Batch codes based on lifted multiplicity codes

Rina Polyanskaya[*], and **Nikita Polyanskii**[†]
[*] Institute for Information Transmission Problems
Moscow, Russia 127051
[†] Skolkovo Institute of Science and Technology
Moscow, Russia 121205
Emails: rev-rina@yandex.ru, nikita.polyansky@gmail.com

***Abstract*—A binary k-batch code maps a binary string $\boldsymbol{x}$ of length n into a binary string $\boldsymbol{y}$ of length N, such that for every collection of k symbols from $\boldsymbol{x}$, there exist k mutually disjoint recovering sets from $\boldsymbol{y}$. We develop a new explicit coding construction of binary batch codes based on bivariate lifted multiplicity codes. For $\varepsilon \in (0, 27, 0.47)$ and $k = n^{\varepsilon}$, our proposed k-batch codes improve the redundancy of previously known ones.**

***Index Terms*—Disjoint recovering sets, batch codes, multiplicity codes, lifted codes**

I. Introduction

Ishai et al. [1] introduced a definition of batch codes in connection with load balancing problems in distributed storage systems. The original definition of *batch codes* was given in a very general form: n information symbols $x_1, \ldots, x_n$ are encoded to an m-tuple of strings $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ (called *buckets*) of total length N, such that for each k-tuple (*batch*) of distinct indices $i_1, \ldots, i_k \in [k]$, the entries $x_{i_1}, \ldots, x_{i_k}$ can be decoded by reading at most s symbols from each bucket. The parameter k is usually referred to as *availability*. If a batch could contain any *multiset* of indices, then we use the term a *multiset batch code*. In a special case when $s = 1$ and each bucket contains one symbol, a multiset batch code is called *primitive*. This class of batch codes is the most studied one in the literature since there are several statements [1] which allow to trade between different choices of k, N, n, m and s. In other words, better constructions of primitive batch codes would imply better constructions of multiset batch codes. Formally, we will use the following definition.

Definition 1. Let $\mathcal{C}$ be a code of length N and dimension n over an alphabet Σ of size q, which maps a q-ary string $x_1, \ldots, x_n$ to a q-ary string $y_1, \ldots, y_N$. The code $\mathcal{C}$ will be called a *primitive k-batch code* (simply, *k-batch code*), and will be denoted by $[N, n, k]_q^B$, if for every multiset of symbols $\{x_{i_1}, \ldots, x_{i_k}\}$, $i_j \in [n]$, there exist k mutually disjoint sets $R_1, \ldots, R_k \subset [N]$ (referred to as *recovering sets*) and functions $f_1, \ldots, f_k$ such that for all $\boldsymbol{c} \in \mathcal{C}$ and for all $j \in [k]$, $f_j(\boldsymbol{c}|_{R_j}) = c_{i_j}$, where $\boldsymbol{c}|_R$ is the projection of $\boldsymbol{c}$ onto coordinates indexed by R.

Reed-Solomon codes and Reed-Muller codes represent two classic families of algebraic error-correcting codes. Moreover, both are based on evaluations of polynomials: a codeword can be obtained by evaluating a polynomial over a finite field $\mathbb{F}_q$ of degree at most d at all points in $\mathbb{F}_q^m$. *Lifted codes* is a family of recently-introduced algebraic error-correcting codes based on evaluations of polynomials with the property that a polynomial restricted to a line in $\mathbb{F}_q^m$ is a low-degree polynomial. This family was originally proposed by Guo, Kopparty and Sudan [2] in order to get new ranges of parameters of codes with good local correction and testing properties. The most interesting result in such lifts is a construction of high-rate codes with sub-linear time decoding. Another construction with similar features, based on *multiplicity codes*, was presented by Kopparty, Saraf and Yekhanin in [3].

Very recently, Wu [4] and Li and Wootters [5] combined both ideas in *lifted multiplicity codes*, and showed that these codes exhibit nice locality properties. More precisely, Li and Wootters discussed codes with so-called *k-disjoint-repair-group property*, a notion of locality in error correcting codes. Informally, we may say that a code has the k-DRGP if any symbol of a codeword from the code can be obtained in k independent ways. It was shown that k-DRGP codes based lifted multiplicity codes have a more flexible construction than one based on just lifted or multiplicity codes.

Given n and k, we denote the minimal integer N such that an $[N, n, k]_q^B$ code exists by $N_B(n, k, q)$. In this paper we mainly focus on the minimal redundancy of batch codes, which we abbreviate by $r_B(n, k, q) := N_B(n, k, q) - n$. We will omit subscript q from the notation of the redundancy whenever this parameter is clear from the context. The goal of our paper is to use some results on lifted multiplicity codes to construct binary batch codes with good parameters.

A. Related work

The authors of [1] provided constructions of various families of batch codes. Those constructions are based on unbalanced expanders, on recursive application of trivial batch codes, on smooth and Reed-Muller codes, and others. Many other constructions proposed later in [6]–[9] improve the redundancy of batch codes. In particular, a systematic linear code, defined by the generator matrix $G = [I_n|E]$, is shown [7] to be a k-batch code, where t is the minimal number of ones in rows of E and the bipartite graph, whose biadjacency matrix is E, has no cycle of length at most 6. Constructions based on array codes and multiplicity codes were investigated in [6]. Very recently, the authors of [9] proposed a new batch code constructions based on some finite geometry frameworks.

978-1-7281-1944-1/19/$31.00 ©2019 IEEE

There is another class of related codes which is called *combinatorial batch codes*. For these codes, the same property as for batch codes is required, but symbols cannot be encoded. Such codes were investigated in [10]–[14]. A special case of batch codes, called *switch codes*, was studied in [15]–[18]. It was suggested in [15] to use such codes to increase the parallelism of data routing in the network switches. Also, we refer the reader to [19], where a definition of functional batch codes is introduced and some bounds on the redundancy of such codes are discussed.

Batch codes can be seen as an instance of *private information retrieval (PIR) codes*. For the latter we require a weaker property that every information symbol has k mutually independent recovering sets. PIR codes were suggested in [20] to decrease storage overhead in PIR schemes preserving both privacy and communication complexity. Some constructions and bounds for PIR codes can be found in [6], [20]–[23]. *One-step majority-logic decodable codes* [24] as well as *codes with the k-disjoint-repair-group property* [2], [5], [25] require a stronger property than PIR codes, namely every encoded symbol should have k mutually independent recovering sets. Also we refer the reader to *locally repairable codes with availability* [26]–[29], which have an additional (with respect to PIR codes) constraint on the size of recovering sets.

We use the notation n^{ε^-} in a statement to demonstrate that the statement remains true for all $n^{\varepsilon-c}$, where c is any fixed positive number. In the rest of the paper we will mainly concentrate on the case when $k = n^{\varepsilon}$, $n \to \infty$. Recall some known results on the minimal redundancy of (binary) batch codes:

1) $r_B(n,k) \geq k-1$;
2) $r_B(n,k) = \Omega(\sqrt{n})$ for linear batch codes and $k \geq 3$ (see [22], [30]);
3) $r_B(n,k) = \Theta(\sqrt{n})$ for linear batch codes and $3 \leq k \leq 5$ (see [6], [8]);
4) $r_B(n,k) = O(k^2\sqrt{n}\log n)$ for $k \leq \sqrt{n}/\log n$ (see [6]);
5) $r_B(n,n^{1/4}) \leq n^{7/8}$ (see [7]);
6) $r_B(n,n^{1/5}) \leq n^{4/5}$ (see [7]);
7) $r_B(n,n^{\varepsilon^-}) = O(n^{5/6+\varepsilon/3})$ for $0 < \varepsilon \leq 1/2$ (see [6]);
8) $r_B(n,n^{\varepsilon^-}) = O(n^{1/2+3\varepsilon/2})$ for $0 < \varepsilon \leq 1/3$ (see [9]);
9) $r_B(n,n^{\varepsilon}) = O(n^{1/2+3\varepsilon/2})$ for $\varepsilon = 1/(2\ell+1)$, $\ell \in \mathbb{N}$ (see [9]);
10) $r_B(n,n^{\varepsilon^-}) = O(n^{g(\varepsilon)})$ for $0 \leq \varepsilon \leq 1$ (see [6]), where

$$g(\varepsilon) := \min_{s\in\mathbb{N}: s > \frac{2}{1-\varepsilon}} \left[1 - \frac{s(1-\varepsilon)-2}{4s(s-1)}\right].$$

In particular, it follows that the best known lower bound on the redundancy of linear batch codes is as follows

$$r_B(n,k) \geq \Omega(\max(\sqrt{n}, k)). \tag{1}$$

B. Our contribution

This work extends the study of batch codes based on multiplicity codes [6]. The main result of our paper is a new explicit coding construction of binary k-batch codes of dimension n with $k = n^{\varepsilon}$, $\varepsilon \in (0, 1/2)$, utilized bivariate lifted multiplicity codes. In Theorem 3 we show that for $\varepsilon \in (0, 1/2)$, the optimal redundancy of binary batch codes satisfies

$$r(n, n^{\varepsilon-}) = O\left(n^{\log_4 3 + (2-\log_2 3)\varepsilon}\right).$$

We note that our main contribution hinges on results from the paper [5] on bivariate lifted multiplicity codes with application to codes with the disjoint-repair-group property.

Let us define $\delta := \delta(\varepsilon)$ to satisfy

$$\delta = \limsup_{n\to\infty} \log_n \left(r_B(n, n^{\varepsilon})\right).$$

The lower bound given by (1) along with old and new upper bounds on $\delta = \delta(\varepsilon)$ are plotted in Figure 1. The existence result of our work shows that the known upper bound on $\delta(\varepsilon)$ can be improved for $\varepsilon \in (0.27, 0.47)$.

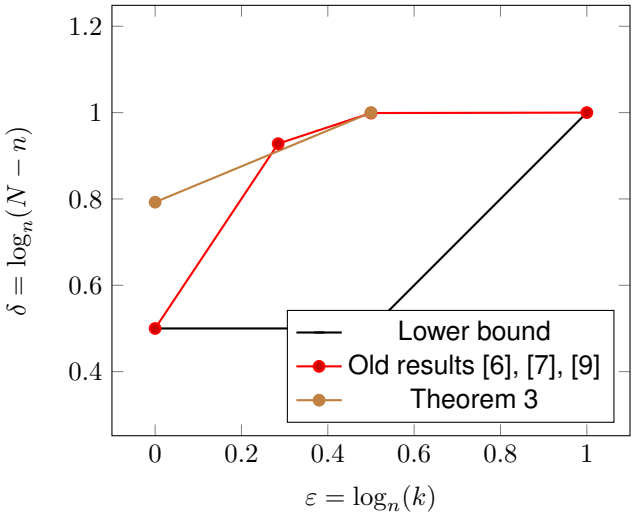

Fig. 1. Asymptotic results for binary primitive batch codes

C. Outline

The remainder of the paper is organized as follows. The bivariate lifted multiplicity codes are introduced in Section II. We show how to construct batch codes from lifted multiplicity codes in Section III. Finally, Section IV concludes the paper.

II. Bivariate Lifted Multiplicity Codes

In this section, we present some basic facts on (bivariate) polynomials and their derivatives. Also we give a definition of good polynomials we shall use for constructing lifted multiplicity codes. Finally we introduce (bivariate) lifted multiplicity codes and remark some properties of these codes.

A. Polynomials

In what follows, we use the finite field $\mathbb{F}_q$ of order q with characteristic 2, i.e., $q = 2^{\ell}$. Let $\mathbb{F}_q[x,y]$ be the ring of polynomials in the variables x, y with coefficients in $\mathbb{F}_q$. Let the degree of monomial $x^i y^j$ be $i+j$. For $P(x,y) \in \mathbb{F}_q[x,y]$, let the degree of $P(x,y)$ be the maximum degree over all monomials in $P(x,y)$.

B. Hasse derivatives

We will use the notion of Hasse derivatives defined below.

Definition 2. For $P(x,y) \in \mathbb{F}_q[x,y]$ and a vector (i,j) with $i,j \geq 0$, the (i,j)th *(Hasse) derivative* of P, denoted by $P^{(i,j)}(x,y)$, is the coefficient of $z^i w^j$ in the polynomial $Q(x,y,z,w) := P(x+z, y+w) \in F[x,y,z,w]$. Therefore, we have

$$Q(x,y,z,w) = \sum_{(i,j)} P^{(i,j)}(x,y) z^i w^j.$$

For $a,b \in \mathbb{F}_q$, integer $r \geq 1$ and $P(x,y) \in \mathbb{F}_q[x,y]$, we use the notation $P^{(<r)}(a,b) \in \mathbb{F}_q^{r+1}$ to denote the vector containing $P^{(i,j)}(a,b)$ for all $i,j \geq 0$ so that $i+j < r$.

We remark two well-known properties on Hasse derivates which will imply the linearity of (bivariate) lifted multiplicity codes over $\mathbb{F}_q$.

Proposition 1. *Let $P(x,y), Q(x,y) \in \mathbb{F}_q[x,y]$, $\lambda \in \mathbb{F}_q$ and let (i,j) be a vector of non-negative integers. Then we have*

1) $P^{(i,j)}(x,y) + Q^{(i,j)}(x,y) = (P+Q)^{(i,j)}(x,y)$.
2) $(\lambda P)^{(i,j)}(x,y) = \lambda P^{(i,j)}(x,y)$.

C. Polynomial equivalence

Definition 3. We say that two univariate polynomials $P(x), Q(x) \in \mathbb{F}_q[x]$ are equivalent up to order r if $P^{(i)}(a) = Q^{(i)}(a)$ for all $i = \{0, \ldots, r-1\}$ and $a \in \mathbb{F}_q$.

The following statement shows how small the degree of an equivalent polynomial could be.

Proposition 2 (Lemma 3.3 in [5])**.** *For every univariate polynomial $P(x)$, there exists a unique degree-at-most $rq-1$ polynomial $Q(x)$ such that $P(x)$ and $Q(x)$ are equivalent up to order r.*

D. Good polynomials

Let us define a set of lines, written as $\mathcal{L}_q$, we will use for constructing recovering sets

$$\mathcal{L}_q := \{(u, au+b)|_{u \in \mathbb{F}_q} : a,b \in \mathbb{F}_q\}.$$

We say that polynomial $P \in \mathbb{F}_q[x_1, x_2]$ is (d,r)-*good* if for any line $L = L(x)$ from the family $\mathcal{L}_q$, the univariate polynomial $P(L(x))$ is equivalent up to order r to the polynomial of degree at most d.

E. Span of good monomials

In fact, it is quite complicated to deal with the definition of good polynomials. To make our analysis simpler, we will work with good monomials only.

Let us consider the set $\mathcal{M}_q(d,r)$ consisting of (d,r)-good monomials $x^i y^j$ such that

$$\lfloor i/q \rfloor + \lfloor j/q \rfloor \leq r-1.$$

Define the family $\mathcal{F}_q(d,r)$ of polynomials spanned by the monomials from $\mathcal{M}_q(d,r)$.

F. Evaluation map

Given a positive integer r, let us define the evaluation map

$$e_r(P) : \mathbb{F}_q[x,y] \to \left(\mathbb{F}_q^{\binom{r+1}{2}}\right)^{q^2}$$

in the following manner

$$e_r(P) := \left(P^{(<r)}(a,b)\right)\Big|_{(a,b) \in \mathbb{F}_q^2}.$$

In other words, the map $e_r(P)$ evaluates the polynomial P together with all its derivatives up to order r in all the points in $\mathbb{F}_q^2$.

Proposition 3 (Follows from Lemma 3.5 in [5])**.** *The evaluation map*

$$e_r(P) : \mathcal{F}_q(d,r) \to \left(\mathbb{F}_q^{\binom{r+1}{2}}\right)^{q^2}$$

is an injection.

G. Bivariate lifted multiplicity codes and their properties

Now we are in a good position to give a key definition of bivariate lifted multiplicity codes.

Definition 4. The (q,r,d) bivariate lifted multiplicity code is a code $\mathcal{C}$ over alphabet $\Sigma = \mathbb{F}_q^{\binom{r+1}{2}}$ of length q^2 defined as follows

$$\mathcal{C} := \{e_r(P) : \quad P \in \mathcal{F}_q(d,r)\}.$$

The next statement gives a lower bound on the rate of the (q,r,d) bivariate lifted code.

Proposition 4 (Corollary 4.3 in [5])**.** *Let $r = 2^{\ell_r}$, $s = 2^{\ell_s}$ and $q = 2^\ell$ with $\ell_r, \ell_s \in \{1,2,\ldots,\ell\}$. The $(q, r, rq-s)$ lifted multiplicity code has rate at least $1 - 6r^{-1}s^{\log_2(4/3)}q^{\log_2(3/4)}$.*

A local recovering property of a symbol of a codeword from the lifted multiplicity code is presented by the following claim.

Proposition 5. *Suppose that $L_1, \ldots, L_r$ are r distinct lines from the family $\mathcal{L}_q$ all passing through a point $(a,b) \in \mathbb{F}_q^2$. Let $P(x,y)$ be from the family $\mathcal{F}_q(d,r)$. Then the value $P^{(<r)}(a,b)$ can be recovered if for each $j \in [r]$, we know values $P^{(<r)}(x,y)$ in at least $\lceil (d+1)/r \rceil$ distinct points on the line L_j.*

Proof of Proposition 5. For $j \in [r]$, let the line $L_j : \mathbb{F}_q \to \mathbb{F}_q^2$ be defined as follows: $L_j(x) = (x, \alpha_j x + \beta_j)$ such that $L_j(a) = (a,b)$. First we note that for any line $L \in \mathcal{L}_q$, the univariate polynomial $P(L(x))$ has degree at most d as P belongs to the family $\mathcal{F}_q(d,r)$. Second, since for each $j \in [r]$, we know at least $\lceil (d+1)/r \rceil$ distinct values $P^{(<r)}(\gamma,\delta)$ with (γ,δ) on the line L_j, we can easily find the values of the polynomial $f_j(x) := P(L_j(x))$ and its derivatives up to the rth order in at least $\lceil (d+1)/r \rceil$ points. Indead, for the line $L(x) = (x, \alpha x + \beta)$, the (i)th derivative of the univariate polynomial $f(x) := P(L(x))$ can be computed as follows

$$f^{(i)}(x) = \sum_{j=0}^{i} \alpha^j P^{(i-j,j)}(x, \alpha x + \beta).$$

Therefore, for each $j \in [r]$, using the Hermite interpolation we can find the polynomial $f_j(x)$ or a polynomial equivalent to $f_j(x)$ up to order r. In particular, this means that we can evaluate $f_j^{(<r)}(a)$. Finally, for any $i < r$, we can write the following system of equations

$$\begin{aligned} f_1^{(i)}(a) &= \sum_{j=0}^{i} \alpha_1^j P^{(i-j,j)}(a,b), \\ f_2^{(i)}(a) &= \sum_{j=0}^{i} \alpha_2^j P^{(i-j,j)}(a,b), \\ &\dots \\ f_{i+1}^{(i)}(a) &= \sum_{j=0}^{i} \alpha_i^j P^{(i-j,j)}(a,b). \end{aligned}$$

After denoting $u_1 := P^{(i,0)}(a,b)$, $u_2 := P^{(i-1,1)}(a,b)$, $\dots$, $u_{i+1} := P^{(0,i)}(a,b)$, $v_p := f_p^{(i)}(a)$ and $A_{p,j} := \alpha_p^{j-1}$ for $p, j \in [i+1]$, we can rewrite the system as follows

$$A\boldsymbol{u} = \boldsymbol{v}.$$

This system has a unique solution since A is a square Vandermonde matrix with different field elements in the second column. Thus, the unknown vector $\boldsymbol{u}$ can be found as $A^{-1}\boldsymbol{v}$. This implies that the values $P^{(i,j)}(a,b)$ can be recovered for any $i, j \geq 0$ such that $i+j < r$. This completes the proof. □

III. Batch Codes from Lifted Multiplicity Codes

In this section we will present a novel construction of binary (non-linear) batch codes. To this end, we first provide a construction of non-binary batch codes based on lifted multiplicity codes. After that, we compute the parameters of this construction in the asymptotic regime when the availability parameter $k = n^{\varepsilon}$. Finally, we show how to transform this construction into a binary batch code.

Theorem 1. *Fix integers $q = 2^{\ell}$, $r = 2^{\ell_r}$, $s = 2^{\ell_s}$ with $\ell_s, \ell_r \in \{1, \dots, \ell\}$. For all $k < \min((q-2)/r, (s-2r)/r^2)$ and $d = r(q - kr - 1) - 1$, the (q, r, d) bivariate lifted multiplicity code has the following properties:*

1) *The length of the code is q^2.*
2) *The rate of the code is at least*
$$1 - 6r^{-1}s^{\log_2(4/3)}q^{\log_2(3/4)}.$$
3) *The code is a k-batch code.*

Proof of Theorem 1. The inequality $k < (q-2)/r$ guarantees that d is positive. The first property follows from the Definition 4. The second property is implied by Proposition 4 and the fact that $s > kr^2 + 2r$ (in particular, it follows that the rate of the $(q, r, rq - s)$ lifted multiplicity code is smaller than the rate of the (q, r, d) lifted multiplicity code).

Let us focus on the last claim of the statement. Now we prove that every request of size k can be recovered. To this end, we shall use Proposition 5. The latter says that the value $P^{(<r)}(a,b)$ can be recovered if there exist r lines $L_1, \dots, L_r$ from the family $\mathcal{L}(q,2)$ and for each $j \in [r]$, we know at least $\lceil (d+1)/r \rceil$ distinct values $P^{(<r)}(\alpha,\beta)$ with (α,β) on the line L_j. The set of the points (α,β) will be called a recovering set for the point (a,b). Since two different lines in $\mathbb{F}_q^2$ can intersect in at most one point, one recovering set (consisting of the points on r lines) can intersect any other line in at most r points. Therefore, we get that every line, consisting of q points, has at most tr points which are used by t recovering sets. We note that every symbol $P^{(<r)}(a,b)$ of the lifted multiplicity code has at least $\lfloor q/r \rfloor$ distinct recovering sets since there are q lines from the family $\mathcal{L}_q$ passing through (a,b). It remains to check whether $\lceil (d+1)/r \rceil$ is at most $q - kr - 1$. However, this property follows from the fact that $d = r(q - kr - 1) - 1$. This completes the proof. □

In the next statement we show a connection between parameters of the non-binary batch code constructed in Theorem 1.

Theorem 2. *For any real c and α such that $c \geq 2$, $0 < \alpha < 1/c$, and n sufficiently large, there exists a k-batch code $[N, n, k]$ over the alphabet Σ such that the redundancy, $N - n$, the availability parameter, k, and the alphabet size, $Q = |\Sigma|$, satisfy*

$$\begin{aligned} N - n &= O(n^{1-((2c-3)-(c-2)\log_2 3)\alpha/2}), \\ k &= \Theta(n^{(1-c\alpha)/2}), \\ Q &= n^{O(n^{\alpha})}. \end{aligned}$$

Remark 1. Theorem 2 turns out to be a generalization of Theorem 21 in [6]. Indead, by plugging $c = 2$ we get exactly the statement from [6]. Recall that the construction in [6] is based on ordinary multiplicity codes. Using similar methods as utilized in Theorem 1, we can find a construction of batch codes based on ordinary lifted codes. However, the redundancy of the latter one appears to be less flexible as one presented in this paper.

Proof of Theorem 2. Let $q = 2^{\ell}$. We fix two real numbers $c \geq 2$ and α such that $0 < \alpha < 1/c$. Let r be the largest power of two such that $r \leq \lfloor q^{\alpha} \rfloor$. It follows that $r = \Theta(2^{\ell\alpha})$. We take $k = \lfloor q^{1-c\alpha}/4 \rfloor = \Theta(2^{\ell - c\alpha\ell})$. Then we set s to be the smallest power of two such that $s > kr^2 + 2r$. Therefore, we have $s = \Theta(kr^2) = \Theta(2^{(2-c)\ell\alpha + \ell})$.

We observe that for $c \geq 2$, we have $s \leq q$ and $r \leq q$. From Theorem 1 it follows that there exists a k-batch code over the alphabet $\mathbb{F}_q^{\binom{r+1}{2}}$ with length $N = q^2 = 2^{2\ell}$ and redundancy at most

$$\begin{aligned} 6Nr^{-1}s^{\log_2(4/3)}q^{\log_2(3/4)} &= O\left(2^{2\ell}2^{-\ell\alpha}(4/3)^{(2-c)\ell\alpha}\right) \\ &= O\left(4^{\ell + ((3/2-c)-(2-c)\log_4 3)\ell\alpha}\right) \\ &= O\left(N^{1-((2c-3)-(c-2)\log_2 3)\alpha/2}\right) \\ &= O\left(n^{1-((2c-3)-(c-2)\log_2 3)\alpha/2}\right). \end{aligned}$$

Let us rewrite k in terms of n

$$k = \Theta(2^{\ell - c\alpha\ell}) = \Theta(N^{(1-c\alpha)/2}) = \Theta(n^{(1-c\alpha)/2}).$$

The alphabet size $Q = |\Sigma|$ can be computed as follows

$$Q = \left|\mathbb{F}_q^{\binom{r+1}{2}}\right| = q^{\binom{r+1}{2}} = \Theta\left(n^{O(q^{2\alpha})}\right) = n^{O(n^{\alpha})}.$$

□

Theorem 3. *For any $0 < \varepsilon < 1/2$ and for any $\delta > 0$, there exists a binary $[N, n, n^{\varepsilon-\delta}]$ batch codes such that its redundancy satisfies*

$$N - n = O\left(n^{\log_4 3+(1-\log_2 3)\varepsilon}\right)$$

Proof of Theorem 3. Let $c \geq 2$ and $0 < \alpha < 1/c$. According to Theorem 2 there exists a k-batch code $\mathcal{C}$ of length N with dimension n over the alphabet of size Q such that

$$\begin{aligned} N - n &= O(n^{1-((2c-3)-(c-2)\log_2 3)\alpha/2}), \\ k &= \Theta(n^{(1-c\alpha)/2}), \\ Q &= n^{O(n^{\alpha})}. \end{aligned}$$

We construct the binary batch code $\mathcal{C}'$ from $\mathcal{C}$ by converting each symbol of the alphabet of size Q to $\log_2 Q = O(n^{\alpha}\log n)$ bits. Denote the length, dimension of the binary code by N', n' respectively. The redundancy of the code will be $r' = N' - n'$ and let k' be the availability parameter of the new code.

First, we note that $k' \geq k$. Indead, we know that each bit in $\mathcal{C}'$ is a bit among $\log Q$ bits representing some symbol in $\mathcal{C}$. For any recovering set for a symbol in $\mathcal{C}$, we have the corresponding recovering set for any bit from the image of this symbol in $\mathcal{C}'$.

Therefore, since $N = \Theta(n)$ we have

$$\begin{aligned} N' &= N\log_2 Q = \Theta\left(n^{\alpha+1}\log n\right), \\ n' &= n\log_2 Q = \Theta\left(n^{\alpha+1}\log n\right), \\ k' &= k = \Theta\left(n^{(1-c\alpha)/2}\right), \\ N' - n' &= (N-n)\log_2 Q \\ &= O\left(n^{\alpha+1-((2c-3)-(c-2)\log_2 3)\alpha/2}\log n\right). \end{aligned}$$

Therefore, we get that

$$n = \Theta\left((n'/\log(n'))^{1/(1+\alpha)}\right)$$

and

$$\begin{aligned} k' &= \Theta\left((n'/\log(n'))^{(1-c\alpha)/(2(1+\alpha))}\right) \\ &= \Theta\left((n'/\log(n'))^{1/2-(c+1)\alpha/(2(1+\alpha))}\right), \\ N' - n' &= O\left(n'^{1-((2c-3)-(c-2)\log_2 3)\alpha/(2(1+\alpha))}\log(n')\right). \end{aligned}$$

Let $\varepsilon = (c+1)\alpha/(2+2\alpha)$. Since $\alpha \in (0, 1/c)$, we obtain that $\varepsilon \in (0, 1/2)$. Then we can write

$$\begin{aligned} k' &= \Theta\left((n'/\log(n'))^{0.5-\varepsilon}\right), \\ N' - n' &= O\left(n'^{1-((2c-3)-(c-2)\log_2 3)\alpha'/(c+1)}\log(n')\right). \end{aligned}$$

Since the last inequality holds for any $c \geq 2$, we conclude that for any ε, $0 < \varepsilon < 1/2$, the optimal redundancy of binary batch codes satisfies

$$r(n, n^{\varepsilon^-}) = O\left(n^{\log_4 3+(2-\log_2(3))\varepsilon}\right).$$

□

IV. Conclusion

In this paper we discussed bivariate lifted multiplicity codes with an application to batch codes. We presented new coding constructions of batch codes over large alphabets and showed how to transform these constructions to binary batch codes. Our results improved the known upper bound on the redundancy of $[N, n, k]$ binary batch codes, when the availability parameter $k = n^{\varepsilon}$ and $\varepsilon \in (0.27, 0.47)$.

Acknowledgment

This research was supported by a grant from the Russian Science Foundation (grant no. 19-71-00137).

References

[1] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Batch codes and their applications," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. ACM, 2004, pp. 262–271.

[2] A. Guo, S. Kopparty, and M. Sudan, "New affine-invariant codes from lifting," in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. ACM, 2013, pp. 529–540.

[3] S. Kopparty, S. Saraf, and S. Yekhanin, "High-rate codes with sublinear-time decoding," *Journal of the ACM (JACM)*, vol. 61, no. 5, p. 28, 2014.

[4] L. Wu, "Revisiting the multiplicity codes: A new class of high-rate locally correctable codes," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 509–513.

[5] R. Li and M. Wootters, "Lifted multiplicity codes," *arXiv preprint arXiv:1905.02270*, 2019.

[6] H. Asi and E. Yaakobi, "Nearly optimal constructions of pir and batch codes," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 947–964, 2018.

[7] A. S. Rawat, Z. Song, A. G. Dimakis, and A. Gál, "Batch codes through dense graphs without short cycles," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1592–1604, 2016.

[8] A. Vardy and E. Yaakobi, "Constructions of batch codes with near-optimal redundancy," in *Information Theory (ISIT), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1197–1201.

[9] N. Polyanskii and I. Vorobyev, "Constructions of batch codes via finite geometry," in *Information Theory Proceedings (ISIT), 2019 IEEE International Symposium on*. IEEE, 2013, pp. 360–364.

[10] S. Bhattacharya, S. Ruj, and B. Roy, "Combinatorial batch codes: A lower bound and optimal constructions," *Advances in Mathematics of Communications*, vol. 6, no. 2, pp. 165–174, 2012.

[11] R. A. Brualdi, K. P. Kiernan, S. A. Meyer, and M. W. Schroeder, "Combinatorial batch codes and transversal matroids," *Advances in Mathematics of Communications*, vol. 4, no. 3, pp. 419–431, 2010.

[12] N. Silberstein and A. Gál, "Optimal combinatorial batch codes based on block designs," *Designs, Codes and Cryptography*, vol. 78, no. 2, pp. 409–424, 2016.

[13] D. Stinson, R. Wei, and M. B. Paterson, "Combinatorial batch codes," *Advances in Mathematics of Communications*, vol. 3, no. 1, pp. 13–27, 2009.

[14] C. Bujtás and Z. Tuza, "Combinatorial batch codes: Extremal problems under hall-type conditions," *Electronic Notes in Discrete Mathematics*, vol. 38, pp. 201–206, 2011.

[15] Z. Wang, O. Shaked, Y. Cassuto, and J. Bruck, "Codes for network switches," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 1057–1061.

[16] S. Buzaglo, Y. Cassuto, P. H. Siegel, and E. Yaakobi, "Consecutive switch codes," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2485–2498, 2018.

[17] Y. M. Chee, F. Gao, S. T. H. Teo, and H. Zhang, "Combinatorial systematic switch codes," in *Information Theory (ISIT), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 241–245.
[18] Z. Wang, H. M. Kiah, and Y. Cassuto, "Optimal binary switch codes with small query size," in *Information Theory (ISIT), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 636–640.
[19] Y. Zhang, T. Etzion *et al.*, "Bounds on the length of functional pir and batch codes," *arXiv preprint arXiv:1901.01605*, 2019.
[20] A. Fazeli, A. Vardy, and E. Yaakobi, "Pir with low storage overhead: coding instead of replication," *arXiv preprint arXiv:1505.06241*, 2015.
[21] S. R. Blackburn and T. Etzion, "Pir array codes with optimal pir rates," in *Information Theory (ISIT), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 2658–2662.
[22] S. Rao and A. Vardy, "Lower bound on the redundancy of pir codes," *arXiv preprint arXiv:1605.01869*, 2016.
[23] Y. Zhang, X. Wang, H. Wei, and G. Ge, "On private information retrieval array codes," *arXiv preprint arXiv:1609.09167*, 2016.
[24] S. Lin and D. J. Costello, *Error control coding*. Pearson Education India, 2001.
[25] S. L. Frank-Fischer, V. Guruswami, and M. Wootters, "Locality via partially lifted codes," *arXiv preprint arXiv:1704.08627*, 2017.
[26] A. Wang, Z. Zhang, and M. Liu, "Achieving arbitrary locality and availability in binary codes," in *Information Theory (ISIT), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 1866–1870.
[27] L. Pamies-Juarez, H. D. Hollmann, and F. Oggier, "Locally repairable codes with multiple repair alternatives," *arXiv preprint arXiv:1302.5518*, 2013.
[28] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," *IEEE Transactions on Information Theory*, vol. 62, no. 8, pp. 4481–4493, 2016.
[29] S. Kruglik, K. Nazirkhanova, and A. Frolov, "New bounds and generalizations of locally recoverable codes with availability," *IEEE Transactions on Information Theory*, 2019.
[30] M. Wootters, "Linear codes with disjoint repair groups," *Not intended for publication, available at https://sites.google.com/site/marywootters/disjoint_repair_groups.pdf*, 2016.