Update Efficient Codes for Error Correction

Arya Mazumdar and Gregory W. Wornell
Dept. EECS, MIT, Cambridge, MA 02139
email: {aryam, gww}@mit.edu

Venkat Chandar
MIT Lincoln Laboratory, Lexington, MA 02420
Email: vchandar@mit.edu

***Abstract*—An update efficient code is a mapping from messages to codewords such that small perturbations in the message induce only slight changes to the corresponding codeword. The parameter that captures this notion is called *update-efficiency*. In this paper we study update-efficient error-correcting codes and develop their basic properties. While update-efficiency and error-correction are two conflicting objectives, we deduce conditions for existence of such codes. In particular, logarithmically growing update-efficiency is achievable with a capacity-achieving linear code in both binary symmetric and binary erasure channels. On the other hand we show a tight converse result. Our result implies that it is not possible to have a capacity-achieving code in binary symmetric channel that has sub-logarithmic update-efficiency. This is true in the case of the binary erasure channel as well for linear codes. We also discuss a number of questions related to update-efficient adversarial error-correcting codes.**

I. Introduction

In a variety of applications such as distributed storage networks, there is a need for update-efficient codes. Such networks consist of multiple distributed and unreliable storage devices across which dynamically changing information must be stored. Each time a portion of the information content changes, its associated reliable encoding must also change, and the contents of the storage devices updated accordingly. In such applications, to minimize the communication bandwidth and power resources required by such networks, it is desirable to have encodings that are update-efficient, i.e., minimize the number of storage devices affected when the information content changes. In this paper we examine aspects of the degree to which such update-efficient codes are possible.

In our development, a *code* $\mathcal{C} \in \mathbb{F}_2^n$ is a collection of binary n-vectors with a one-to-one encoding map $\phi : \mathbb{F}_2^k \to \mathcal{C}, k < n$. We restrict our attention specifically to *error-correcting codes* (i.e., channel codes). The *support* of a vector $\boldsymbol{x}$, denoted $\mathrm{supp}(\boldsymbol{x})$, is the set of coordinates where $\boldsymbol{x}$ has nonzero values. In turn, the *weight* of a vector, denoted $\mathrm{wt}(\boldsymbol{x})$, is the size its support.

The notion of *update-efficiency* for channel codes is introduced in [2]. A code is called update-efficient if for a small change in the message $\boldsymbol{x} \in \mathbb{F}_2^k$, the corresponding codeword $\phi(\boldsymbol{x})$ changes only slightly. Formally, we have the following definition.

This work was supported in part by the US Air Force Office of Scientific Research under Grant No. FA9550-11-1-0183, by the National Science Foundation under Grant No. CCF-1017772, and by Hewlett-Packard Labs.

The results of this paper can be generalized to q-ary alphabets ($q > 2$) with little effort.

Definition 1: A code $(\mathcal{C}, \phi)$ is (u, t)-*update-efficient* if for all $\boldsymbol{x} \in \mathbb{F}_2^k$, and for all $\boldsymbol{e} \in \mathbb{F}_2^k$ such that $\mathrm{wt}(\boldsymbol{e}) \leq u$, we have $\phi(\boldsymbol{x}+\boldsymbol{e}) = \phi(\boldsymbol{x})+\boldsymbol{e}'$, for some $\boldsymbol{e}' \in \mathbb{F}_2^n$ such that $\mathrm{wt}(\boldsymbol{e}') \leq t$. For much of our development in this paper, we focus on the case where $u = 1$, and equivalently refer to an $(1, t)$-update-efficient code as one having *update efficiency* t.

As background for the present work, much of the initial work on update efficient codes has focused on their use on the binary erasure channel (BEC), which is a natural model for capturing server failures. For example, in [2] it is shown that there exist capacity-achieving codes for the BEC with update-efficiency $O(\log n)$. A subsequent paper [13] shows, using the randomized codes proposed [2], that it is possible for capacity-achieving codes for the BEC to have both update-efficiency and repair-bandwidth efficiency, a property desirable in distributed storage. Yet another recent paper [9] considers the update-efficiency of linear codes.

In this paper we are concerned with update-efficient codes that also correct errors. In addition to distributed storage applications, another potential application for such codes is transmitting uncompressed video over a noisy communication link. As the messages (video-frames) only change slightly from one frame to the next, update-efficient codes can be an efficient mechanism for encoding the sequence of frames.

We consider two models of error: random and adversarial. The random errors take the form of independent bit flips, corresponding to a binary symmetric channel (BSC), with flip probability $0 < p < 1/2$. Such a channel is denoted via BSC(p).

Correcting errors is generally a more difficult task than correcting erasures. In this paper we show that the result for erasures in the paper [2] carries over for the case of errors, viz., there exist linear codes with update efficiency $O(\log n)$ that achieve rates arbitrary close to the BSC(p) capacity of $1 - \mathrm{h}(p)$, where $\mathrm{h}(p) = -p \log p - (1-p)\log(1-p)$ is the binary entropy function, with probability of error approaching zero. In addition, we show a converse result. To be specific, we show that for a suitable $\alpha > 0$, within the ensemble of positive rate linear codes with update efficiency $\alpha \log n$, almost all codes have a probability of error bounded away from zero on the BSC(p). We also show a stronger version of this result, namely, that for some other $\alpha > 0$, there does not exist a code with update-efficiency $\alpha \log n$ that has both positive rate and arbitrarily small probability of error.

All logarithms in this paper are base 2 unless otherwise indicated.

978-1-4673-2579-0/12/$31.00 ©2012 IEEE

For the case of adversarial errors, update-efficiency and error correction are conflicting objectives. However, following [2] and [11], it can be shown that there exist codes with update-efficiency $O(\log n)$ that correct any pn adversarial errors if there is sufficient shared randomness between the encoder and the decoder. We discuss several properties of linear codes that result in good update-efficient error correcting codes, and give a number of examples. Finally, we turn our attention to general (u, t)-update-efficient codes and provide bounds on the size of an update-efficient code in terms of the minimum distance of the code.

In this paper, we focus primarily on linear codes, which are attractive in terms of representation and both encoding and decoding complexity. A linear code $(\mathcal{C}, \phi)$ is such that $\phi : \mathbb{F}_2^k \to \mathcal{C}$ is a homomorphism. It can always be represented by a $k \times n$ *generator matrix* G such that $\phi(\boldsymbol{x}) = \boldsymbol{x}^T G$, for any $\boldsymbol{x} \in \mathbb{F}_2^k$. Note that G for a code is not unique; distinct G give different labelings. By an $[n, k, d]$ code we mean a linear code with length n, dimension k and minimum distance d.

For a linear code, the maximum number of bits that change in the codeword when one bit in the message changes is the maximum over the weights of the rows of the generator matrix. Hence, for an update-efficient code, we need a representation of the linear code where the maximum weight of the rows of the generator matrix is low.

Proposition 1: A linear code $\mathcal{C}$ will have update-efficiency t if and only if there is a generator matrix G of $\mathcal{C}$ with maximum row weight t.

Proof: It is easy to see that if the maximum number of ones in any row is bounded above by t, then at most t bits need to be changed following a one bit change in the message.

On the other hand, if the code has update-efficiency t then there must exist a labeling ϕ that gives a sparse generator matrix. Specifically, the vectors $(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0 \ldots, 1) \in \mathbb{F}_2^k$ must produce vectors of weight at most t under ϕ, so the generator matrix given by ϕ has row weight at most t. ∎

Proposition II-B implies that given a linear code, to see whether it is update-efficient or not, we need to find the sparsest basis for the code. A linear code with a sparse basis is informally called a *low density generator matrix (LDGM)* code.

II. LDGM codes and capacity of BSC

A. No good codes with update efficiency $< \alpha \log n$

We start this section with a negative result regarding update-efficient codes on the BSC. Suppose we want to construct a code with update efficiency t. We will look at the ensemble of linear codes with update-efficiency t and show that almost all codes in this ensemble are bad for t less than certain value. Proposition II-B shows that a linear code with update efficiency t always has a generator matrix with maximum row weight t. For simplicity, we consider generator matrices where all rows have weight t, but all the results can be extended to the case where the row weight is at most t.

Let $\Gamma_{n,k,t}$ be the set of all $k \times n$ matrices over $\mathbb{F}_2$ such that each row has exactly t ones. First, we recall the following lemma from [5], which shows that almost all the matrices in $\Gamma_{n,k,t}$ generate codes with dimension k (i.e., the rank of the matrix is k).

Lemma 2: Randomly and uniformly choose a matrix G from $\Gamma_{n,k,t}$. If k is such that $k \leq \left(1 - \frac{e^{-t}}{\ln 2} - o(e^{-t})\right)n$, then with probability $1 - o(1)$ the rank of G is k.

This lemma, along with the next theorem, prove that almost all codes in $\Gamma_{n,k,t}$ are bad for small t.

Theorem 3: Fix an $0 < \alpha < 1/2$, and assume that $k \geq n^\alpha, t \leq \sqrt{n}/2$. Then, for at least $1 - \frac{t^2 n^{2\alpha}}{n-t}$ proportion of the matrices in $\Gamma_{n,k,t}$, the associated linear code has probability of error at least $\frac{n^\alpha}{\sqrt{t}} 2^{-\lambda_p t}$ over a BSC(p) for $p < 1/2$ and $\lambda_p = -1 - 1/2 \log p - 1/2 \log(1-p) > 0$.

Before proving this theorem, we state the following corollary.

Corollary 4: For at least $1 - o(1)$ proportion of all linear codes with update efficiency $t < \frac{\alpha - \epsilon}{\lambda_p} \log n, \alpha < 1/2, \epsilon > 0$ and dimension k, $k > n^\alpha$, the probability of error is $1 - o(1)$ over a BSC(p) for $p < 1/2$.

In particular, this shows that codes with update efficiency $< \log n/(2\lambda_p)$ and rate $> n^{\alpha - 1}$ are almost always bad.

Proof of Corollary 4: From Lemma 2 it is clear that $1 - o(1)$ proportion of all codes in $\Gamma_{n,k,t}$ have rank k. Hence, if $1 - o(1)$ proportion of codes in $\Gamma_{n,k,t}$ have some property, $1 - o(1)$ proportion of codes with update-efficiency t and dimension k also have that property. Plugging in the value of t in the expression for the error probability from Theorem 3 gives the corollary. ∎

To prove Theorem 3 we will need the following series of lemmas.

Lemma 5: Let $\boldsymbol{x} \in \{0, 1\}^n$ be a vector of weight t. Let the all-zero vector of length n be transmitted over a BSC with flip probability $p < 1/2$. If the received vector is $\boldsymbol{y}$, then,

$$\Pr(\mathrm{wt}(\boldsymbol{y}) > d_H(\boldsymbol{x}, \boldsymbol{y})) \geq \frac{1}{\sqrt{t}} 2^{-\lambda_p t},$$

where $\lambda_p = -1 - 1/2 \log p - 1/2 \log(1-p) > 0$.

The proof is omitted.

Lemma 6: Suppose two random vectors $\boldsymbol{x}, \boldsymbol{y} \in \{0, 1\}^n$ are chosen independently and uniformly from the set of all length-n binary vectors of weight $t \leq \sqrt{n}/2$. Then,

$$\Pr(\mathrm{supp}(\boldsymbol{x}) \cap \mathrm{supp}(\boldsymbol{y}) = \emptyset) > 1 - \frac{t^2}{n - t + 1}.$$

Proof: The probability in question equals,

$$\begin{aligned}
\frac{\binom{n-t}{t}}{\binom{n}{t}} &= \frac{((n-t)!)^2}{(n-2t)!n!} \\
&= \frac{(n-t)(n-t-1)(n-t-2)\ldots(n-2t+1)}{n(n-1)(n-2)\ldots(n-t+1)} \\
&= \left(1 - \frac{t}{n}\right)\left(1 - \frac{t}{n-1}\right)\ldots\left(1 - \frac{t}{n-t+1}\right) \\
&> \left(1 - \frac{t}{n-t+1}\right)^t \geq 1 - \frac{t^2}{n-t+1}.
\end{aligned}$$

In the last step we have truncated the series expansion of $\left(1-\frac{t}{n-t+1}\right)^t$ after the first two terms. The inequality will be justified if the terms of the series are decreasing in absolute value. Let us verify that to conclude the proof. In the following X_i denote the ith term in the series, $0 \le i \le t$.

$$\frac{X_{i+1}}{X_i} = \frac{\binom{t}{i+1}}{\binom{t}{i}} \cdot \frac{t}{n-t+1} = \frac{t-i}{i+1} \cdot \frac{t}{n-t+1} \le 1,$$

for all $i \le t-1$. ∎

Lemma 7: For $0 < \alpha < 1/2$, choose random vectors $\boldsymbol{x}_i, 1 \le i \le n^\alpha$ of weight $t \le \sqrt{n}/2$ independently and uniformly from the set of weight-t vectors. Then,

$$\Pr(\forall i \ne j, \mathrm{supp}(\boldsymbol{x}_j) \cap \mathrm{supp}(\boldsymbol{x}_i) = \emptyset) \ge 1 - \frac{t^2 n^{2\alpha}}{n-t}.$$

This implies all of the vectors have disjoint supports with probability at least $1 - \frac{t^2 n^{2\alpha}}{n-t}$.

Proof: The claim follows from taking a union bound over all pairs of randomly chosen vectors. ∎

Now we are ready to prove Theorem 3.

Proof of Theorem 3: We begin by choosing a matrix G uniformly at random from $\Gamma_{n,k,t}$. This is equivalent to choosing each row of G uniformly and independently from the set of all weight-t binary vectors. Now $k > n^\alpha$, hence there exist n^α vectors among the rows of G such that any two of them have disjoint support with probability at least $1-\frac{t^2 n^{2\alpha}}{n-t}$ (from Lemma 7). Hence for at least a proportion $1 - \frac{t^2 n^{2\alpha}}{n-t}$ of matrices of $\Gamma_{n,k,t}$, there are n^α rows with disjoint supports. Suppose G is one such matrix. It remains to show that the code $\mathcal{C}$ defined by G has probability of error at least $\frac{n^\alpha}{\sqrt{t}} 2^{-\lambda_p t}$ over a BSC(p).

Suppose, without loss of generality, that the all zero vector is transmitted over a BSC(p) and $\boldsymbol{y}$ is the vector received. Let $\boldsymbol{x}_i, 1 \le i \le n^\alpha$ be codewords of weight t with disjoint support, guaranteed to exist by our assumption on $\mathcal{C}$. The probability that the maximum likelihood decoder incorrectly decodes $\boldsymbol{y}$ to $\boldsymbol{x}_i$ satisfies

$$\Pr(\mathrm{wt}(\boldsymbol{y}) > d_H(\boldsymbol{x}_i, \boldsymbol{y})) \ge \frac{1}{\sqrt{t}} 2^{-\lambda_p t}$$

from Lemma 5. As the codewords $\boldsymbol{x}_1, \ldots \boldsymbol{x}_{n^\alpha}$ have disjoint supports, the probability that the maximum likelihood decoder incorrectly decodes to any one of them is at least $\frac{n^\alpha}{\sqrt{t}} 2^{-\lambda_p t}$. ∎

It should be noted that the above theorem is easily extendable to the random ensemble of matrices whose entries are independently chosen from $\mathbb{F}_2$ with $\Pr(1) = t/n$.

B. No good codes with update efficiency $< \alpha \log n$, revisited

In this subsection, for a smaller α than discussed above, we show that no code can simultaneously achieve low probability of error on the binary symmetric channel and have update complexity less than $\alpha \log n$. More precisely, we give two results. The first result says that linear codes cannot have low update complexity when used over the binary erasure channel (BEC). Since the binary symmetric channel is degraded with respect to the binary erasure channel, this is a stronger result. To see that a BSC(p) is a degraded version of BEC with erasure probability $2p$, one can concatenate a BEC($2p$) with a channel with ternary input $\{0, 1, ?\}$ and binary output $\{0, 1\}$ such that with probability 1 the inputs $\{0, 1\}$ remain the same, and with uniform probability ? goes to $\{0, 1\}$.

Our second result says that for the binary symmetric channel, even non-linear codes cannot have low update efficiency. We include both results because although the second result applies to more general codes, we have not been able to extend the second result to the binary erasure channel. We conjecture that for positive rates, even nonlinear codes must have logarithmic update complexity for the binary erasure channel (at zero rate, trivial counterexamples can be found).

The proof for linear codes used over a binary erasure channel is based on Proposition , i.e., when the update complexity is low, the generator matrix G is very sparse. Let the random subset $I \in \{1, \ldots, n\}$ denote the coordinates not erased by the binary erasure channel. Let G_I denote the submatrix of G induced by the unerased received symbols, i.e., the columns of G corresponding to I. Then, because G is so sparse, it is quite likely that G_I has several all zero rows, and the presence of such rows implies a large error probability. We formalize the argument below.

Theorem 8: Consider using some linear code of length n, dimension k and update-efficiency t, specified by generator matrix G over BEC(p). Assume that, for some $\epsilon > 0$, $t < \ln \frac{-k^2}{2n \ln \epsilon} / (-2 \ln p)$. Then, the average probability of error is at least $1/2 - \epsilon$.

Proof: For linear codes over the binary erasure channel, analyzing the probability of error essentially reduces to analyzing the probability that the matrix G_I induced by the unerased columns of G has rank k (note that the rank is computed over $\mathbb{F}_2$). To show that the rank is likely to be less than k for sufficiently small t, let us first compute the expected number of all zero rows of G_I. Since G has update-efficiency t, every row of G has weight at most t, so the expected number of all zero rows of G_I is at least kp^t. The rank of G_I, $\mathrm{rk}(G_I)$, is at most k minus the number of all zero rows, so the expected rank of G_I is at most $k - kp^t$.

Now, observe that the rank is a 1-Lipschitz functional of the independent random variables denoting the erasures introduced by the channel. Therefore, by Azuma's inequality [1, Thm. 7.4.2], the rank of G_I satisfies

$$\Pr(\mathrm{rk}(G_I) \ge \mathbb{E}\,\mathrm{rk}(G_I) + \lambda) < e^{-\frac{\lambda^2}{2n}}.$$

Therefore,

$$\Pr(\mathrm{rk}(G_I) \ge k - kp^t + \lambda) < e^{-\frac{\lambda^2}{2n}}.$$

In particular,

$$\Pr(\mathrm{rk}(G_I) = k) < e^{-\frac{k^2 p^{2t}}{2n}}.$$

Assuming the value given for t we see that

$$\Pr(\mathrm{rk}(G_I) = k) < \epsilon.$$

Since even the maximum likelihood decoder makes an error with probability at least 0.5 when $\mathrm{rk}(G_I) < k$, this shows that when $t < \ln \frac{-k^2}{2n \ln \epsilon}/(-2 \ln p)$, the probability of error is at least $1/2-\epsilon$. (In fact, the average error probability converges to 1. The above argument can easily be extended to show that the probability of decoding successfully is at most $e^{-\Omega(\frac{k^\delta}{\log k})}$ for some $\delta > 0$, but we omit the details.) ∎

Now, we show that even nonlinear codes cannot have low update efficiency for the binary symmetric channel. The argument is based on a simple observation. If a code has dimension k and update efficiency t, then any given codeword has k neighboring codewords within distance t, corresponding to the k possible 1-bit changes to the information bits. If t is sufficiently small, it is not possible to pack $k+1$ codewords into a Hamming ball of radius t and maintain a low probability of error.

Theorem 9: Consider using some (possibly non-linear) code of length n, dimension (possibly fractional) k, and update-efficiency t over BSC(p). Assume that for some $\alpha > 0$, $t \leq (1-\alpha) \log k / \log((1-p)/p)$. Then, the average probability of error is at least $1 - o(1)$, where $o(1)$ denotes a quantity that goes to zero as $k \to \infty$.

The proof will appear in the full version of the paper.

C. Good codes exists with update efficiency $O(\log n)$

On the other hand, it is relatively easy to construct a code with update efficiency $O(\log n)$ that achieves capacity on the binary symmetric channel. One can in principle choose the rows of the generator matrix randomly from all low weight vectors and argue that this random ensemble contain many codes that achieve capacity of the binary symmetric channel (BSC). Some steps in this direction have been made in [10]. However there are easier ways to construct capacity achieving codes that have update efficiency $O(\log n)$. Let us describe one such construction[1].

It is known that for every $\epsilon > 0$ and any sufficiently large n, there exist a linear code of length n and rate $1 - \mathrm{h}(p) - \epsilon$ that has probability of error at most $2^{-E(p,\epsilon)n}$. There are numerous evaluations of this result and estimates of $E(p, \epsilon) > 0$. We refer the reader to [3] as an example.

Let $m = \frac{1+\alpha}{E(p,\epsilon)} \log n$, for $\epsilon, \alpha > 0$ (we avoid using ceiling and floor to have a clean presentation). We know that for sufficiently large n, there exists a linear code $\hat{\mathcal{C}}$ given by the $mR \times m$ generator matrix $\hat{G}$ with rate $R = 1 - \mathrm{h}(p) - \epsilon$ that has probability of error at most $2^{-E(p,\epsilon)m}$.

Let G be the $nR \times n$ matrix that is the Kronecker product of $\hat{G}$ and the $n/m \times n/m$ identity matrix $I_{n/m}$, i.e.,

$$G = I_{n/m} \otimes \hat{G}.$$

Clearly a codeword of the code $\mathcal{C}$ given by G is given by n/m codewords of the code $\hat{\mathcal{C}}$ concatenated side-by-side. The probability of error of $\mathcal{C}$ is therefore, by the union bound, at most

$$\frac{n}{m} 2^{-E(p,\epsilon)m} = \frac{nE(p,\epsilon)}{(1+\alpha)n^{1+\alpha} \log n} = \frac{E(p,\epsilon)}{(1+\alpha)n^{\alpha} \log n}.$$

However, notice that the generator matrix has row weight bounded above by $m = \frac{1+\alpha}{E(p,\epsilon)} \log n$. Hence, we have constructed a code with update efficiency $\frac{1+\alpha}{E(p,\epsilon)} \log n$, and rate $1 - \mathrm{h}(p) - \epsilon$ that achieves a probability of error $< \frac{E(p,\epsilon)}{(1+\alpha)n^{\alpha} \log n}$ on a BSC(p).

III. Adversarial channel

In the adversarial error model, the adversary is allowed to introduce up to s errors at locations of his choice. It is known that to correct s adversarial errors the minimum distance $d(\mathcal{C})$ of a code $\mathcal{C}$ needs to be at least $2s+1$. However, if a code has update-efficiency t, then there must exist two codewords within distance t of each other. Hence, small update-efficiency implies limited error correction capability. We investigate these observations in more detail below.

A. Correcting adversarial errors with a randomized code

Although it is impossible for a fixed error-correction code with small update efficiency to correct a large number of errors, if we randomize the code then it is possible to fool the adversary. In fact, with a randomized code it is possible to correct pn adversarial errors with a code rate approaching the capacity of a BSC(p). This idea has been used in the case of erasures in [2]. Let $(\hat{\mathcal{C}}, \hat{\phi})$ be a random code defined as follows from another code $(\mathcal{C}, \phi)$. Suppose $\sigma \in S_n$ is an uniform random permutation on the set $\{1, \ldots, n\}$, and $\boldsymbol{z} \in \mathbb{F}_2^n$ is a uniform random vector. The random encoding in $\hat{\mathcal{C}}$ is defined by $\hat{\phi}(\boldsymbol{x}) = \sigma(\phi(\boldsymbol{x})) + \boldsymbol{z}, \boldsymbol{x} \in \mathbb{F}_2^k$. If the operation of the decoding algorithm of $\mathcal{C}$ and $\hat{\mathcal{C}}$ are denoted by ψ and $\hat{\psi}$ respectively, then $\hat{\psi}(\boldsymbol{y}) = \psi(\sigma^{-1}(\boldsymbol{y} + \boldsymbol{z})), \boldsymbol{y} \in \mathbb{F}_2^n$. We have the following theorem that stems from [11].

Theorem 10: Let $\mathcal{C}$ be a codes with rate $1 - \mathrm{h}(p) - \epsilon$ that achieves probability of error approaching 0 as $n \to \infty$ over a BSC(p). Suppose $\hat{\mathcal{C}}$ is a random code formed as above. Then, against any adversarial pn errors, the code $\hat{\mathcal{C}_n}$ will have probability of error approaching 0 as $n \to \infty$.

In the above theorem if we take the code $\mathcal{C}$ to be the explicit code designed in Section II-C, then the code $\hat{\mathcal{C}}$ remains an update-efficient code with update-efficiency $O(\log n)$. Hence by sharing $O(n \log n)$ bits between the encoder and decoder, it is possible to correct a large number of adversarial errors with a high rate code. We omit the proof of the above theorem here, as it follows directly from [11].

B. Codes with small weight bases and the Griesmer bound

As noted at the start of this section, in an error correcting code minimum distance and update efficiency are conflicting objectives. A code with distance d must have update-efficiency at least d because the nearest codeword is at least distance d away. If the update-efficiency of the code $\mathcal{C}$ is denoted by $t(\mathcal{C})$ then $t(\mathcal{C}) \geq d(\mathcal{C})$, where $d(\mathcal{C})$ is the minimum distance of the code. Hence, the aim of a code-designer would be to design

[1]This construction was suggested by Yury Polyanskiy in private communication.

a code whose update-efficiency is as close to the distance as possible. As we saw in the Introduction, for a linear code $\mathcal{C}$ the update efficiency is simply the weight of the maximum weight row of a generator matrix. We recall the following theorem from [6].

Theorem 11: Any binary linear code of length n, dimension k and distance d has a generator matrix consisting of rows of weight $\leq d+s$ where $s = \left(n - \sum_{j=0}^{k-1} \left\lceil \frac{d}{2^j} \right\rceil\right)$ is a nonnegative integer.

The fact that s is a non-negative integer also follows from the well-known Griesmer bound [12] that states for any linear code with length n, dimension k and distance d, $n \geq \sum_{j=0}^{k-1} \left\lceil \frac{d}{2^j} \right\rceil$.

Corollary 12: For any linear $[n,k,d]$ code $\mathcal{C}$ with update-efficiency t, $d \leq t \leq d + \left(n - \sum_{j=0}^{k-1} \left\lceil \frac{d}{2^j} \right\rceil\right)$.

It is clear that for codes achieving the Griesmer bound with equality, the update-efficiency is exactly equal to the minimum distance, i.e., the best possible. There are a number of families of codes that achieve the Griesmer bound. For examples of such families and their characterizations we refer the reader to [4], [7].

Example: Suppose $\mathcal{C}$ is an $[n = 2^m - 1, k = 2^m - 1 - m, 3]$ Hamming code. For this code $t(\mathcal{C}) \leq 3 + (n-3-2-(k-2)) = n - k = m = \log(n+1)$. In fact, for all n, Hamming codes have update-efficiency 3. One way to prove this is by explicitly constructing a generator matrix for the Hamming code with weight 3 rows. One can also appeal to the following theorem of Simonis [14].

Theorem 13: Any $[n,k,d]$ binary linear code can be transformed in to a code with same parameters that has a generator matrix consisting of only weight d rows.

The implication of this theorem is that if there exists an $[n,k,d]$ linear code, then there exists an $[n,k,d]$ linear code with update-efficiency d. In his paper [14], Simonis gave an algorithm to transform any linear code into an update-efficient linear code (a code with update-efficiency equal to the minimum distance). However, the algorithm is of exponential complexity. It is of interest to have a polynomial time algorithm for the procedure.

On the other hand, the above theorem says that there exists a linear $[n = 2^m - 1, k + 2^m - 1 - m, 3]$ code that has update-efficiency only 3. All codes with these parameters are equivalent to the Hamming code of the same parameters up to a permutation of coordinates [8], providing an indirect proof that Hamming codes have update-efficiency 3.

Analysis of the update-efficiency of BCH codes and other linear codes is of independent interest. In general, finding a sparse basis for a linear code seems to be a hard problem.

IV. General update efficient codes

Let us now return now to codes satisfying Definition 1. Extending the $u = 1$ case, clearly any (u,t)-update-efficient code must satisfy $t \geq d$, the minimum distance of the code, but for general u we can strengthen this bound.

Proposition 14: Suppose a (u,t)-update-efficient code of length n, dimension k and minimum distance d exists. Then $\sum_{i=0}^{u} \binom{k}{i} \leq B(n,d,t)$, where $B(n,d,w)$ is the size of the largest code with distance d such that each codeword has weight at most w.

Proof: Suppose $\mathcal{C}$ is an update-efficient code where $\boldsymbol{x} \in \mathbb{F}_2^k$ is mapped to $\boldsymbol{y} \in \mathbb{F}_2^n$. Now, the $\sum_{i=0}^{u} \binom{k}{i}$ different message vectors that are within distance u from $\boldsymbol{x}$ should map to codewords within distance t from $\boldsymbol{y}$. Suppose these codewords are $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$. Consider the vectors $\boldsymbol{y} - \boldsymbol{y}, \boldsymbol{y}_1 - \boldsymbol{y}, \boldsymbol{y}_2 - \boldsymbol{y}, \ldots$. These must be at least distance d apart from one another and all of their weights are at most t. This proves the claim. ■

It is not very difficult to construct a $(u, O(u \log n))$ update efficient code that achieves the capacity of a BSC(p) by modifying the constructions of Section II-C. On the other hand, one expects a converse result of the form

$$\sum_{i=0}^{u} \binom{k}{i} \leq K(n,t,p),$$

where $K(n,t,p)$ is the maximum size of a code with codewords having weight bounded by t that achieves arbitrarily small probability of error. A formal expression for $K(n,t,p)$ is a subject of our ongoing work.

Acknowledgement: A. M. thanks Yury Polyanskiy and Barna Saha for useful discussions.

References

[1] N. Alon, J. Spencer, *The Probabilistic Method,* Wiley-Interscience, 2000.

[2] N. P. Anthapadmanabhan, E. Soljanin, S. Vishwanath, "Update-efficient codes for erasure correction," *48th Annual Allerton Conference on Communication, Control, and Computing,* pp. 376–382, October, 2010.

[3] A. Barg, G. D. Forney, Jr., "Random codes: minimum distances and error exponents," *IEEE Transactions on Information Theory,* vol. 48, no. 9, pp. 2568–2573, September, 2002.

[4] B. I. Belov, V. N. Logachev, V. P. Sandimirov, "Construction of a class of linear binary codes achieving the Varshamov/Griesmer bound," *Probl. Peredachi Inf.*, vol. 10, issue 3, pp. 36–44, 1974.

[5] N. Calkin, "Dependent sets of constant weight binary vectors," *Combinatorics, Probability and Computing,* vol. 6, no. 3, pp. 263–271, 1997.

[6] S. D. Dodunekov, N. L. Manev, "An improvement of the Griesmer bound for some small minimum distances," *Discrete Applied Mathematics,* vol. 12, pp. 103–114, 1985.

[7] T. Helleseth, "New constructions of codes meeting the Griesmer bound," *IEEE Transactions on Information Theory,* vol. 29, no. 3, May, 1983.

[8] W. C. Huffman, V. Pless, *Fundamentals of Error-Correcting Codes,* Cambridge, 2003.

[9] A. Jule, I. Andriyanova, "Some Results on update complexity of a linear code ensemble," *International Sym. on Network Coding*, pp. 1–5, 2011.

[10] A. M. Kakhaki, H. K. Abadi, P. Pad, H. Saeedi, F. Marvasti, K. Alishahi, "Capacity achieving linear codes with binary sparse generating matrices," *preprint*, arXiv:1102.4099, 2011.

[11] M. Langberg, "Private codes or Succinct random codes that are (almost) perfect," Proc. Foundations of Computer Science, pp. 325–334, 2004.

[12] F. Macwilliams, N. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, 1977.

[13] A. S. Rawat, S. Vishwanath, A. Bhowmick, E. Soljanin, "Update efficient codes for distributed storage," *IEEE International Symposium on Information Theory*, pp. 1457–1461, Jul 31–Aug 5, 2011.

[14] J. Simonis,"On generator matrices of codes," *IEEE Transactions on Information Theory,* vol. 38, no. 2, March, 1992.