Constructions of Batch Codes via Finite Geometry

Nikita Polyanskii[*], and **Ilya Vorobyev**[*†]
[*]Center for Computational and Data-Intensive Science and Engineering,
Skolkovo Institute of Science and Technology
Moscow, Russia 121205
[†]Advanced Combinatorics and Complex Networks Lab,
Moscow Institute of Physics and Technology
Dolgoprudny, Russia 141701
Emails: nikita.polyansky@gmail.com, vorobyev.i.v@yandex.ru

***Abstract*—A primitive k-batch code encodes a string x of length n into string y of length N, such that each multiset of k symbols from x has k mutually disjoint recovering sets from y. We develop new explicit and random coding constructions of linear primitive batch codes based on finite geometry. In some parameter regimes, our proposed codes have lower redundancy than previously known batch codes.**

***Index Terms*—Private information retrieval, finite geometry, primitive batch codes**

A full version of this paper is accessible at [1].

I. Introduction

Batch codes were originally proposed by Ishai et al. [2] for load balancing in distributed systems, and amortizing the computational cost of private information retrieval and related cryptographic protocols. Ishai et al. gave a definition of *batch codes* in a general form, namely n information symbols $x_1, \dots, x_n$ are encoded to an m-tuple of strings $y_1, \dots, y_m$ (referred to as *buckets*) of total length N, such that for each k-tuple (*batch*) of distinct indices $i_1, \dots, i_k \in [n]$, the entries $x_{i_1}, \dots, x_{i_k}$ can be decoded by reading at most t symbols from each bucket. The parameter k is usually called *availability* and it plays an important role in supporting high throughput of the distributed storage system. If a batch could contain any *multiset* of indices (not only distinct indices), then we use the term a *multiset batch code*. In a special case when $t = 1$ 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 [2] which allow to trade between different choices of n, N, m, t and k. In other words, better constructions of primitive batch codes would imply better constructions of multiset batch codes.

A. Notation

We denote the field of size 2 by $\mathbb{F}_2$. The symbol $[n]$ stands for the set of integers $\{1, 2, \dots, n\}$. Let us give a formal definition of codes studied in this paper.

Definition 1. Let $\mathcal{C}$ be a linear code of length N and dimension n over the field $\mathbb{F}_2$, which encodes a string $x_1, \dots, x_n$ to $y_1, \dots, y_N$. The code $\mathcal{C}$ will be called a *primitive linear k-batch code* (simply, *k-batch code*), and will be denoted by $[N, n, k]^B$, if for every multiset of symbols $\{x_{i_1}, \dots, x_{i_k}\}$, $i_j \in [n]$, there exist k mutually disjoint sets $R_{i_1}, \dots, R_{i_k} \subset [N]$ (referred to as *recovering sets*) such that for all $j \in [k]$, x_{i_j} is a sum of the symbols y_p with indices p from R_{i_j}.

Given n and k, we denote the minimal integer N such that an $[N, n, k]^B$ code exists by $N_B(n, k)$. In this paper we focus on the minimal redundancy of batch codes, which we abbreviate by $r_B(n, k) := N_B(n, k) - n$.

Recall that a *systematic linear code* is a linear code in which the input data is embedded in the encoded output, i.e., $y_i = x_i$ for $i \in [n]$. In what follows we are going to construct systematic linear batch codes. The following special case of recovering sets will be particularly useful.

Definition 2. For a systematic linear code, we say that the recovering set R for information symbol x_i is *simple* if R contains exactly one index greater than n. In other words, if j is such an index, then

$$y_j = x_i + \sum_{t \in R \setminus \{j\}} x_t.$$

Note that many constructions, suggested earlier and in this paper, possess a more stronger property than one described in Definition 1 – the existence of mutually disjoint simple recovering sets. The only benefit of this additional restriction is that the analysis of such constructions is much easier.

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 $k = n^{\varepsilon}$, $n \to \infty$.

B. Related Work

The authors of [2] provided constructions of various families of batch codes. Those constructions were 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 [3]–[5] improve the redundancy of batch codes. In particular, a systematic linear code, defined by the generator matrix $G = [I_n | E]$, is shown [4] to be a k-batch code, where k 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 [3].

978-1-5386-9291-2/19/$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 the batch codes is required, but symbols cannot be encoded. Such codes were investigated in [6]–[10]. A special case of batch codes, called *switch codes*, was studied in [11]–[14]. It was suggested in [11] to use such codes to increase the parallelism of data routing in the network switches. Also, we refer the reader to [15], 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 [16] to decrease storage overhead in PIR schemes preserving both privacy and communication complexity. Some constructions and bounds for PIR codes can be found in [3], [16]–[19]. *One-step majority-logic decodable codes* [20] 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* [21]–[23], which have an additional (with respect to PIR codes) constraint on the size of recovering sets.

Recall some known results on the minimal redundancy of batch codes:

1) $r_B(n,k) \geq k-1$;
2) $r_B(n,k) = \Omega(\sqrt{n})$ for $k \geq 3$, [18], [24];
3) $r_B(n,k) = O(k^2\sqrt{n}\log n)$ for $k \leq \sqrt{n}/\log n$, [3];
4) $r_B(n,n^\varepsilon) \leq n^{7/8}$ for $7/32 < \varepsilon \leq 1/4$, [4];
5) $r_B(n,n^\varepsilon) \leq n^{4\varepsilon}$ for $1/5 < \varepsilon \leq 7/32$, [4];
6) $r_B(n,n^{\varepsilon^-}) \leq n^{5/6+\varepsilon/3}$ for $0 < \varepsilon \leq 1/2$, [3];
7) $r_B(n,n^{1-\varepsilon}) \leq n^{1-\delta}$ for $0 \leq \varepsilon \leq 1$, where $\delta = \delta(\varepsilon) > 0$, [3].

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

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

C. Our contribution

In this paper we develop new explicit and random coding constructions of linear primitive batch codes based on finite geometry. In Table I our contribution (upper bounds on $r_B(N,k)$) is summarized.

TABLE I
BINARY BATCH CODES SUMMARY

Construction	Availability k	Redundancy $r_B(n,k)$
Theorem 1 (random)	$k = o(n^{1/3}/\log n)$	$O(k^{3/2}\sqrt{n}\log n)$
Theorem 3 (explicit) for any $\ell \in \mathbb{N}$	$k < \frac{1}{\ell^2}n^{1/(2\ell+1)}$	$O\left(kn^{\frac{\ell+1}{2\ell+1}}\right)$

Let us denote $r_B(n,k=n^\varepsilon) =: O(n^\delta)$. 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 $\varepsilon \in (0,2/7)\setminus\{1/5,1/4\}$. Furthermore, we emphasize that the endpoints of novel explicit constructions by Theorem 3 lie on the segment given by the random construction in Theorem 1.

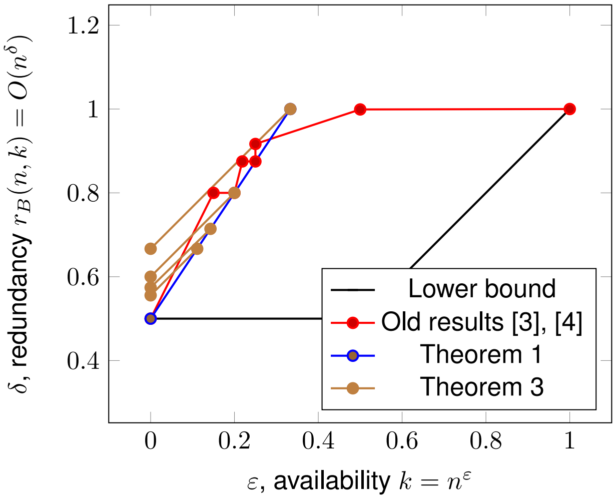

Fig. 1. Asymptotic results for binary primitive batch codes

D. Outline

The remainder of the paper is organized as follows. In Section II we prove the existence of batch codes using the probabilistic method. The achieved upper bound on the redundancy improves previously known results when $k = n^\varepsilon$ and $\varepsilon \in (0,2/7)\setminus\{1/5,1/4\}$. We note that for $k = n^{1/4}$ and $k = n^{1/5}$, the redundancy of our construction is worse by the multiplicative factor $\log n$ than one in [4]. In Section III we describe our main results and give new explicit constructions of batch codes. In a more detail, we associate information bits with elements of vector space $\mathbb{F}_q^{2\ell+1}$, $\ell \in \mathbb{N}$, and define parity-check bits as sums of information bits lying in some affine ℓ-dimensional subspaces. For $\ell \geq 2$, our explicit construction improves upon previously known constructions when $\varepsilon \in (1/(2(2\ell+1)), 1/(2\ell+1)]$. In particular, it gives the best currently known upper bound on the redundancy for $k = n^{1/(2\ell+1)}$. Finally, Section IV concludes the paper.

II. Random Construction of Batch Codes

To prove the following statement, we consider a systematic linear code defined by the generator matrix $G = [I_n|E]$, where E is taken as an incidence matrix of randomly chosen family of subsets of lines in the affine plane.

Theorem 1. *For $k = o(n^{1/3}/\log n)$, the redundancy of k-batch codes is*

$$r_B(n,k) = O(k^{3/2}\sqrt{n}\log n).$$

Proof. For simplicity of notation and without loss of generality, we assume that $n = q^2$, q is a prime power integer and $k < q/12$. Consider a finite affine plane (P,L) of order q, where P, $|P| = n$, is a set of points, and L, $|L| = n+q$, is a set of lines. Each line is known to contain q points, and each point is in $q+1$ lines, any two lines in the affine plane cross each other in at most 1 point.

Let us randomly choose a family $F := \{S_1, \ldots, S_M\}$ of sets of points that are subsets of lines in the affine space. First, we take each line in the affine space independently with probability p_1, which will be specified later. Second, we define a subset of any included line by leaving each point on the line independently with probability p_2, which will be specified later. It can be seen that for a proper choice p_1, the cardinality of F, $|F| = M$ (total number of subsets), is "close" to its average $p_1(n+q)$ with high probability, and for a proper choice of p_2, the cardinality of any subset S_i is "close" to its average $p_2 q$. We define event W_1 when the total number of lines $M > 3p_1 n$, and W_2 if there exists some S_i of size $> 2p_2 q$. Moreover, we define $W_{2,j}$, $j \in [n]$, if there exists S_i of size $> 2p_2 q$ such that the line corresponding to subset S_i does not contain the jth point.

Now we consider some bijection between n information symbols and n points. Therefore, the information symbols are associated with the points in the plane. Given a subset S_i, we can define a parity-check symbol y_i as a sum of information symbols corresponding to points in S_i. Let us consider a systematic linear code $\mathcal{C}$ of length $n+M$ and dimension n defined as a map $\phi : \mathbb{F}_2^n \to \mathbb{F}_2^{n+M}$:

$$\phi(x_1, \ldots, x_n) := (x_1, \ldots, x_n, y_1, \ldots, y_M).$$

Given a multiset of information symbols of size k, we can uniquely represent it in the form

$$((x_{i_1}, k_1), \ldots, (x_{i_\ell}, k_\ell)),$$

where

$$1 \le i_1 < \ldots < i_\ell \le n \text{ and } \sum_{i=1}^{\ell} k_i = k.$$

We define a greedy algorithm for constructing a collection of recovering sets for any given multiset of information bits of size at most k. This algorithm is similar to one in prior work [2], where Reed-Muller codes and smooth codes are used. Assume that the algorithm can construct simple recovering sets for the multiset

$$((x_{i_1}, k_1), \ldots, (x_{i_{j-1}}, k_{j-1})), \quad j-1 < \ell,$$

representing the first $j-1$ groups of the multiset

$$((x_{i_1}, k_1), \ldots, (x_{i_\ell}, k_\ell)).$$

Then find first k_j parity-check symbols depending on symbol x_{i_j}, such that the corresponding k_j simple recovering sets are disjoint with already chosen recovering sets, and k_j lines corresponding to the parity-check symbols does not go through any point in the set

$$\{x_{i_1}, \ldots, x_{i_{j-1}}, x_{i_{j+1}}, \ldots, x_{i_\ell}\}.$$

Let us add these k_j recovering sets to the collection of recovering sets. We note that added k_j simple recovering sets are mutually disjoint by our construction.

To show that the code $\mathcal{C}$ is likely to be a k-batch code, we are going to estimate the probability of event B that the greedy algorithm fails for some multiset of information symbols. To get an estimate of this event, we introduce auxiliary terminology. We say that the information symbol x_i is *s-bad*, $0 \le s < k$, if there exists some multiset

$$((x_{i_1}, k_1), \ldots, (x_{i_\ell}, k_\ell)) \text{ with } i = i_j,\ i_1 < \ldots < i_\ell,$$
$$\sum_{f \in [\ell] \setminus \{j\}} k_f = s,\ s + k_j = k,$$

so that the algorithm finds recovering sets for the first $(j-1)$ groups of the multiset and fails to find k_j recovering sets for $x_i = x_{i_j}$. Let $B_{i,s}$ be an event that information symbol x_i is s-bad. If no event among $B_{i,s}$ occurs, then the event B doesn't happen.

We note that k-batch code with redundancy at most $3p_1 n$ exists if $\Pr(B \cup W_1) < 1$. Now we estimate this event as follows

$$\begin{aligned} \Pr(B \cup W_1) &\le \Pr(W_1) + \Pr\left(\bigcup_{\substack{i \in [n] \\ s \in \{0,\ldots,k-1\}}} B_{i,s}\right) \\ &\le \Pr(W_1) + \Pr(W_2) + kn \max_{\substack{i \in [n] \\ s \in \{0,\ldots,k-1\}}} \Pr\left(B_{i,s} \cap \overline{W}_2\right). \end{aligned} \tag{2}$$

It is easy to estimate $\Pr(W_1)$ and $\Pr(W_2)$ applying the Chernoff bound in the form

$$\Pr(X \ge (1+\delta)\mu) \le e^{-\frac{\delta^2 \mu}{3}},$$

where $0 < \delta < 1$, and X is a sum of independent random variables taking values in $\{0,1\}$ with $\mathrm{E}\, X = \mu$. We have

$$\begin{aligned} \Pr(W_1) &= \Pr(M > 3p_1 n) \\ &\le \Pr(M > 2p_1(n+q)) \le e^{-\frac{p_1 n}{3}} \end{aligned} \tag{3}$$

and

$$\begin{aligned} \Pr(W_2) &= \Pr(\text{"there exists } S_i \text{ of size } > 2p_2 q\text{"}) \\ &\le 2ne^{-\frac{p_2 q}{3}}. \end{aligned} \tag{4}$$

Now we estimate the third probability in (2) as follows

$$\begin{aligned} &\Pr\left(B_{i,s} \cap \overline{W}_2\right) \le \Pr\left(B_{i,s} \cap \overline{W}_{2,i}\right) \\ &\le n^{k-1} \Pr\left(A \cap C \cap \overline{W}_{2,i}\right) \le n^{k-1} \Pr\left(A \mid \overline{W}_{2,i} \cap C\right), \end{aligned} \tag{5}$$

where C stands for the event that the algorithm finds recovering sets

$$R_{i_1,1}, \ldots, R_{i_1,k_{i_1}}, R_{i_2,1} \ldots, R_{i_{j-1},k_{i_{j-1}}}$$

for the first $j-1$ groups of

$$((x_{i_1}, k_1), \ldots, (x_{i_\ell}, k_{j_\ell})),$$

and A denotes the event that the algorithm fails to find $k-s$ recovering sets for x_i that are disjoint with all recovering sets the algorithm found. Note that the term n^{k-1} in (5) corresponds to the number of multisets having size k and containing element i. Let $I_1 := \{i_1, \dots, i_\ell\}$, and I_2 be a set of information symbols included in recovering sets

$$R_{i_1,1}, \dots, R_{i_1,k_{i_1}}, R_{i_2,1} \dots, R_{i_{j-1},k_{i_{j-1}}}.$$

The cardinality of I_2 given the event $\overline{W}_{2,i}$ (consequently, given the event $\overline{W}_{2,i} \cap C$) is upper bounded as follows

$$|I_2| = \sum_{u=1}^{j-1} \sum_{v=1}^{k_u} (|R_{i_u,v}| - 1) \le 2qp_2k, \tag{6}$$

since $\overline{W}_{2,i}$ stands for the event that all the subsets corresponding to the lines disjoint with x_i are of size at most $2p_2q$. The total number of lines containing x_i is equal to $q+1$. One can easily see that there are at most k of them which have a nonzero intersection with I_1. Since all the lines containing fixed point x_i share only x_i, we claim that there are at most $q/2$ lines which intersect I_2 by at least $4p_2k$ points. Indeed, otherwise we can lower bound the cardinality of I_2 by $\ge 4p_2k(q/2+1)$ which contradicts with (6). We shall try to recover symbol x_i with the help of other t, $t \ge q/2 - k \ge q/3$, lines. Enumerate them from 1 to t. Let $\xi_1, \dots, \xi_t$ be indicator random variables, which equals 1 iff

1) the corresponding line was randomly taken (with probability p_1),
2) the symbol x_i was left (with probability p_2) and included in the parity-check sum,
3) none of the symbols from I_2 were added in the corresponding parity-check.

Define the random variable

$$\eta := \sum_{i=1}^{q/3} \xi_i.$$

Since ξ_i is an independent Bernoulli random variable with probability $p'_i \ge p_1p_2(1-p_2)^{4p_2j}$, we claim that Binomial random variable χ with parameters $q/3$ and $p_1p_2(1-p_2)^{4p_2j}$ is stochastically dominated by η. Now we proceed with upper bounding (5) as follows

$$\begin{aligned}\Pr\left(A \mid \overline{W}_{2,i} \cap C\right) &\le \Pr\left(\chi < k - s\right) \\ &\le \binom{q/3}{k} \left(1 - p_1p_2(1-p_2)^{4p_2k}\right)^{q/3-k} \\ &\le q^k \left(1 - p_1p_2(1-p_2)^{4p_2k}\right)^{q/4}.\end{aligned}$$

Combining the last inequality together with (2)-(5) yields

$$\begin{aligned}&\Pr(B \cup W_1) \\ &\le 2ne^{-\frac{p_2q}{3}} + e^{-\frac{p_1n}{3}} + kn^kq^k(1 - p_1p_2(1-p_2)^{4p_2k})^{q/4}.\end{aligned} \tag{7}$$

Given $\varepsilon > 0$, there exists sufficiently large q_0 such that for $q > q_0$ the first two terms are at most ε. Now we proceed with the last term

$$kn^kq^k(1-p_1p_2(1-p_2)^{4p_2k})^{q/4} \le kn^{1.5k}e^{-p_1p_2(1-p_2)^{4p_2k}q/4}.$$

Taking $p_2 := 1/\sqrt{8k}$, we have $4p_2k \ge 1$ and

$$(1-p_2)^{4p_2k} \ge 1 - 4p_2^2k = 1/2.$$

From this it follows that for

$$p_1 := 36\frac{k^{3/2}\log n}{\sqrt{n}}$$

and sufficiently large n, $n = q^2$, the last term in (7) is at most ε. Therefore, we obtain that there exists a k-batch code with redundancy $M < 108k^{3/2}\sqrt{n}\log n$ with probability at least $1 - 3\varepsilon$. This completes the proof. □

III. Explicit Construction of Batch Codes

In this section to construct batch codes we associate information bits with elements of vector space $\mathbb{F}_q^{2\ell+1}$, $\ell \in \mathbb{N}$, and define parity-check bits as sums of information bits lying in some affine ℓ-dimensional subspaces. In particular, the following finite geometry framework turns out to be useful.

Definition 3. Suppose $\{V_1, \dots, V_m\}$ is a collection of ℓ-dimensional subspaces in $\mathbb{F}_q^{2\ell+1}$. This collection is said to be *L-nice* if the two properties hold:

1) any two distinct subspaces from this collection have the trivial intersection in the origin only, i.e. $|V_i \cap V_j| = 1$ for $i \neq j$;
2) for all $i \in [m]$ and for all $v \in \mathbb{F}_q^{2\ell+1}$, $v \notin V_i$, the affine subspace $v + V_i$ intersects at most L subspaces from this collection.

Such a framework appears to be new in the literature up to our best knowledge. In the following statement we show how to use a nice collection of subspaces to construct batch codes.

Lemma 2. *Suppose $\{V_1, \dots, V_m\}$ is an L-nice collection of ℓ-dimensional subspaces in $\mathbb{F}_q^{2\ell+1}$. Then there exists a $[q^{2\ell+1} + mq^{\ell+1}, q^{2\ell+1}, \lfloor m/L \rfloor]^B$ code.*

We give the proof of Lemma 2 in a full version of the present paper [1]. We emphasize that the property we used in the proof of Lemma 2 allows to construct recovering sets in an arbitrary order, that is a multiset request of information symbols could be given bit-by-bit and the corresponding recovering sets could be output bit-by-bit also. Now we give a construction of nice subspaces, which represents a collection of Reed-Solomon codes of length $2\ell+1$ and dimension ℓ.

Construction 1. Let V stand for a $(2\ell+1)$-dimensional $\mathbb{F}_q$-vector space, and B be an $\mathbb{F}_q$-basis for V. Now let us define a collection $\mathcal{C}$ of subspaces of size $m := \lfloor q/\ell \rfloor$. Let the ith, $0 \le i < m$, subspace $V_i \in \mathcal{C}$ be the linear span of ℓ vectors $\{v_1^i, \dots, v_\ell^i\}$, where vector v_j^i, $j \in \{0, \dots, \ell-1\}$, is written in basis B as follows

$$v_j^i := (1, \alpha^{\ell i+j}, \alpha^{2(\ell i+j)}, \dots, \alpha^{2\ell(\ell i+j)}).$$

We prove that $\mathcal{C}$ is ℓ-nice in Proposition 1. Let $m(L, \ell, q)$ be the maximal number m such that there exists an L-nice collection of ℓ-dimensional subspace in $\mathbb{F}_q^{2\ell+1}$ of cardinality m. The next two propositions establish a quite tight estimate on the maximal cardinality of a nice collection of subspaces.

Proposition 1. *Construction 1 is ℓ-nice. This implies, in particular, for any ℓ, $L \in \mathbb{N}$, $L \geq \ell$, and prime power integer q, the lower bound on $m(L, \ell, q)$ holds*

$$m(L, \ell, q) \geq \lfloor q/\ell \rfloor.$$

Proposition 2. *[25] For any ℓ, $L \in \mathbb{N}$ and prime power integer q, the upper bound on $m(L, \ell, q)$ holds*

$$m(L, \ell, q) \leq (L + 1)q.$$

The proof of Proposition 1 is given in [1].

Finally Lemma 2 and Proposition 1 imply the following upper bound on the redundancy of batch codes.

Theorem 3. *For any $\ell \in \mathbb{N}$, prime power integer q and integer k, $0 < k \leq \lfloor q/\ell^2 \rfloor$, the redundancy of k-batch codes is upper bounded by*

$$r_B(n, k) \leq \ell k q^{\ell+1},$$

where $n = q^{2\ell+1}$.

Remark 1. Proposition 2 verifies that the proposed framework based on finite geometry could not be significantly improved in terms of the range of parameter k in Theorem 3, that is k could not be larger than $\lfloor (L + 1)q/L \rfloor$.

Proof of Theorem 3. From Proposition 1 it follows that there exists an ℓ-nice collection of ℓ-dimensional subspaces in $\mathbb{F}_q^{2\ell+1}$, which has cardinality $\lfloor q/\ell \rfloor$. Take any subset of this collection of size $m = \ell k$, where $k \leq \lfloor q/\ell^2 \rfloor$. Lemma 2 states that there exists a $[q^{2\ell+1} + \ell k q^{\ell+1}, q^{2\ell+1}, k]^B$ code. This completes the proof. □

IV. Conclusion

In this paper new random coding bound and new explicit constructions of primitive linear batch codes based on finite geometry were developed. In some parameter regimes, our codes improves the redundancy than previously known batch codes. We note that the random coding bound coincides with the constructive bound in a countable number of points and gives better result in others. The natural open question arose in this work is to construct codes which would achieve random coding bound in all others points too. Another interesting question is how to improve the lower bound given by inequality (1).

Acknowledgment

We thank Eitan Yaakobi for the fruitful discussion on batch codes and Mary Wootters for the proof of Proposition 2. The research was conducted in part while the first author was a postdoctoral fellow at the Technion.

N. Polyanskii was supported in part the Russian Foundation for Basic Research (RFBR) through grant nos. 18-07-01427, 18-31-00310. I. Vorobyev was supported in part by RFBR through grant nos. 18-07-01427, 18-31-00361.

References

[1] N. Polyanskii and I. Vorobyev, "Constructions of batch codes via finite geometry," *arXiv preprint arXiv:1901.06741*, 2019.

[2] 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.

[3] H. Asi and E. Yaakobi, "Nearly optimal constructions of pir and batch codes," *IEEE Transactions on Information Theory*, 2018.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Y. Zhang, T. Etzion *et al.*, "Bounds on the length of functional pir and batch codes," *arXiv preprint arXiv:1901.01605*, 2019.

[16] A. Fazeli, A. Vardy, and E. Yaakobi, "Pir with low storage overhead: coding instead of replication," *arXiv preprint arXiv:1505.06241*, 2015.

[17] 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.

[18] S. Rao and A. Vardy, "Lower bound on the redundancy of pir codes," *arXiv preprint arXiv:1605.01869*, 2016.

[19] Y. Zhang, X. Wang, H. Wei, and G. Ge, "On private information retrieval array codes," *arXiv preprint arXiv:1609.09167*, 2016.

[20] S. Lin and D. J. Costello, *Error control coding*. Pearson Education India, 2001.

[21] 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.

[22] L. Pamies-Juarez, H. D. Hollmann, and F. Oggier, "Locally repairable codes with multiple repair alternatives," *arXiv preprint arXiv:1302.5518*, 2013.

[23] 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.

[24] M. Wootters, "Linear codes with disjoint repair groups," *personal communication*, 2016.

[25] ——, "Note on subspaces with nice properties," *personal communication*, 2018.