Linear Batch Codes

Helger Lipmaa and Vitaly Skachek

4-th Castle Meeting on Coding Theory and Applications

Castle of Palmela, Portugal

17 September 2014

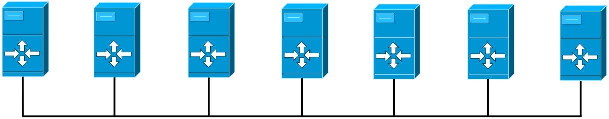
Supported by the research grants PUT405 and IUT2-1 from the Estonian Research Council, by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS, and by the COST Action IC1104 on random network coding and designs over $\mathbb{F}_q$.

Distributed storage systems

- Enormous amounts of data are stored in a huge number of servers.
- Occasionally servers fail.
- Failed server is replaced and the data has to be copied to the new server.

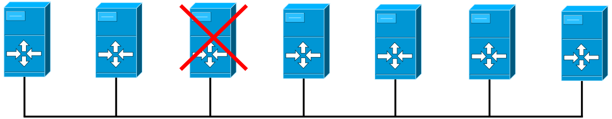
Distributed storage systems

- Enormous amounts of data are stored in a huge number of servers.
- Occasionally servers fail.
- Failed server is replaced and the data has to be copied to the new server.



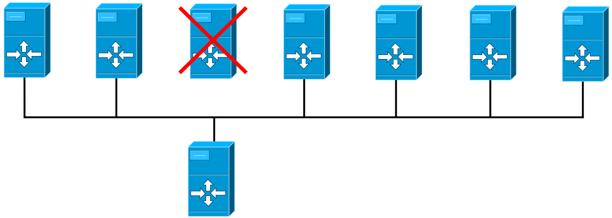
Distributed storage systems

- Enormous amounts of data are stored in a huge number of servers.
- Occasionally servers fail.
- Failed server is replaced and the data has to be copied to the new server.



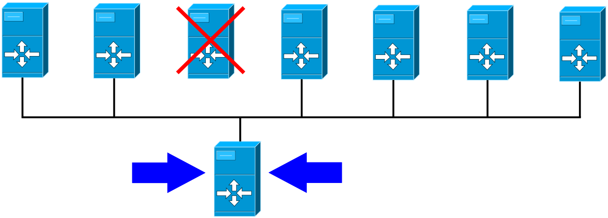
Distributed storage systems

- Enormous amounts of data are stored in a huge number of servers.
- Occasionally servers fail.
- Failed server is replaced and the data has to be copied to the new server.



Distributed storage systems

- Enormous amounts of data are stored in a huge number of servers.
- Occasionally servers fail.
- Failed server is replaced and the data has to be copied to the new server.



Locally repairable codes

- Consideration: minimize amount of transferred data.
- Proposed in [Dimakis, Godfrey, Wu, Wainwright, Ramchandran 2008].
- Error-correcting codes.
- Additional property: symbols can be corrected by using a small number of other symbols (locality).

Locally repairable codes

- Consideration: minimize amount of transferred data.
- Proposed in [Dimakis, Godfrey, Wu, Wainwright, Ramchandran 2008].
- Error-correcting codes.
- Additional property: symbols can be corrected by using a small number of other symbols (locality).

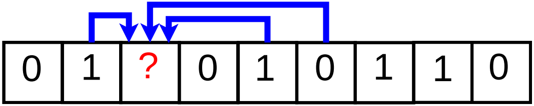
0	1	0	0	1	0	1	1	0

Locally repairable codes

- Consideration: minimize amount of transferred data.
- Proposed in [Dimakis, Godfrey, Wu, Wainwright, Ramchandran 2008].
- Error-correcting codes.
- Additional property: symbols can be corrected by using a small number of other symbols (locality).

0	1	?	0	1	0	1	1	0

- Consideration: minimize amount of transferred data.
- Proposed in [Dimakis, Godfrey, Wu, Wainwright, Ramchandran 2008].
- Error-correcting codes.
- Additional property: symbols can be corrected by using a small number of other symbols (locality).



Batch codes

- Proposed in [Ishai, Kushilevitz, Ostrovsky, Sahai 2004].
- Can be used in:
  - Load balancing.
  - Private information retrieval.
  - Distributed storage systems.

Batch codes

- Proposed in [Ishai, Kushilevitz, Ostrovsky, Sahai 2004].
- Can be used in:
  - Load balancing.
  - Private information retrieval.
  - Distributed storage systems.

Constructions:

- [Ishai *et al.* 2004]: algebraic, expander graphs, subsets, RM codes, locally-decodable codes

Prior art

Design-based constructions and bounds:

- [Stinson, Wei, Paterson 2009]
- [Brualdi, Kiernan, Meyer, Schroeder 2010]
- [Bujtas, Tuza 2011]
- [Bhattacharya, Ruj, Roy 2012]
- [Silberstein, Gal 2013]

Prior art

Design-based constructions and bounds:

- [Stinson, Wei, Paterson 2009]
- [Brualdi, Kiernan, Meyer, Schroeder 2010]
- [Bujtas, Tuza 2011]
- [Bhattacharya, Ruj, Roy 2012]
- [Silberstein, Gal 2013]

Application to distributed storage:

- [Rawat, Papailiopoulos, Dimakis, Vishwanath 2014]
- [Silberstein 2014]

Batch codes

Definition [Ishai *et al.* 2004]

$\mathcal{C}$ is an $(n, N, m, M, t)_\Sigma$ batch code over Σ if it encodes any string $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \Sigma^n$ into M strings (buckets) of total length N over Σ, namely $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_M$, such that for each m-tuple (batch) of (not neccessarily distinct) indices $i_1, i_2, \cdots, i_m \in [n]$, the symbols $x_{i_1}, x_{i_2}, \cdots, x_{i_m}$ can be retrieved by m users, respectively, by reading $\leq t$ symbols from each bucket, such that x_{i_ℓ} is recovered from the symbols read by the ℓ-th user alone.

Batch codes

Definition [Ishai *et al.* 2004]

$\mathcal{C}$ is an $(n, N, m, M, t)_\Sigma$ batch code over Σ if it encodes any string $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \Sigma^n$ into M strings (buckets) of total length N over Σ, namely $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_M$, such that for each m-tuple (batch) of (not neccessarily distinct) indices $i_1, i_2, \cdots, i_m \in [n]$, the symbols $x_{i_1}, x_{i_2}, \cdots, x_{i_m}$ can be retrieved by m users, respectively, by reading $\leq t$ symbols from each bucket, such that x_{i_ℓ} is recovered from the symbols read by the ℓ-th user alone.

Definition

If $t = 1$, then we use notation $(n, N, m, M)_\Sigma$ for it. Only one symbol is read from each bucket.

Batch codes

Definition [Ishai *et al.* 2004]

$\mathcal{C}$ is an $(n, N, m, M, t)_\Sigma$ batch code over Σ if it encodes any string $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \Sigma^n$ into M strings (buckets) of total length N over Σ, namely $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_M$, such that for each m-tuple (batch) of (not neccessarily distinct) indices $i_1, i_2, \cdots, i_m \in [n]$, the symbols $x_{i_1}, x_{i_2}, \cdots, x_{i_m}$ can be retrieved by m users, respectively, by reading $\leq t$ symbols from each bucket, such that x_{i_ℓ} is recovered from the symbols read by the ℓ-th user alone.

Definition

If $t = 1$, then we use notation $(n, N, m, M)_\Sigma$ for it. Only one symbol is read from each bucket.

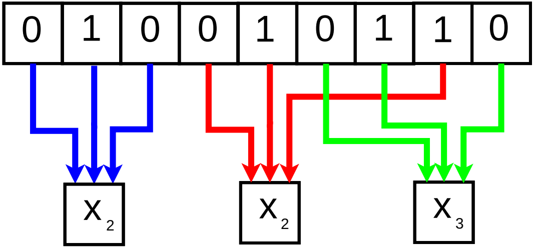
Definition

An $(n, N, m, M, t)_q$ batch code is *linear*, if every symbol in every bucket is a linear combination of original symbols.

Small buckets

In what follows, consider *linear codes* with $t = 1$ and $N = M$: each encoded bucket contains just one symbol in $\mathbb{F}_q$.

In what follows, consider *linear codes* with $t = 1$ and $N = M$: each encoded bucket contains just one symbol in $\mathbb{F}_q$.



For simplicity we refer to a linear $(n, N = M, m, M)_q$ batch code as $[M, n, m]_q$ batch code.

Linear batch codes

For simplicity we refer to a linear $(n, N = M, m, M)_q$ batch code as $[M, n, m]_q$ batch code.

- Let $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ be an information string.
- Let $\mathbf{y} = (y_1, y_2, \cdots, y_M)$ be an encoding of $\mathbf{x}$.
- Each encoded symbol y_i, $i \in [M]$, is written as $y_i = \sum_{j=1}^{n} g_{j,i} x_j$.
- Form the matrix $\mathbf{G}$:

$$\mathbf{G} = \left(g_{j,i} \right)_{j \in [n], i \in [M]} ;$$

the encoding is $\mathbf{y} = \mathbf{x}\mathbf{G}$.

Retrieval

Theorem

Let $\mathcal{C}$ be an $[M, n, m]_q$ batch code. It is possible to retrieve $x_{i_1}, x_{i_2}, \cdots, x_{i_m}$ simultaneously if and only if there exist m non-intersecting sets $T_1, T_2, \cdots, T_m$ of indices of columns in $\mathbf{G}$, and for T_r there exists a linear combination of columns of $\mathbf{G}$ indexed by that set, which equals to the column vector $\mathbf{e}_{i_r}^T$, for all $r \in [m]$.

Retrieval

Theorem

Let $\mathcal{C}$ be an $[M, n, m]_q$ batch code. It is possible to retrieve $x_{i_1}, x_{i_2}, \cdots, x_{i_m}$ simultaneously if and only if there exist m non-intersecting sets $T_1, T_2, \cdots, T_m$ of indices of columns in $\mathbf{G}$, and for T_r there exists a linear combination of columns of $\mathbf{G}$ indexed by that set, which equals to the column vector $\mathbf{e}_{i_r}^T$, for all $r \in [m]$.

Example

[Ishai *et al.* 2004] Consider the following linear binary batch code $\mathcal{C}$ whose 4×9 generator matrix is given by

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} .$$

Retrieval (cont.)

Example

Let $\mathbf{x} = (x_1, x_2, x_3, x_4)$, $\mathbf{y} = \mathbf{x}\mathbf{G}$.
Assume that we want to retrieve the values of (x_1, x_1, x_2, x_2). We can retrieve (x_1, x_1, x_2, x_2) from the following set of equations:

$$\begin{cases} x_1 &= y_1 \\ x_1 &= y_2 + y_3 \\ x_2 &= y_5 + y_8 \\ x_2 &= y_4 + y_6 + y_7 + y_9 \end{cases} .$$

It is straightforward to verify that any 4-tuple $(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4})$, where $i_1, i_2, i_3, i_4 \in [4]$, can be retrieved by using columns indexed by some four non-intersecting sets of indices in [9]. Therefore, the code $\mathcal{C}$ is a $[9, 4, 4]_2$ batch code.

Properties of linear batch codes

Lemma

Let $\mathcal{C}$ be an $[M, n, m]_q$ batch code. Then, the matrix $\mathbf{G}$ is full rank.

Properties of linear batch codes

Lemma

Let $\mathcal{C}$ be an $[M, n, m]_q$ batch code. Then, the matrix $\mathbf{G}$ is full rank.

Theorem

Let $\mathcal{C}$ be an $[M, n, m]_2$ batch code $\mathcal{C}$ over $\mathbb{F}_2$. Then, $\mathbf{G}$ is a generator matrix of the classical error-correcting $[M, n, \geq m]_2$ code.

Example

The converse is not true. For example, take $\mathbf{G}$ to be a generator matrix of the classical $[4,3,2]_2$ ECC as follows:

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} .$$

Let $\mathbf{x} = (x_1, x_2, x_3)$, $\mathbf{y} = (y_1, y_2, y_3, y_4) = \mathbf{xG}$.
It is impossible to retrieve (x_2, x_3). This can be verified by the fact that

$$x_2 = y_1 + y_2 = y_3 + y_4 \quad \text{and} \quad x_3 = y_1 + y_3 = y_2 + y_4 ,$$

and so one of the y_i's is always needed to compute each of x_2 and x_3.

Bounds on the parameters

- Various well-studied properties of linear ECCs, such as MacWilliams identities, apply also to linear batch codes (for $t = 1$, $M = N$ and $q = 2$).

Bounds on the parameters

- Various well-studied properties of linear ECCs, such as MacWilliams identities, apply also to linear batch codes (for $t = 1$, $M = N$ and $q = 2$).
- A variety of bounds on the parameters of ECCs, such as sphere-packing bound, Plotkin bound, Griesmer bound, Elias-Bassalygo bound, McEliece-Rodemich-Rumsey-Welch bound apply to the parameters of $[M, n, m]_2$ batch codes.

Construction 1

Theorem

Let $\mathcal{C}_1$ be an $[M_1, n, m_1]_q$ batch code and $\mathcal{C}_2$ be an $[M_2, n, m_2]_q$ batch code. Then, there exists an $[M_1 + M_2, n, m_1 + m_2]_q$ batch code.

Construction 1

Theorem

Let $\mathcal{C}_1$ be an $[M_1, n, m_1]_q$ batch code and $\mathcal{C}_2$ be an $[M_2, n, m_2]_q$ batch code. Then, there exists an $[M_1 + M_2, n, m_1 + m_2]_q$ batch code.

Let $\mathbf{G}_1$ and $\mathbf{G}_2$ be $n \times M_1$ and $n \times M_2$ generator matrices of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. Take $n \times (M_1 + M_2)$ matrix

$$\hat{\mathbf{G}} = [\ \mathbf{G}_1 \mid \mathbf{G}_2\ ]\ .$$

Construction 2

Theorem

Let $\mathcal{C}_1$ be an $[M_1, n_1, m_1]_q$ batch code and $\mathcal{C}_2$ be an $[M_2, n_2, m_2]_q$ batch code. Then, there exists an $[M_1 + M_2, n_1 + n_2, \min\{m_1, m_2\}]_q$ batch code.

Theorem

Let $\mathcal{C}_1$ be an $[M_1, n_1, m_1]_q$ batch code and $\mathcal{C}_2$ be an $[M_2, n_2, m_2]_q$ batch code. Then, there exists an $[M_1 + M_2, n_1 + n_2, \min\{m_1, m_2\}]_q$ batch code.

Denote by $\mathbf{G}_1$ and $\mathbf{G}_2$ the $n_1 \times M_1$ and $n_2 \times M_2$ generator matrices corresponding to $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. Take the following $(n_1 + n_2) \times (M_1 + M_2)$ matrix

$$\hat{\mathbf{G}} = \left[\begin{array}{c|c} \mathbf{G}_1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{G}_2 \end{array} \right] .$$

Theorem

Let $\mathcal{C}$ be an $[M, n, m]_q$ batch code, and let $\mathbf{G}$ be the corresponding $n \times M$ matrix. Then, the code $\hat{\mathcal{C}}$, defined by the $(n+1) \times (M+m)$ matrix

$$\hat{\mathbf{G}} = \left(\begin{array}{ccccc|cccc} & & \mathbf{G} & & & 0 & 0 & \cdots & 0 \\ & & & & & \vdots & \vdots & \ddots & \vdots \\ & & & & & 0 & 0 & \cdots & 0 \\ \hline \bullet & \bullet & \bullet & \cdots & \bullet & 1 & 1 & \cdots & 1 \end{array} \right)$$
$$\underbrace{\qquad\qquad\qquad\qquad}_{M}\underbrace{\qquad\qquad\qquad}_{m}$$

is an $[M+m, n+1, m]$ batch code, where $\bullet$ stands for an arbitrary symbol in $\mathbb{F}_q$.

Thank you!

Questions?