Batch and PIR Codes and Their Connections to Locally Repairable Codes

Vitaly Skachek

Abstract Two related families of codes are studied: batch codes and codes for private information retrieval. These two families can be viewed as natural generalizations of locally repairable codes, which were extensively studied in the context of coding for fault tolerance in distributed data storage systems. Bounds on the parameters of the codes, as well as basic constructions, are presented. Connections between different code families are discussed.

1 Introduction

In this survey, we discuss two related families of codes: batch codes and codes for private information retrieval (PIR codes). These two families can be viewed as natural generalizations of locally repairable codes, which were extensively studied in the context of coding for fault tolerance in distributed data storage systems.

Batch codes were first presented in [15], where it was suggested to use them for load balancing in the multi-server distributed data storage systems. It was also suggested in [15] to use these codes in private information retrieval. A number of constructions of batch codes were presented therein. Later, the authors of [33] proposed to use so-called "switch codes" for facilitating the routing of data in the network switches. It turns out, however, that switch codes are a special case of batch codes.

Coding schemes for PIR were studied in [12]. The authors showed that a family of codes, which is a relaxed version of batch codes, can be employed in classical linear PIR schemes in order to reduce the redundant information stored in a distributed server system. This relaxed version of the batch codes is termed PIR codes.

The author is with the Institute of Computer Science, University of Tartu, Tartu 50409, Estonia, e-mail: vitaly.skachek@ut.ee

In these schemes, typically a distributed data storage system is considered. The coded words are written across the block of disks (servers), where each disk stores a single symbol (or a group of symbols). The reading of data is done by accessing a small number of disks. Mathematically, this can be equivalently represented by the assumption that each information symbol depends on a small number of other symbols. However, the type of requested queries varies in different code models. Thus, in PIR codes several copies of the same information symbols are requested, while in batch codes combinations of different symbols are also possible.

In this survey, we mathematically define the corresponding families of codes, and study their properties. We derive bounds on the parameters of such codes, and show some basic constructions. We also show relations between different families of codes. In Section 3, we introduce various models of locally repairable codes. In Section 4, we define batch codes. In Section 5, we discuss properties of linear batch codes. In Section 6, we introduce codes for private information retrieval. In Section 7, we study connections between locally repairable and batch/PIR codes. In Sections 8 and 9, we present bounds on the parameters of various families of codes. In Section 10, we pose some open questions. For the sake of completeness, we introduce all necessary notations and definitions, which are used in the sequel.

2 General Settings

We denote by $\mathbb{N}$ the set of nonnegative integer numbers. For $n \in \mathbb{N}$, define $[n] \triangleq \{1, 2, \cdots, n\}$. Let $\mathbf{e}_i$ be the row vector having one at position i and zeros elsewhere (the length of vectors will be clear from the context).

Let Σ be a finite alphabet. Let $\mathbf{x} = (x_1, x_2, \cdots, x_k) \in \Sigma^k$ be an information vector. The code is a set of coded vectors

$$\left\{\mathbf{y} = (y_1, y_2, \cdots, y_n) = \mathcal{C}(\mathbf{x}) \ : \ \mathbf{x} \in \Sigma^k\right\} \subseteq \Sigma^n \ ,$$

where $\mathcal{C} : \Sigma^k \to \Sigma^n$ is a bijection, for some $n \in \mathbb{N}$. By slightly abusing the notation, sometimes we denote the above set by $\mathcal{C}$.

Let $\mathbb{F} = \mathbb{F}_q$ be a finite field with q elements, where q is a prime power. If $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^n$ is a linear mapping, then $\mathcal{C}$ is a linear $[n, k, d]$ code over $\mathbb{F}$. Here, d is the minimum Hamming distance of $\mathcal{C}$. In that case, the encoding can be viewed as a multiplication by a $k \times n$ generator matrix $\mathbf{G}$ over $\mathbb{F}$ of an information vector $\mathbf{x}$,

$$\mathbf{y} = \mathbf{x} \cdot \mathbf{G} \ . \tag{1}$$

The rate of the code is defined as $\mathcal{R} \triangleq k/n$.

3 Codes with Locality and Availability

Codes with locality were proposed for use in the distributed data storage systems [10]. In such systems, the data is stored in many disks (servers), and these servers may fail from time to time. It is possible to use erasure-correcting codes, where parts of a codeword are stored in different servers. A failure of one server can be viewed as an erasure of a symbol or of a group of symbols. In order to repair the erased symbols, there is a need to bring a few other symbols from other servers. In general, it would be beneficial to minimize the traffic in the network. We continue by recalling the definition of *locally repairable codes* (LRCs).

Definition 1. The code $\mathcal{C}$ has locality $r \geq 1$, if for any $\mathbf{y} \in \mathcal{C}$, any symbol in $\mathbf{y}$ can be recovered by using *at most* r other symbols of $\mathbf{y}$.

Codes with low locality were extensively discussed, for example, in [10]. A somewhat similar family of codes, known as *one-step majority-logic decodable codes*, was investigated in the classical literature [18]. Recently, the bounds on the parameters of LRCs were derived in [14]. It was shown therein that the parameters of a linear $[n, k, d]$ code with locality r over $\mathbb{F}$ satisfy:

$$n \geq k + d + \left\lceil \frac{k}{r} \right\rceil - 2 \, . \tag{2}$$

This bound can be viewed as a refinement of the classical Singleton bound, where $\lceil \frac{k}{r} \rceil - 1$ is an additive penalty for locality of the code, when compared to the classical Singleton bound. The proof is done by iterative expurgating of the code, and by taking into account that there are dependencies between sets of $r + 1$ symbols, which consist of an arbitrary symbol in $\mathbf{y}$ and its recovery set. The bound in (2) is tight. In fact, several known constructions attain it with equality (see, for example, [14, 27, 30]).

Assume that the linear code $\mathcal{C}$ is systematic, i.e. the matrix $\mathbf{G}$ contains a $k \times k$ identity submatrix $\mathbf{I}$. Then, the symbols of $\mathbf{y}$ corresponding to $\mathbf{I}$ are called *information symbols*. It is possible to require recoverability of information symbols only (from sets of size at most r). In that case, the code is said to have *locality of information symbols*. Otherwise, if *all* symbols of $\mathbf{y}$ are recoverable from small sets, the code has *locality of all symbols*.

The above model was extended to codes with *locality and availability* in [24].

Definition 2. The code $\mathcal{C}$ has locality $r \geq 1$ and availability $\delta \geq 1$, if for any $\mathbf{y} \in \mathcal{C}$, any symbol in $\mathbf{y}$ can be reconstructed by using any of δ disjoint sets of symbols, each set is of size at most r.

In [32], the authors consider linear codes with locality r and availability δ of all symbols. They derive the following bound on the parameters of the code:

$$n \geq k + d + \left\lceil \frac{\delta(k-1)+1}{\delta(r-1)+1} \right\rceil - 2 \, . \tag{3}$$

In [24], systematic codes (linear or non-linear) are considered, with locality r and availability δ of information symbols. The authors show the bound analogues to (3) for that case. In particular, we observe that when availability $\delta = 1$, i.e. there is only one recovery set for each symbol, then (3) coincides with (2). The proof technique in both cases is based on the idea similar to that of [14].

Another related model is considered in [23]. In that model, several *different* symbols are recovered from a small set of recovery symbols. By building on the ideas in the previous works, the authors derive a variation of the bound (3) for the model under consideration. Other related works, for example, include [6, 19, 21, 29, 35, 36] and the references therein.

4 Batch Codes

Batch codes were first presented in the cryptographic community in [15]. In that work, the authors proposed to use batch codes for load balancing in the distributed systems, as well as for private information retrieval. The authors of [15] have also presented a few constructions of various families of batch codes. Those constructions were based on recursive application of simple batch codes (so-called "sub-cube codes"), on classical Reed-Muller codes, on locally decodable codes, and others.

The following definition is based on [15].

Definition 3. Let Σ be a finite alphabet. We say that $\mathcal{C}$ is an $(k, n, t, M, \tau)_\Sigma$ *batch code* over a finite alphabet Σ if it encodes any string $\mathbf{x} = (x_1, x_2, \cdots, x_k) \in \Sigma^k$ 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 t-tuple (batch) of (not neccessarily distinct) indices $i_1, i_2, \cdots, i_t \in [k]$, the symbols $x_{i_1}, x_{i_2}, \cdots, x_{i_t}$ can be retrieved by reading at most τ symbols from each bucket.

More formally, by following the presentation in [31], we can state an equivalent definition.

Definition 4. An $(k, n, t, M, \tau)_\Sigma$ batch code $\mathcal{C}$ over a finite alphabet Σ is defined by an encoding mapping $\mathcal{C} : \Sigma^k \to (\Sigma^*)^M$ (there are M buckets in total), and a decoding mapping $\mathcal{D} : \Sigma^n \times [k]^t \to \Sigma^t$, such that

1. The total length of all buckets is n;
2. For any $\mathbf{x} \in \Sigma^k$ and

$$i_1, i_2, \cdots, i_t \in [k] \, , \tag{4}$$

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

and $\mathcal{D}$ depends only on τ symbols in each bucket in $\mathcal{C}(\mathbf{x})$.

In particular, an interesting case for consideration is when $\tau = 1$, namely only at most one symbol is read from each bucket. If the requested information symbols $(x_{i_1}, x_{i_2}, \cdots, x_{i_t})$ can be reconstructed from the data read by t different users independently (i.e., the symbol x_{i_ℓ} is reconstructed by the user ℓ, $\ell = 1, 2, \cdots, t$, respectively), and the sets of the symbols read by these t users are all disjoint, such a model is called a *multiset* batch code.

In the sequel, we only consider multiset batch codes, and therefore we usually omit the word "multiset" for convenience.

An important special case of batch codes is defined as follows.

Definition 5 ([15]). A *primitive* batch code is a batch code, where each bucket contains exactly one symbol. In particular, $n = M$.

Following the work [15], a number of subsequent papers have studied *combinatorial* batch codes. In combinatorial batch codes, a number of replicas of the information symbols are stored in different positions in the codeword. Usually, the symbols are associated with servers according to some optimal or sub-optimal combinatorial objects, such as block designs. Combinatorial batch codes were studied, for example, in [2, 4, 5, 26, 28].

5 Linear Batch Codes

In what follows, we consider a special case of primitive multiset batch codes with $n = M$ and $\tau = 1$. Under these conditions, each symbol can be viewed as a separate bucket, and only one reading per bucket is allowed.

We assume that the information and the coded symbols are taken from the finite field $\mathbb{F} = \mathbb{F}_q$, where q is a prime power. Additionally, we assume that the encoding mapping $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^n$ is linear over $\mathbb{F}$, and therefore the code $\mathcal{C}$ is a linear $[n, k]$ code over $\mathbb{F}$. In that case, $\mathcal{C}$ falls under the linear coding framework defined in (1). We also refer to the parameter t as the size of a query of the code. The batch code with the parameters n, k and t over $\mathbb{F}_q$ is denoted as $[n, k, t]_q$-batch code (or simply $[n, k, t]$-batch code) in the sequel.

This framework was first considered in [17], and the similarities with locally repairable codes were mentioned. The main difference between these two families, however, is that the supported query types are different. In batch codes we are interested in reconstruction of the information symbols in $\mathbf{x}$, while in locally repairable codes the coded symbols in $\mathbf{y}$ are to be recovered.

The following simple result was established in [17, Theorem 1].

Theorem 1. *Let $\mathcal{C}$ be an $[n, k, t]_q$ batch code. It is possible to retrieve $x_{i_1}, x_{i_2}, \cdots, x_{i_t}$ by t different users in the primitive multiset batch code model (where the symbol x_{i_ℓ} is retrieved by the user ℓ, $\ell = 1, 2, \cdots, t$, respectively)*

if and only if there exist t non-intersecting sets $T_1, T_2, \cdots, T_t$ of indices of columns in the generator matrix $\mathbf{G}$, and for each T_ℓ, $1 \leq \ell \leq t$, there exists a linear combination of columns of $\mathbf{G}$ indexed by that set, which equals to the column vector $\mathbf{e}_{i_\ell}^T$, for all $\ell \in [t]$.

The reader can find the proof of this theorem in [17]. Next, we show examples that further illustrate this concept.

Example 1 ([15]). Consider the following binary 2×3 generator matrix of a batch code $\mathcal{C}$ given as

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

The corresponding code is a sub-cube code constructed in [15, Section 3.2]. By using this code, the information symbols (x_1, x_2) are encoded into $(y_1, y_2, y_3) = (x_1, x_2, x_1 + x_2)$.

Assume that the query contains two different symbols (x_1, x_2). Then, we can retrieve these symbols direclty by using the following equations:

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

Alternatively, assume that the query contains two copies of the same symbol, for example (x_1, x_1). Then, we can retrieve these symbols by using the following equations:

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

Similarly, (x_2, x_2) can be retrieved. We conclude that $\mathcal{C}$ is a $[3, 2, 2]_2$ batch code.

Example 2 ([17]). Pick the following binary 4×9 generator matrix of a batch code $\mathcal{C}$ given as

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

The corresponding code is a second-order sub-cube code constructed as in [15, Section 3.2].

Assume that the query contains the information symbols (x_1, x_1, x_2, x_2). Then, we can retrieve these symbols using the following 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 can be verified in a similar manner 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 the symbols of $\mathbf{y}$, by using each symbol at most once. We conclude that $\mathcal{C}$ is a $[9, 4, 4]_2$ batch code.

Example 3 ([34]). Pick the following binary 3×7 generator matrix of a batch code $\mathcal{C}$ given as

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

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

Assume that the query contains the information symbols (x_1, x_1, x_2, x_2). Then, we can retrieve these symbols using the following equations:

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

It can be verified in a similar manner 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 the symbols of $\mathbf{y}$, by using each symbol at most once. We conclude that $\mathcal{C}$ is a $[7, 3, 4]_2$-batch code. Moreover, it was shown in [34] that all queries can be satisfied when each user reads at most $r = 2$ symbols from $\mathbf{y}$.

A family of codes, related to batch codes, and corresponding to the case $t = k$, termed *switch codes*, was studied in [8, 33, 34]. It was suggested in [33] to use such codes for efficient routing of data in the network switches.

The following property of batch codes was observed in [17] for binary linear codes, and later generalized to nonbinary (and also to non-linear) codes in [40].

Theorem 2. *Let $\mathcal{C}$ be an $[n, k, t]_q$-batch code. Then, the minimum Hamming distance of $\mathcal{C}$ is at least t.*

Proof. Let $\mathbf{y}_1 = \mathcal{C}(\mathbf{x}_1)$ and $\mathbf{y}_2 = \mathcal{C}(\mathbf{x}_2)$ be two codewords of $\mathcal{C}$, and $\mathbf{x}_1 \neq \mathbf{x}_2$. Then, $\mathbf{x}_1$ and $\mathbf{x}_2$ differ in at least one symbol, i.e. $(\mathbf{x}_1)_\ell \neq (\mathbf{x}_2)_\ell$, for some $1 \leq \ell \leq k$. Consider the query $(\underbrace{x_\ell, x_\ell, \cdots, x_\ell}_{t})$. The i-th copy of x_ℓ is recovered from the set of symbols indexed by the set T_i, $1 \leq i \leq t$. Since $\mathbf{x}_1$ and $\mathbf{x}_2$ differ in the ℓ-th symbol, the codewords $\mathbf{y}_1$ and $\mathbf{y}_2$ should differ in at least one symbol in each T_i, $1 \leq i \leq t$. The sets T_i are all disjoint, and therefore $\mathbf{y}_1$ and $\mathbf{y}_2$ differ in at least t symbols. □

It follows that any $[n, k, t]_q$-batch code is in particular a classical $[n, k, \geq t]_q$ error-correcting code, and a variety of classical bounds, such as Singleton bound, Hamming bound, Plotkin bound, Griesmer bound, Johnson bound,

Elias-Bassalygo bound, are all applicable to batch codes (when the minimum distance d is replaced by the query size t).

Example 4. Let $m > 1$ be an integer. A binary simplex $[2^m - 1, m, 2^{m-1}]$ code $\mathcal{C}$ is defined by its generator matrix

$$\mathbf{G} = \left(\mathbf{g}_1 \,|\, \mathbf{g}_2 \,|\, \cdots \,|\, \mathbf{g}_{2^m-1}\right) \ ,$$

where $\mathbf{g}_i$ are all possible different binary nonzero column vectors of length m, $i = 1, 2, \cdots, 2^m - 1$ [25, Problem 2.18].

For a classical error-correcting $[n, k, d]$ code over $\mathbb{F}_q$, the Plotkin bound is defined as follows [20, Theorem 2.2.29]:

$$\text{if } qd > (q-1)n, \quad \text{then} \quad q^k \leq \left\lfloor \frac{qd}{qd - (q-1)n} \right\rfloor .$$

It is straightforward to see that, as an error-correcting code, the binary simplex code as above (with $q = 2$) attains the Plotkin bound with equality [25, Problem 2.18] for all $m \geq 2$.

As it was shown in [34], the code $\mathcal{C}$ is a $[2^m - 1, m, 2^{m-1}]_2$ batch code, with $t = 2^{m-1}$. Therefore, by Theorem 2, it attains the corresponding Plotkin-based bound

$$q^k \leq \left\lfloor \frac{qt}{qt - (q-1)n} \right\rfloor$$

with equality, and therefore it is a Plotkin-optimal batch code.

In [38], a variation of batch codes *with restricted size of reconstruction sets* is defined. These codes are batch codes as in Definition 3 with an additional property that every queried information symbol x_i is reconstructed from *at most* $r \geq 1$ symbols of $\mathbf{y}$. This additional property can be viewed as analogous to locality of the LRCs. Small size of reconstruction sets allows for recovering the requested data symbol from a small number r of servers, thus reducing the traffic and the load in the system.

For example, the binary simplex code $\mathcal{C}$ in the previous example was shown in [34] to have the size of reconstruction sets of at most $r = 2$.

6 Codes for Private Information Retrieval

The topic of private information retrieval (PIR) protocols has been a subject of a lot of research over the last two decades [9]. In the 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. In the classical approach, the data is replicated, and the replicas are stored in a number of different servers. The user accesses some of these servers, such that no server learns what data the user is interested in (it is assumed that the servers do not collude).

A novel approach to PIR is based on coding, and it was studied, for example in [1, 7, 16]. More specifically, assume that $\mathbf{x} = (x_1, x_2, \cdots, x_k)$ is an information vector, which is encoded into $\mathcal{C}(\mathbf{x}) = \mathbf{y} = (y_1, y_2, \cdots, y_n)$. The symbols of $\mathbf{y}$ are stored in different servers in a distributed manner.

In [7], the authors show that there is a fundamental trade-off between download communication complexity of the protocol (the number of symbols or bits downloaded by the user from the database) and the storage overhead (the number of redundant symbols or bits stored in the database). Later, the authors of [12] show that it is possible to emulate many of the existing PIR protocols by using a code of length n that approaches $(1 + \epsilon)k$ for vanishing ϵ (for sufficiently large k). This approach leads to PIR schemes with storage data rate arbitrarily close to 1. The authors define a special class of codes, which allows for such efficient PIR protocols.

Definition 6. ([12]) An $k \times n$ binary matrix $\mathbf{G}$ has property $\mathcal{A}_t$ if for all $i \in [k]$, there exist t disjoint sets of columns of $\mathbf{G}$ that add up to $\mathbf{e}_i$. A binary linear $[n, k]$ code $\mathcal{C}$ is called a *t-server PIR code* (or, simply, PIR code) if there exists a generator matrix $\mathbf{G}$ for $\mathcal{C}$ with property $\mathcal{A}_t$.

The batch codes turn out to be a special case of PIR codes, with a difference that PIR codes support only queries of type $(x_i, x_i, \cdots, x_i)$, $i \in [k]$, while batch codes support queries of a more general form $(x_{i_1}, x_{i_2}, \cdots, x_{i_t})$, possibly for different indices $i_1, i_2, \cdots, i_t \in [k]$. It follows that batch codes can be used as PIR codes.

Since for PIR codes (as well as for batch codes), the code rate $\mathcal{R}$ approaches 1 [15, 22] for large values of k, it is more appropriate to talk about redundancy (as a function of k), rather than about the code rate. This is in contrast to PIR/batch codes with restricted size of reconstruction sets, where the asymptotic loss of code rate takes place. The redundancy of the codes will be defined and analyzed in the following sections.

Constructions of PIR codes were presented very recently, for example, in [3, 12, 39].

7 Connections Between Batch/PIR Codes and General LRCs

As it was mentioned above, there are two types of LRCs considered in the literature: LRCs with locality of information symbols and LRCs with locality of all symbols.

In order to preserve the information symbols in the coded word, a code with locality of information symbols has a systematic encoding matrix $\mathbf{G} = [\mathbf{I}|\mathbf{A}]$ for some matrix $\mathbf{A}$. Consider LRCs with locality r and availability $\delta = t - 1$ of information symbols. Then, each information symbol y_i, $1 \leq i \leq k$, in $\mathbf{y}$, can be recovered from δ disjoint sets T_i of symbols, $|T_i| \leq r$. Such a code can also be viewed as a PIR code that supports any query of t copies of an information symbol with locality r (including one copy of the information symbol in the systematic part). Generally, it does not follow that such a code is a batch code, since there is no guarantee that mixed queries of different information symbols are supported by disjoint reconstruction sets.

On the other hand, a systematic batch or PIR code with restricted size of reconstruction sets allows to recover any query of t information symbols with recovery sets of size r. Since in the systematic case, the information symbols are a part of a coded word $\mathbf{y}$, it follows that this code is an LRC with locality r and availability $\delta = t - 1$ of information symbols.

We obtain the following corollary (see also Theorem 21 in [12]).

Corollary 1. *A linear systematic code $\mathcal{C}$ is an LRC with locality r and availability $\delta = t - 1$ of information symbols if and only if $\mathcal{C}$ is a PIR code that supports queries of size t with size of reconstruction sets at most r.*

It follows that the bounds derived for the parameters of the LRCs with *locality and availability of information symbols* can be applied also to systematic batch or PIR codes.[1]

On the other hand, for the non-systematic case, there is no simple known connection between linear batch codes with restricted size of reconstruction sets and LRCs with availability, as it is illustrated in the following examples.

Example 5. Let $\mathbf{G}$ be a $k \times (3k)$ generator matrix of a linear binary code $\mathcal{C}$ defined as follows:

$$\mathbf{G} = \left(\begin{array}{c|c|c} 1\,0\,\ldots\,0\,1 & 1\,0\,\ldots\,0\,1 & 1\,0\,\ldots\,0\,1 \\ 0\,1\,\ldots\,0\,1 & 0\,1\,\ldots\,0\,1 & 0\,1\,\ldots\,0\,1 \\ 0\,0\,\ddots\,0\,1 & 0\,0\,\ddots\,0\,1 & 0\,0\,\ddots\,0\,1 \\ 0\,0\,\ldots\,1\,1 & 0\,0\,\ldots\,1\,1 & 0\,0\,\ldots\,1\,1 \\ 0\,0\,\ldots\,0\,1 & 0\,0\,\ldots\,0\,1 & 0\,0\,\ldots\,0\,1 \end{array}\right) .$$

Specifically, the binary information vector $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ is encoded into the codeword $\mathbf{y}$, which consists of three copies of the same sub-vector,

$$\begin{aligned} \mathbf{y} &= (y_1, y_2, \ldots, y_{3k}) \\ &= (x_1, x_2, \ldots, \sum_{i=1}^{k} x_i, \;\; x_1, x_2, \ldots, \sum_{i=1}^{k} x_i, \;\; x_1, x_2, \ldots, \sum_{i=1}^{k} x_i) \, . \end{aligned}$$

[1] Please note that generally it does not follow here that the bounds for LRCs with *locality of all symbols* are applicable to systematic batch or PIR codes.

The code $\mathcal{C}$, when viewed as an LRC, has locality $r = 1$ and availability $\delta = 2$, since every symbol in $\mathbf{y}$ can be recovered from a single symbol in $\mathbf{y}$, and there are 2 different recovery sets.

On the other hand, the code $\mathcal{C}$, when viewed as a batch or PIR code, must have a maximal size of reconstruction sets at least k, since it is impossible to recover a single information symbol x_k from less than k coded symbols in $\mathbf{y}$.

The following example shows that batch or PIR code with small size of reconstruction sets is not necessarily LRC with all symbols locality, even in the systematic case (the example can be also modified to a non-systematic case).

Example 6. Take $\mathbf{G}$ to be a binary $2\kappa \times (3\kappa + 1)$ matrix, where κ is some integer, as follows:

$$\mathbf{G} = \left(\begin{array}{ccc|ccc|c|ccc|c} 1&1&0&0&0&0&\dots&0&0&0&1\\ 0&1&1&0&0&0&\dots&0&0&0&1\\ \hline 0&0&0&1&1&0&\dots&0&0&0&1\\ 0&0&0&0&1&1&\dots&0&0&0&1\\ \hline \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots&\vdots\\ \hline 0&0&0&0&0&0&\dots&1&1&0&1\\ 0&0&0&0&0&0&\dots&0&1&1&1 \end{array}\right).$$

Here, $k = 2\kappa$. This matrix $\mathbf{G}$ is a diagonal block matrix, where each block is a 2×3 generator matrix of a basic sub-cube code in [15], with an additional all-ones column. Let $\mathcal{C}$ be a binary linear code generated by this matrix.

The code $\mathcal{C}$, when viewed as a batch code, supports any two queries of the form (x_i, x_j) $(1 \le i \le k,\ 1 \le j \le k)$, with size of reconstruction sets at most 2. Therefore, $\mathcal{C}$ is a batch code with $r = 2$, $t = 2$. In particular, it is a PIR code with $r = 2$ and $t = 2$.

On the other hand, the code $\mathcal{C}$, when viewed as an LRC, has locality of at least κ, since in order to recover $y_n = \sum_{i=1}^{k} x_k$, one needs to combine at least κ other symbols of $\mathbf{y}$.

As we see, this batch code with $r = 2$ and $t = 2$ has locality at least $k/2 = \kappa$, when used as an LRC with all symbols locality.

8 Bounds on the Parameters of PIR and Batch Codes with Unrestricted Size of Reconstruction Sets

In [31], the systematic batch codes with *unrestricted* size of reconstruction sets are considered. The codes under consideration have restriction on the value of t (typically, it is a small constant), yet there is no restriction on r, so we can assume that $r = n$.

Define the parameters $\mathcal{B}(k,t)$ and $\mathcal{P}(k,t)$ to be the shortest length n of any linear systematic batch and PIR code, respectively, for given values of k and t. Define the *optimal redundancy* of systematic batch and PIR codes, respectively, as

$$r_B(k,t) \triangleq \mathcal{B}(k,t) - k \quad \text{and} \quad r_P(k,t) \triangleq \mathcal{P}(k,t) - k \ .$$

It is known [31] that for any fixed t,

$$\lim_{k\to\infty} \frac{\mathcal{B}(k,t)}{k} = 1 \ .$$

In a case of switch codes, $k = t$, it is shown in [33] that $\mathcal{B}(k,k) = O\left(k^2/\log(k)\right)$. A constructive proof showing that $r_P(k,t) = t\cdot\sqrt{k}(1+o(1))$ was given in [12]. As for the lower bound on $r_P(k,t)$, it was recently shown in [22] (see also [37]) that for a fixed $t \geq 3$, $r_P(k,t) = \Omega(k)$, thus establishing an asymptotic behavior for the redundancy of the PIR codes.

Since every batch code is also a PIR code, it follows that $\mathcal{B}(k,t) \geq \mathcal{P}(k,t)$, and $r_B(k,t) \geq r_P(k,t)$. The relations between $\mathcal{B}(k,t)$ and $\mathcal{P}(k,t)$ for specific choices of t were extensively studied in [31]. Thus, for example, it was shown that $\mathcal{B}(k,t) = \mathcal{P}(k,t)$ for $1 \leq t \leq 4$, while for $5 \leq t \leq 7$,

$$r_B(k,t) \leq r_P(k,t) + 2\lceil\log(k)\rceil \cdot r_P(k/2, t-2) \ .$$

It is quite straightforward to verify that $r_B(k,1) = 0$ and $r_B(k,2) = 1$ for any k. It was additionally shown in [31] that

$$\begin{aligned} &r_B(k,t) = O(\sqrt{k}) \text{ for } t = 3,4 \ , \\ &r_B(k,t) = O(\sqrt{k}\cdot\log(k)) \text{ for } 5 \leq t \leq 7 \ . \end{aligned}$$

The following more general result was proven in [31].

Theorem 3. *For all values of k and t, it holds*

$$r_B(k,t) \leq r_P(k,t) + \left\lfloor \frac{t}{2} \right\rfloor \left\lceil \frac{\log\binom{k}{\lfloor t/2\rfloor}}{-\log\left(1 - \frac{\lfloor t/2\rfloor!}{\lfloor t/2\rfloor^{\lfloor t/2\rfloor}}\right)} \right\rceil \cdot r_P\left(\left\lceil \frac{k}{\lfloor t/2\rfloor} \right\rceil, t-2\right) \ .$$

In particular, it follows from Theorem 3 that for any fixed t,

$$r_B(k,t) = O\left(\sqrt{k}\cdot\log(k)\right) \ .$$

9 Bounds on the Parameters of PIR and Batch Codes with Restricted Size of Reconstruction Sets

In [38], the authors study linear batch codes with restricted size of reconstruction sets. They aim at refining the Singleton bound for that case by using ideas in [14] and subsequent works. Note, however, that these ideas cannot be applied directly, because in LRCs there are dependencies between different coded symbols, and expurgation of the code in the proof of the bound (2) (and similar bounds) uses these dependencies. Therefore, the authors of [38] consider a query of t copies of the same symbol (for example, $(x_1, x_1, \cdots, x_1)$), and show that the symbols in different reconstruction sets possess certain dependencies. By using this property, they apply an expurgation technique similar to that of [14, 23, 24, 32], and obtain the following relation on the parameters of batch codes with size of reconstruction sets restricted to r. The proof actually only assumes property $\mathcal{A}_t$, and therefore it is directly applicable to PIR codes as well.

Theorem 4 ([38]). *Let $\mathcal{C}$ be a linear $[n, k, t]_q$-batch code (or PIR code) with the size of reconstruction sets restricted to r. Then, it holds:*

$$n \geq k + d + (t-1)\left(\left\lceil \frac{k}{rt - t + 1} \right\rceil - 1\right) - 1 \,. \tag{5}$$

Now, observe that if the $[n, k, t]_q$-batch code (or PIR code) allows for reconstruction of any batch of t symbols, then it also allows for reconstruction of any batch of β symbols, $1 \leq \beta \leq t$. Therefore, expression (5) in Theorem 4 can be adjusted as follows:

$$n \geq k + d + \max_{1 \leq \beta \leq t, \beta \in \mathbb{N}} \left\{ (\beta - 1)\left(\left\lceil \frac{k}{r\beta - \beta + 1} \right\rceil - 1\right)\right\} - 1 \,. \tag{6}$$

If the code is systematic, then there is always a reconstruction set of size 1 for one of the queried symbols. In that case, the last expression can be rewritten as:

$$n \geq k + d + \max_{2 \leq \beta \leq t, \beta \in \mathbb{N}} \left\{ (\beta - 1)\left(\left\lceil \frac{k}{r\beta - \beta - r + 2} \right\rceil - 1\right)\right\} - 1 \,. \tag{7}$$

The reader can refer to [38] for the full proofs.

Example 7 ([38]). Take $r = 2$ and $t = \beta = 2$. Then, the bound in (7) is attained with equality by the linear systematic codes of minimum distance 2, defined as follows:

- $y_i = x_i$ for $1 \leq i \leq k$, and $y_j = x_{2(j-k)-1} + x_{2(j-k)}$ for $k+1 \leq j \leq k + k/2$, when k is even,

- $y_i = x_i$ for $1 \leq i \leq k$, $y_j = x_{2(j-k)-1} + x_{2(j-k)}$ for $k+1 \leq j \leq k+(k-1)/2$, and $y_{k+(k+1)/2} = x_k$, when k is odd.

In that case, $d = 2$, and we obtain

$$\begin{aligned} n &= k + k/2 \qquad \text{if } k \text{ is even} \ , \\ n &= k + (k+1)/2 \qquad \text{if } k \text{ is odd} \ . \end{aligned}$$

In both cases, the bound (7) is attained with equality for all $k \geq 1$.

Example 8. Consider the code $\mathcal{C}$ in Example 3, which was studied in [34]. As discussed, $\mathcal{C}$ is a linear $[7, 3, 4]_2$-batch code, with the size of reconstruction sets at most $r = 2$. Its minimum Hamming distance is $d = 4$. We pick $\beta = 2$, and observe that the right-hand side of equation (7) can be re-written as

$$3 + 4 + (2-1)\left(\left\lceil \frac{3}{2 \cdot 2 - 2 - 2 + 2} \right\rceil - 1\right) - 1 = 7 \ ,$$

and therefore the bound in (7) is attained with equality for the choice $\beta = 2$. The code $\mathcal{C}$ in Example 3 is optimal with respect to that bound. We note, however, that general simplex codes (of larger length) do not attain (7) with equality.

A slight improvement to the above bounds for both batch and PIR codes can be obtained, if one considers simultaneously reconstruction sets for, say, two queried batches $(x_1, x_1, \ldots, x_1)$ and $(x_2, x_2, \ldots, x_2)$, and studies intersections of their reconstruction sets. The analysis along those lines was done in [38], and the following result was derived.

Assume that

$$k \geq 2(rt - t + 1) + 1 \ . \tag{8}$$

Denote

$$\begin{aligned} \mathbb{A} = \mathbb{A}(k, r, d, \beta, \epsilon) &\triangleq k + d + (\beta - 1)\left(\left\lceil \frac{k+\epsilon}{r\beta - \beta + 1} \right\rceil - 1\right) - 1 \ , \\ \mathbb{B} = \mathbb{B}(k, r, d, \beta, \lambda) &\triangleq k + d + (\beta - 1)\left(\left\lceil \frac{k+\lambda}{r\beta - \beta + 1} \right\rceil - 1\right) - 1 \ , \\ \mathbb{C} = \mathbb{C}(k, r, \beta, \lambda, \epsilon) &\triangleq (r\beta - \lambda + 1)k - \binom{k}{2}(\epsilon - 1) \ . \end{aligned}$$

Theorem 5 ([38]). *Let $\mathcal{C}$ be a linear $[n, k, t]$-batch code over $\mathbb{F}$ with the minimum distance d and size of reconstruction sets at most t. Then,*

$$n \geq \max_{\beta \in \mathbb{N} \cap \left[1, \min\left\{t, \left\lfloor \frac{k-3}{2(r-1)} \right\rfloor\right\}\right]} \left\{ \max_{\epsilon, \lambda \in \mathbb{N} \cap [1, r\beta - \beta]} \{\min\{\mathbb{A}, \mathbb{B}, \mathbb{C}\}\} \right\} \ . \tag{9}$$

10 Open Questions

Below, we list some open questions related to batch and PIR codes.

1. Derive tighter bounds on the length or redundancy of batch and PIR codes, in particular, for small alphabet size, for large values of t, or for bounded values of r .
2. Construct new optimal or sub-optimal batch and PIR codes.
3. Do non-linear batch (or PIR) codes have better parameters than their best linear counterparts?
4. Do non-systematic batch (or PIR) codes have better parameters than their best systematic counterparts?
5. Propose batch and PIR codes that allow for efficient reconstruction algorithms.

Acknowledgements The material in this survey has benefited a lot from discussions of the author with his students and colleagues, including Venkatesan Guruswami, Camilla Hollanti, Helger Lipmaa, Sushanta Paudyal, Eldho Thomas, Alexander Vardy, Hui Zhang and Jens Zumbrägel. This work is supported in part by the grants PUT405 and IUT2-1 from the Estonian Research Council and by the EU COST Action IC1104.

References

1. D. Augot, F. Levy-Dit-Vehel, and A. Shikfa, "A storage-efficient and robust private information retrieval scheme allowing few servers," arXiv:1412.5012, Dec. 2014.
2. 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.
3. S.R. Blackburn and T. Etzion, "PIR array codes with optimal PIR rates," arXiv:1609.07070, Sept. 2016.
4. R.A. Brualdi, K. 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.
5. C. Bujtás and Z. Tuza, "Batch codes and their applications," *Electronic Notes in Discrete Mathematics*, vol. 38, pp. 201–206, 2011.
6. V. Cadambe and A Mazumdar, "An upper bound on the size of locally recoverable codes," in *Proceedings International Symposium on Network Coding (NetCod),* pp. 1-5, June 2013.
7. T.H. Chan, S. Ho, and H. Yamamoto, "Private information retrieval for coded storage," arXiv:1410.5489, Oct. 2014.
8. Y.M. Chee, F. Gao, S. T. H. Teo, and H. Zhang, "Combinatorial systematic switch codes," in *Proceedings IEEE International Symposium on Information Theory (ISIT),* Hong Kong, China, pp. 241–245, June 2015.
9. B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," in *Proceedings 36-th IEEE Symposium on Foundations of Computer Science (FOCS),* pp. 41-50, 1995.

10. A.G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, No. 3, March 2011.
11. A.G. Dimakis, A. Gál, A.S. Rawat, and Z. Song, "Batch codes through dense graphs without short cycles," arXiv:1410.2920, Oct. 2014.
12. A. Fazeli, A. Vardy, and E. Yaakobi, "PIR with low storage overhead: coding instead of replication," arXiv:1505.06241, May 2015.
13. M. Forbes and S. Yekhanin, "On the locality of codeword sysmbols in non-linear codes," *Discrete Math*, vol. 324, pp. 78–84, 2014.
14. P. Gopalan, C. Huang, H. Simitchi, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. on Inform. Theory,* vol. 58, no. 11, pp. 6925-6934, Nov. 2012.
15. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Batch codes and their applications," in *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, June 2004, Chicago, IL.
16. S. Kopparty, S. Saraf, and S. Yekhanin, "High-rate code with sublinear-time decoding," in *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC),* pp. 167-176, New York, NY, 2011.
17. H. Lipmaa and V. Skachek, "Linear batch codes," in *Proceedings 4th International Castle Meeting on Coding Theory and Applications,* Palmela, Portugal, Sept. 2014. Earlier version arXiv:1404.2796, Apr. 2014.
18. J.L. Massey, "Threshold decoding," Technical Report TR-410, MIT, 1963.
19. S. Paudyal, "Multi-symbol locally repairable codes," Master's Thesis, University of Tartu, June 2015.
20. R. Pellikaan, X.-W. Wu, S. Bulygin, and R. Jurrius, "Error-correcting codes," preprint available at http://www.win.tue.nl/~ruudp/courses/2WC09/2WC09-book.pdf.
21. N. Prakash, V. Lalitha, and P.V. Kumar, "Codes with locality for two erasures," in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, pp. 1962-1966, June-July 2014.
22. S. Rao and A. Vardy, "Lower Bound on the Redundancy of PIR Codes," arXiv:1605.01869, May 2016.
23. A.S. Rawat, A. Mazumdar, and S. Vishwanath, "Cooperative local repair in distributed storage," *EURASIP Journal on Advances in Signal Processing,* Dec. 2015.
24. A.S. Rawat, D.S. Papailiopoulos, A.G. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," *IEEE Trans. on Information Theory*, vol. 62, no. 8, pp. 4481–4493, Aug. 2016.
25. R.M. Roth, "Introduction to Coding Theory," Cambridge University Press, Cambridge, UK, 2006.
26. 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.
27. N. Silberstein, A.S. Rawat, O.O. Koyluoglu, S. Vishwanath, "Optimal locally repairable codes via rank-metric codes," in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, Istanbul, Turkey, pp. 1819-1823, July 2013.
28. D. Stinson, R. Wei, and M. Paterson, "Combinatorial batch codes," *Advances in Mathematics of Communications*, vol. 3, no. 1, pp. 13–17, 2009.
29. I. Tamo and A. Barg, "Bounds on locally recoverable codes with multiple recovering sets," in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, Honolulu, HI, pp. 691-695, June-July 2014.
30. I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4661-4676, Aug. 2014.
31. A. Vardy and E. Yaakobi, "Constructions of batch codes with near-optimal redundancy," in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, Barcelona, Spain, pp. 1197-1201, July 2016.

32. A. Wang and Z. Zhang, "Repair locality with multiple erasure tolerance," *IEEE Trans. Information Theory,* vol. 60, no. 11, pp. 6979-6987, Nov. 2014.
33. Z. Wang, O. Shaked, Y. Cassuto, and J. Bruck, "Codes for network switches," in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, Istanbul, Turkey, pp. 1057–1061, July 2013.
34. Z. Wang, H.M. Kiah, and Y. Cassuto, "Optimal binary switch codes with small query size," in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, China, pp. 636–640, June 2015.
35. T. Westerbäck, R. Freij, C. Hollanti, "Applications of polymatroid theory to distributed storage systems," in *Proceedings 53rd Allerton Conference on Communication, Control, and Computing,* Allerton, IL, USA, pp. 231–237, Sept.-Oct. 2015.
36. T. Westerbäck, R. Freij-Hollanti, T. Ernvall, C. Hollanti, "On the combinatorics of locally repairable codes via matroid theory," *IEEE Transactions on Information Theory*, vol. 62, no. 10, pp. 5296–5315, Oct. 2016.
37. M. Wootters, "Linear codes with disjoint repair groups," unpublished manuscript, Feb. 2016.
38. H. Zhang and V. Skachek, "Bounds for batch codes with restricted query size," in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, Barcelona, Spain, pp. 1192-1196, July 2016.
39. Y. Zhang, X. Wang, H. Wei, and G. Ge, "On private information retrieval array codes," arXiv:1609.09167, Sept. 2016.
40. J. Zumbrägel and V. Skachek, "On bounds for batch codes," in *Algebraic Combinatorics and Applications (ALCOMA)*, Kloster Banz, Germany, March 2015.