Constructions and Bounds for Batch Codes with Small Parameters

Eldho Thomas, Vitaly Skachek

31 August 2017

Institute of Computer Science
University of Tartu
Estonia

Batch Codes and PIR codes

Batch codes

Can potentially be used in:

- Load balancing in the distributed systems;
- Private information retrieval (PIR).

Ishai, Kushilevitz, Ostrovsky, Sahai, "Batch codes and their applications", STOC 2004

Batch Codes

Definition

An (n, k, t, r) *batch code* $\mathcal{C}$ *over a finite alphabet* $\mathcal{Q}$ *is defined by an encoding mapping* $\mathcal{C} : \mathcal{Q}^k \to \mathcal{Q}^n$*, and a decoding mapping* $\mathcal{D} : \mathcal{Q}^n \times [k]^t \to \mathcal{Q}^t$*, such that*

1. *For any* $\mathbf{x} \in \mathcal{Q}^k$ *and a multiset* $(i_1, i_2, \cdots, i_t) \subseteq [k]^t$*,*

$$\mathcal{D}\left(\mathbf{y} = \mathcal{C}(\mathbf{x}), i_1, i_2, \cdots, i_t\right) = \left(x_{i_1}, x_{i_2}, \cdots, x_{i_t}\right).$$

2. *The symbols in the query* $(x_{i_1}, x_{i_2}, \cdots, x_{i_t})$ *can be reconstructed from* t *respective disjoint recovery sets of symbols of* $\mathbf{y}$ *of size at most* r *each (the symbol* x_{i_ℓ} *is reconstructed from the* ℓ*-th recovery set for each* ℓ*,* $1 \leq \ell \leq t$*).*

Batch Codes

Definition

An (n, k, t, r) batch code $\mathcal{C}$ over a finite alphabet $\mathcal{Q}$ is defined by an encoding mapping $\mathcal{C} : \mathcal{Q}^k \to \mathcal{Q}^n$, and a decoding mapping $\mathcal{D} : \mathcal{Q}^n \times [k]^t \to \mathcal{Q}^t$, such that

1. *For any $\mathbf{x} \in \mathcal{Q}^k$ and a multiset $(i_1, i_2, \cdots, i_t) \subseteq [k]^t$,*

$$\mathcal{D}(\mathbf{y} = \mathcal{C}(\mathbf{x}), i_1, i_2, \cdots, i_t) = (x_{i_1}, x_{i_2}, \cdots, x_{i_t}) .$$

2. *The symbols in the query $(x_{i_1}, x_{i_2}, \cdots, x_{i_t})$ can be reconstructed from t respective disjoint recovery sets of symbols of $\mathbf{y}$ of size at most r each (the symbol x_{i_ℓ} is reconstructed from the ℓ-th recovery set for each ℓ, $1 \leq \ell \leq t$).*

If $\mathcal{Q}$ is a field and $\mathcal{C}$ is linear, then the code is a linear batch code.

Example 1

Consider the binary 2×3 generator matrix of a linear code $\mathcal{C}$ given as

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

The information symbols (x_1, x_2) encoded into $\mathbf{y} = (x_1, x_2, x_1 + x_2)$.

Take $t = 2$. For the query (x_1, x_2), the recovery sets are:

$$\begin{aligned} x_1 &= y_1 \\ x_2 &= y_2. \end{aligned}$$

For the query (x_1, x_1),

$$\begin{aligned} x_1 &= y_1 \\ x_1 &= y_2 + y_3. \end{aligned}$$

Conclusion: $\mathcal{C}$ is a batch code with $n = 3$, $k = 2$, $r = 2$, $t = 2$.

Example 2

Consider the following 3×7 generator matrix

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

The corresponding code is a binary $[7, 3, 4]$ classical error-correcting code.

Corresponding to the query (x_1, x_1, x_2, x_2), we have the following equation:

$$\begin{aligned} x_1 &= y_1 \\ x_1 &= y_2 + y_4 \\ x_2 &= y_3 + y_6 \\ x_2 &= y_5 + y_7 \end{aligned}$$

Similarly any 4-tuple $(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4})$ of queries can be retrieved using each symbol of $\mathbf{y}$ at most once.

Remark

All queries of length 4 can be retrieved by using at most $r = 2$ symbols from $\mathbf{y}$ for each recoverable x_i. Conclusion: $\mathcal{C}$ is a $(7, 3, 4, 2)$ batch code.

Unrestricted size of reconstruction sets

Remark

A variation of batch codes with unrestricted r *(say,* $r = n$*) is denoted as an* (n, k, t) *batch code (original definition of Ishai et al.)*

PIR Codes

The multi-server Private Information Retrieval (PIR) scenario:

- The database is stored in a number of servers in a distributed manner.
- The user is interested in reading an item from the database without revealing to any server what item was read.

A novel approach to PIR is based on coding.

Assume that $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ is an information vector, which is encoded into $C(\mathbf{x}) = \mathbf{y} = (y_1, y_2, \ldots, y_n)$.
The symbols of $\mathbf{y}$ are stored in different servers in a distributed manner.
Scenario: A user wants to know x_i without revealing i to the servers.

Fazeli, Vardy, and Yaakobi, PIR with low storage overhead: coding instead of replication, arXiv:1505.06241, 2015.

Definition

A $k \times n$ binary matrix G has property A_t if

- *for all $i \in [k]$, there exist t disjoint sets of columns of G that add up to e_i.*

A binary linear $[n, k]$ code $\mathcal{C}$ is called a PIR code if there exists a generator matrix G with property A_t.

Definition

A $k \times n$ binary matrix G has property A_t if

- *for all $i \in [k]$, there exist t disjoint sets of columns of G that add up to e_i.*

A binary linear $[n, k]$ code $\mathcal{C}$ is called a PIR code if there exists a generator matrix G with property A_t.

We define (n, k, t, r) and (n, k, t) PIR codes similarly to batch codes.

Batch codes are a special case of PIR codes:

Batch codes are a special case of PIR codes:

PIR codes support only queries of type $(x_i, x_i, \ldots, x_i), 1 \leq i \leq k$, while batch codes support queries $(x_{i_1}, x_{i_2}, \ldots, x_{i_t})$, where $i_1, i_2, \ldots, i_t$ can be different.

Known Bounds for PIR/Batch Codes

Restricted size of recovery sets

Theorem

Let $\mathcal{C}$ be a linear (n, k, t, r) batch code (or PIR code). Then, it holds:

$$n \geq k + d + (t-1)\left(\left\lceil \frac{k}{rt - t + 1} \right\rceil - 1\right) - 1.$$

Zhang and Skachek, Bounds for batch codes with restricted query size, ISIT, 2016.

Restricted size of recovery sets

Theorem

Let $\mathcal{C}$ be a linear (n, k, t, r) batch code (or PIR code). Then, it holds:

$$n \geq k + d + (t-1)\left(\left\lceil \frac{k}{rt - t + 1} \right\rceil - 1\right) - 1.$$

This is a refinement of the Singleton bound for these codes.

Zhang and Skachek, Bounds for batch codes with restricted query size, ISIT, 2016.

A batch code that supports query of size t, supports queries of size $\beta, 1 \leq \beta \leq t$.

A batch code that supports query of size t, supports queries of size $\beta, 1 \leq \beta \leq t$.

Corollary

$$n \geq k + d + \max_{1 \leq \beta \leq t} \left\{ (\beta - 1) \left(\left\lceil \frac{k}{r\beta - \beta + 1} \right\rceil - 1 \right) \right\} - 1.$$

A batch code that supports query of size t, supports queries of size $\beta, 1 \leq \beta \leq t$.

Corollary

$$n \geq k + d + \max_{1 \leq \beta \leq t} \left\{ (\beta - 1) \left(\left\lceil \frac{k}{r\beta - \beta + 1} \right\rceil - 1 \right) \right\} - 1.$$

When the code is systematic, the bound can be tighten a bit:

$$n \geq k + d + \max_{2 \leq \beta \leq t} \left\{ (\beta - 1) \left(\left\lceil \frac{k}{r\beta - \beta - r + 2} \right\rceil - 1 \right) \right\} - 1 \, .$$

Constructive New Bounds

Codes achieving these bounds

In this section we consider systematic codes only.

Codes achieving these bounds

In this section we consider systematic codes only. Take $r \geq 2$ and $t = 2$. Then the above bound becomes

$$n \geq \left\lceil \frac{k}{r} \right\rceil + k.$$

Codes achieving these bounds

In this section we consider systematic codes only. Take $r \geq 2$ and $t = 2$. Then the above bound becomes

$$n \geq \left\lceil \frac{k}{r} \right\rceil + k.$$

Consider

$$\mathbf{G} = \left(\begin{array}{ccccc|ccccc} \mathbf{I}_r & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_r & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_r & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I}_s & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} \end{array} \right),$$

where $s = k \mod r$, and recall that $\mathbf{1}$ denotes all-one column vector.

- Any query $t = 2$ is satisfied.

- Any query $t = 2$ is satisfied.
- When $r \mid k$, $n = \frac{k}{r}(r+1)$.

- Any query $t = 2$ is satisfied.
- When $r \mid k$, $n = \frac{k}{r}(r+1)$.
- When $r \nmid k$,

$$n = \left\lceil \frac{k}{r} \right\rceil + k \ .$$

Optimal length

Given k, r and t, denote:

Optimal length

Given k, r and t, denote:
$\mathcal{B}(k, r, t)$ - optimal n for a systematic (n, k, r, t) batch code;
$\mathcal{P}(k, r, t)$ - optimal n for a systematic (n, k, r, t) PIR code.

Optimal length

Given k, r and t, denote:
$\mathcal{B}(k, r, t)$ - optimal n for a systematic (n, k, r, t) batch code;
$\mathcal{P}(k, r, t)$ - optimal n for a systematic (n, k, r, t) PIR code.
$\mathcal{B}(k, r, t) \geq \mathcal{P}(k, r, t)$ implies:

Proposition

For any k, $t = 2$ and $r \geq 2$,

$$\mathcal{B}(k, r, t) = \mathcal{P}(k, r, t) = \left\lceil \frac{k}{r} \right\rceil + k \ .$$

Batch codes with $t \geq 2$ and $r = 2$

We construct batch codes using a generator matrix with columns of weight at most 2

$$\mathbf{G} = [\mathbf{I_k}|\mathbf{A}].$$

Batch codes with $t \geq 2$ and $r = 2$

We construct batch codes using a generator matrix with columns of weight at most 2

$$\mathbf{G} = [\mathbf{I_k}|\mathbf{A}].$$

A is constructed as follows:

Case 1: k even or $t-1$ even.

A has $wt(\mathbf{x}) = 2$ and $wt(\mathbf{y}) = t-1$, where **x** is any column and **y** is any row and no '1-square pattern' (to have disjoint recovery sets).

$$n = k + (t-1)k/2.$$

Batch codes with $t \geq 2$ and $r = 2$

We construct batch codes using a generator matrix with columns of weight at most 2

$$\mathbf{G} = [\mathbf{I_k}|\mathbf{A}].$$

A is constructed as follows:

Case 1: k even or $t-1$ even.

A has $wt(\mathbf{x}) = 2$ and $wt(\mathbf{y}) = t-1$, where **x** is any column and **y** is any row and no '1-square pattern' (to have disjoint recovery sets).

$$n = k + (t-1)k/2.$$

Case 2: k odd and $t-1$ odd.

All columns except last one of weight 2 and the last column of weight 1. For all **y**, $wt(\mathbf{y}) = t-1$, and there is no '1-square pattern'.

$$n = k + \left\lceil (t-1)\frac{k}{2} \right\rceil.$$

Example

Consider a binary (n, k, r, t) *batch (PIR) code* $\mathcal{C}$ *with* $k = 5$*,* $t = 3$ *and* $r = 2$*.*

The following GM generates a batch (PIR) code $\mathcal{C}$ *of length* 10

$$\mathbf{G} = \left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{array}\right).$$

Example

Take a binary (n, k, r, t) *batch (PIR) code* $\mathcal{C}$ *with* $k = 5$, $t = 4$ *and* $r = 2$.

A batch (PIR) code $\mathcal{C}$ *of length* 13 *with the above parameters is given by the generator matrix:*

$$\mathbf{G} = \left(\begin{array}{ccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) .$$

Proposition

For $t = 3$, the above code supports any query of the form (x_i, x_j, x_ℓ) with recovery sets of size at most 2, $i, j, \ell \in [k]$.

Proposition

For $t = 3$*, the above code supports any query of the form* (x_i, x_j, x_ℓ) *with recovery sets of size at most* 2*,* $i, j, \ell \in [k]$*.*

Proposition

For $t = 4$*, the above code supports any query of the form* (x_i, x_j, x_ℓ, x_h) *with recovery sets of size at most* 2*,* $i, j, \ell, h \in [k]$*.*

Proposition

For $t = 3$, the above code supports any query of the form (x_i, x_j, x_ℓ) with recovery sets of size at most 2, $i, j, \ell \in [k]$.

Proposition

For $t = 4$, the above code supports any query of the form (x_i, x_j, x_ℓ, x_h) with recovery sets of size at most 2, $i, j, \ell, h \in [k]$.

Proposition

For $t \geq 5$, the above code supports any query of size t of the form $(x_i, x_i, \cdots, x_i)$ with recovery sets of size at most 2.

Proposition

For $r = 2$ *and* $3 \leq t \leq \max\{\lceil \frac{k}{r} \rceil, r\} + 2$,

$$k + t - 2 + \left\lceil \frac{(t-1)(k-1)+1}{t} \right\rceil \leq \mathcal{P}(k, r, t) \leq k + \left\lceil (t-1) \cdot \frac{k}{2} \right\rceil .$$

Proposition

For $r = 2$ *and* $3 \leq t \leq \max\{\lceil \frac{k}{r} \rceil, r\} + 2$,

$$k + t - 2 + \left\lceil \frac{(t-1)(k-1)+1}{t} \right\rceil \leq \mathcal{P}(k, r, t) \leq k + \left\lceil (t-1) \cdot \frac{k}{2} \right\rceil .$$

The left-most inequality is obtained by substituting $r = 2$ into the result in [Zhang-Skachek '16].

Proposition

For $r = 2$ *and* $3 \leq t \leq \max\{\lceil \frac{k}{r} \rceil, r\} + 2$*,*

$$k+t-2+\left\lceil \frac{(t-1)(k-1)+1}{t} \right\rceil \leq \mathcal{P}(k,r,t) \leq k+\left\lceil (t-1)\cdot\frac{k}{2} \right\rceil .$$

The left-most inequality is obtained by substituting $r = 2$ into the result in [Zhang-Skachek '16].

Proposition

For $r = 2$ *and* $t \in \{3, 4\}$*,*

$$k+t-2+\left\lceil \frac{(t-1)(k-1)+1}{t} \right\rceil \leq \mathcal{B}(k,r,t) \leq k+\left\lceil (t-1)\cdot\frac{k}{2} \right\rceil .$$

Proposition

For $t = 3$ and $r \geq 3$,

$$k+1+\left\lceil \frac{2k-1}{2r-1} \right\rceil \leq \mathcal{B}(k,r,t) \leq \begin{cases} (r+1)\frac{k}{r} + \zeta \\ (r+1)\left\lfloor \frac{k}{r} \right\rfloor + 2s + 1 + \left\lceil \frac{(k-s)-\tau-\eta\cdot s}{\gamma} \right\rceil \end{cases}$$

where

$$\zeta = \max\left\{\frac{k}{r}, r\right\}, \quad s = k \mod r,$$

$$\tau \triangleq \min\left\{r-s, \left\lfloor \tfrac{k}{r} \right\rfloor\right\}, \eta \triangleq \min\left\{r-1, \left\lfloor \tfrac{k}{r} \right\rfloor\right\}, \gamma \triangleq \min\left\{r, \left\lfloor \tfrac{k}{r} \right\rfloor\right\}.$$

Example

Let $k = 12,\ r = 3,\ t = 3$ so that $k/r = 4$. The following generator matrix generates a batch code of length $n = 12 + 4 + 4 = 20$:

$$\mathbf{G} = \left(\begin{array}{cccc|cccc|cccc}
 & & & & & & & & 1 & 0 & 0 & 0 \\
\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 1 & 0 & 0 \\
 & & & & & & & & 0 & 0 & 1 & 0 \\
\hline
 & & & & & & & & 1 & 0 & 0 & 0 \\
\mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 & 0 & 1 & 0 \\
 & & & & & & & & 0 & 0 & 0 & 1 \\
\hline
 & & & & & & & & 1 & 0 & 0 & 0 \\
\mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & 0 & 1 & 0 & 0 \\
 & & & & & & & & 0 & 0 & 0 & 1 \\
\hline
 & & & & & & & & 0 & 1 & 0 & 0 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & 0 & 0 & 1 & 0 \\
 & & & & & & & & 0 & 0 & 0 & 1
\end{array}\right).$$

Example

Let $k = 11,\ r = 3,\ t = 3$ so that $\lfloor k/r \rfloor = 3$ and $s = 2$. The following generator matrix $\mathbf{G}$ generates a batch code of length $n = 19$:

$$
\mathbf{G} = \left(\begin{array}{cccc|ccc|c|cc|cc}
\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \begin{matrix}1\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\1\\0\end{matrix} & \begin{matrix}0\\0\\1\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} \\
\hline
\mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}1\\0\\0\end{matrix} & \begin{matrix}0\\1\\0\end{matrix} & \begin{matrix}0\\0\\1\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} \\
\hline
\mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}1\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\1\end{matrix} & \begin{matrix}0\\1\\0\end{matrix} \\
\hline
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \multicolumn{2}{c|}{\mathbf{I}_2} & \begin{matrix}0\\0\end{matrix} & \begin{matrix}0\\0\end{matrix}
\end{array}\right).
$$

Lower and upper bounds

	$t=2$	$t=3$	$t=4$
$r \geq 2$	$k+\lceil k/r \rceil$	$k+1+\left\lceil \frac{2k-1}{2r-1} \right\rceil$	$k+2+\left\lceil \frac{3k-2}{3r-2} \right\rceil$

Table 1: Lower bounds for $\mathcal{B}(k,r,t)$ and $\mathcal{P}(k,r,t)$

Lower and upper bounds

	$t=2$	$t=3$	$t=4$
$r \geq 2$	$k + \lceil k/r \rceil$	$k+1+\left\lceil \frac{2k-1}{2r-1} \right\rceil$	$k+2+\left\lceil \frac{3k-2}{3r-2} \right\rceil$

Table 1: Lower bounds for $\mathcal{B}(k,r,t)$ and $\mathcal{P}(k,r,t)$

	$t=2$	$t=3$	$t=4$
$r=2,\ k>3$	$k+\lceil k/2 \rceil$	$2k$	$k+\lceil 3k/2 \rceil$
$r \geq 3$	$k+\left\lceil \frac{k}{r} \right\rceil$	$\begin{cases} (r+1)\frac{k}{r}+\zeta & \text{if } r \mid k \\ (r+1)\left\lfloor \frac{k}{r} \right\rfloor + 2s + 1 + \left\lceil \frac{(k-s)-\tau-\eta\cdot s}{\gamma} \right\rceil & \text{if } r \nmid k \end{cases}$	$(r+1)\frac{k}{r}+2\zeta,\ \text{if } r \mid k$

Table 2: Upper bounds for $\mathcal{B}(k,r,t)$ and $\mathcal{P}(k,r,t)$

A new bound on the dimension

Denote $k_q^{\text{opt}}(n, d)$ - largest possible dimension of a linear code of length n and minimum distance d, for an alphabet of size q.

Cadambe, Mazumdar, "Bounds on the Size of Locally Recoverable Codes", IEEE Trans. Inform. Theory 2015

A new bound on the dimension

Denote $k_q^{\text{opt}}(n, d)$ - largest possible dimension of a linear code of length n and minimum distance d, for an alphabet of size q.

For a linear (n, k, r, t) batch code with $n - tr \geq d$,

Proposition

$$k \leq \min_{1 \leq \beta \leq t} \left\{ \beta r - (\beta - 1) + k_q^{\text{opt}}(n - \beta r, d) \right\}$$

Cadambe, Mazumdar, "Bounds on the Size of Locally Recoverable Codes", IEEE Trans. Inform. Theory 2015

A new bound on the dimension

Denote $k_q^{\text{opt}}(n, d)$ - largest possible dimension of a linear code of length n and minimum distance d, for an alphabet of size q.

For a linear (n, k, r, t) batch code with $n - tr \geq d$,

Proposition

$$k \leq \min_{1 \leq \beta \leq t} \left\{ \beta r - (\beta - 1) + k_q^{\text{opt}}(n - \beta r, d) \right\}$$

The new sphere-packing-based bound is tighter than the known Singleton-based bound [Zhang-Skachek '16] for very large n.

Cadambe, Mazumdar, "Bounds on the Size of Locally Recoverable Codes", IEEE Trans. Inform. Theory 2015

Open Questions

Open Questions

- Tighter bounds on the length or redundancy of batch and PIR codes, in particular, for small alphabet size, or for bounded values of r.

Open Questions

- Tighter bounds on the length or redundancy of batch and PIR codes, in particular, for small alphabet size, or for bounded values of r.
- Constructions of new optimal or sub-optimal batch and PIR codes.

Open Questions

- Tighter bounds on the length or redundancy of batch and PIR codes, in particular, for small alphabet size, or for bounded values of r.
- Constructions of new optimal or sub-optimal batch and PIR codes.
- Do non-linear batch (or PIR) codes have better parameters than their best linear counterparts?

Open Questions

- Tighter bounds on the length or redundancy of batch and PIR codes, in particular, for small alphabet size, or for bounded values of r.
- Constructions of new optimal or sub-optimal batch and PIR codes.
- Do non-linear batch (or PIR) codes have better parameters than their best linear counterparts?
- Do non-systematic batch (or PIR) codes have better parameters than their best systematic counterparts?

Open Questions

- Tighter bounds on the length or redundancy of batch and PIR codes, in particular, for small alphabet size, or for bounded values of r.
- Constructions of new optimal or sub-optimal batch and PIR codes.
- Do non-linear batch (or PIR) codes have better parameters than their best linear counterparts?
- Do non-systematic batch (or PIR) codes have better parameters than their best systematic counterparts?
- Large batch and PIR codes that allow for efficient reconstruction algorithms.

Thank you!