in general \( \text{RDSS}_q(G) \neq n - \text{INDEX}_q(G) \), these two quantities are not too far from each other. In particular, for large enough alphabet, the left and right hand sides can be arbitrarily close. This is reflected in Thm. 1 below.

It is to be noted that, we refrain from using ceiling and floor functions for clarity in this paper. In many cases, it is clear that the number of interest is not an integer and should be rounded off to the nearest larger or smaller integer. The main results do not change for this.

\section*{A. Implication of the results of [1]}

The result of [1] can be cast in our context in the following way.

**Theorem 1:** Given a graph \( G(V, E) \), we must have,

\[
\begin{align*}
\eta - \text{RDSS}_q(G) &\leq \text{INDEX}_q(G) \leq \eta - \text{RDSS}_q(G) \\
&+ \log_q \left( \min(\eta \ln q, 1 + \text{RDSS}_q(G) \ln q) \right).
\end{align*}
\]

This result is purely graph-theoretic, the way it was presented in [1]. In particular, the size of maximum independent set of the confusion graph, \( \gamma(G) \) was identified as the size of the RDSS code, and its relation to the chromatic number of the confusion graph, which represents the size of the index code was found. Namely the proof was dependent on the following two crucial steps.

1) The chromatic number of the graph can only be so much away from the fractional chromatic number (see, [1] for detailed definition).

2) The confusion graph is vertex transitive. This implies that the maximum size of an independent set is equal to the number of vertices divided by the fractional chromatic number.

A proof of the first fact above can be found in [10]. In what follows, we give a completely coding theoretic proof of this main theorem. We do not use the notion of the confusion graph or its vertex transitivity. Instead, we use some interesting properties of Hamming space for the proof.

\section*{III. A CODING PROOF OF THE DUALITY}

To prove Theorem 1, we use a lemma of coding theory, that was used to prove the famous Bassalygo-Elias bound on the size of error-correcting codes. This helps us completely bypass the notion of confusion graph. We prove Theorem 1 with the help of following two lemmas. The first of them is immediate.

**Lemma 2:** If there exists an index code \( \mathcal{C} \) of length \( \ell \) for a side information graph \( G(n, \ell) \), then there exists an RDSS code of dimension \( n - \ell \) for the distributed storage graph \( G \).

**Proof:** Suppose, the encoding and decoding functions of the index code \( \mathcal{C} \) are \( f: \mathbb{F}_q^n \rightarrow \mathbb{F}_q^\ell \) and \( g: \mathbb{F}_q^{\ell + N(\ell)} \rightarrow \mathbb{F}_q^n ; i = 1, \ldots, n \). There must exists some \( x \in \mathbb{F}_q^\ell \) such that \( |\{y \in \mathbb{F}_q^n : f(y) = x\}| \geq q^{n-\ell} \). Let, \( D_x = \{y \in \mathbb{F}_q^n : f(y) = x\} \) be the distributed storage code with recovery functions,

\[
f_i([X_j, j \in N(i)]) = g_i(x, [X_j, j \in N(i)]).
\]

The second lemma is the more interesting one.

**Lemma 3:** If there exists an RDSS code \( \mathcal{C} \) of dimension \( k \) for a distributed storage graph \( G(n, \ell) \), then there exists an index code of length \( n - k + \log_q \min(\eta \ln q, 1 + k \ln q) \) for the side information graph \( G \).

To prove this result, we need the help of a number of other lemmas. First of all notice that, translation of any RDSS code is an RDSS code.

**Lemma 4:** Suppose, \( \mathcal{C} \subseteq \mathbb{F}_q^n \) is an RDSS code. Then any known translation of \( \mathcal{C} \) is also an RDSS code of same
where \( \text{RDSS}(G) \) is an RDSS code of dimension \( \log_q n \) or \( n - \text{INDEX}(G) \).

To show the existence of good linear covering codes in \([5]\) (see also, \([4]\), The covering argument that we use here was used to show the existence of the entire covering by \( \mathcal{F} \) for any set \( \mathcal{F} \subseteq \mathbb{F}_q^n \).

\( F \) For any set \( \mathcal{F} \subseteq \mathbb{F}_q^n \), define \( F \) where \( \mathcal{F} \) is such that, \( \mathcal{F} \circ \mathcal{F} \) also use \( \mathcal{F} \) to show the existence of good linear covering codes in \([5]\) (see also, \([4]\), \([6]\), \([7]\), and was reintroduced in \([12]\) to show the existence of balancing sets.

**Lemma 5 (Bassalygo-Elias):** Suppose, \( \mathcal{C}, \mathcal{B} \subseteq \mathbb{F}_q^n \). Then, \( \sum_{x \in \mathbb{F}_q^n} |(\mathcal{C} + x) \cap \mathcal{B}| = |\mathcal{C}| |\mathcal{B}| \). \( \sum_{x \in \mathbb{F}_q^n} |(\mathcal{C} + x) \cap \mathcal{B}| = |(x, y) : x \in \mathbb{F}_q^n, y \in \mathcal{B}, y \in \mathcal{C} + x| \).

\[ \sum_{x \in \mathbb{F}_q^n} |(\mathcal{C} + x) \cap \mathcal{B}| = |(x, y) : x \in \mathbb{F}_q^n, y \in \mathcal{B}, y \in \mathcal{C} + x| \]
\[ = |(x, y) : x \in \mathbb{F}_q^n, y \in \mathcal{B}, x \in \mathcal{C} + y| \]
\[ = |(x, y) : y \in \mathcal{B}, x \in \mathcal{C} + y| \]
\[ = |\mathcal{B}| |\mathcal{C} + y| \]
\[ = |\mathcal{C}||\mathcal{B}| \]

In words, \( \mathcal{Q}(\mathcal{F}) \) denote the proportion of \( \mathbb{F}_q^n \) that is not covered by \( \mathcal{F} \). The following property is a result of Lemma 5.

**Lemma 6:** For every subset \( \mathcal{F} \subseteq \mathbb{F}_q^n \),
\[ q^{-n} \sum_{x \in \mathbb{F}_q^n} Q(\mathcal{F} \cup (\mathcal{F} + x)) = Q(\mathcal{F})^2. \] (8)

**Proof:** We have,
\[ |\mathcal{F} \cup (\mathcal{F} + x)| = 2|\mathcal{F}| - |\mathcal{F} \cap (\mathcal{F} + x)|. \]

Therefore,
\[ Q(\mathcal{F} \cup (\mathcal{F} + x)) = 1 - 2|\mathcal{F}|q^{-n} + |\mathcal{F} \cap (\mathcal{F} + x)|q^{-n}, \]
and hence,
\[ q^{-n} \sum_{x \in \mathbb{F}_q^n} Q(\mathcal{F} \cup (\mathcal{F} + x)) = 1 - 2|\mathcal{F}|q^{-n} \]
\[ + q^{-2n} \sum_{x \in \mathbb{F}_q^n} |\mathcal{F} \cap (\mathcal{F} + x)| \]
\[ = 1 - 2|\mathcal{F}|q^{-n} + q^{-2n} |\mathcal{F}|^2 \]
\[ = (1 - |\mathcal{F}|q^{-n})^2, \]

where in the second line we have used Lemma 5.

The implication of the above lemma is the following result.

**Lemma 7:** For every subset \( \mathcal{F} \subseteq \mathbb{F}_q^n \), there exists \( n = q^n|\mathcal{F}|^{-1} \min \{n \ln q, 1 + \ln |\mathcal{F}|\} \) vectors \( x_0 = 0, x_1, x_2, \ldots, x_{m-1} \in \mathbb{F}_q^n \), such that \( \bigcup_{i=0}^{m-1} (\mathcal{F} + x_i) = \mathbb{F}_q^n \).

**Proof:** From Lemma 6, for every subset \( \mathcal{F} \subseteq \mathbb{F}_q^n \), there exists \( x \in \mathbb{F}_q^n \) such that \( Q(\mathcal{F} \cup (\mathcal{F} + x)) \leq Q(\mathcal{F})^2 \).

For the set \( \mathcal{F} \equiv \mathcal{F}_0 \), recursively define, for \( i = 1, 2, \ldots \)
\[ \mathcal{F}_i = \mathcal{F}_{i-1} \cup (\mathcal{F}_{i-1} + z_{i-1}), \]
where \( z_i \in \mathbb{F}_q^n \) is such that,
\[ Q(\mathcal{F}_i \cup (\mathcal{F}_i + z_i)) \leq Q(\mathcal{F}_i)^2, \]
\[ i = 0, 1, \ldots \]

Clearly,
\[ Q(\mathcal{F}_i) \leq Q(\mathcal{F}_0)^{2^i} \leq (1 - q^{-n}|\mathcal{F}|)^{2^i} \leq e^{-q^{-n}|\mathcal{F}|2^i}. \]

If \( t \) is such that \( 2^t > \frac{n q^n \ln q}{|\mathcal{F}|} \), then
\[ Q(\mathcal{F}_t) < q^{-n}. \]

Now, from the definition of \( Q(\cdot) \) in (7), this implies \( |\mathcal{F}_t| = q^n \) or \( \mathcal{F}_t = \mathbb{F}_q^n \). Moreover, if \( t \) is such that \( 2^t > \frac{n q^n \ln q}{|\mathcal{F}|} \), then
\[ Q(\mathcal{F}_t) < \frac{1}{e|\mathcal{F}|}. \]

But, from (7), this implies,
\[ 1 - \frac{|\mathcal{F}|}{q^n} < \frac{1}{e|\mathcal{F}|}, \]
or,
\[ |\mathcal{F}|^2 - q^n |\mathcal{F}| + q^n/e > 0, \]
existence of $x_{i,j} = 1, \ldots, m - 1$ such that property (9) is satisfied.

Remark 1: Using our technique, it is possible to have a little improvement on Theorem 1. Namely, the right hand inequality can be written as,

$$\text{INDEX}_{q}(G) \leq n - \text{RDSS}_{q}(G) + \log_q \left( \min[n \ln q, \ln 2 + \text{RDSS}_{q}(G) \ln q] \right).$$

This improvement comes from our result of Lemma 7, where it suffices to have $Q(\mathcal{F}_t) < \frac{1}{2^t \overline{x}}$ instead of $Q(\mathcal{F}_t) < \frac{1}{e^t \overline{x}}$.

Remark 2: It is a bit surprising that the method of [1] (or, in fact the method of [10]) gives the exact same result as our method (barring the correction of Remark 1). It is worth investigating exactly how different (or similar) these methods are.

Acknowledgement

The author thanks Abhishek Agarwal for pointing out the results of [1] to the author. This work was supported in part by a grant from University of Minnesota.

REFERENCES