(67902) Topics in Theory and Complexity Nov 1, 2006

Lecture 1

Lecturer: Irit Dinur *Scribe: Robby Lampert*

1 Error-Correcting Codes

1.1 Introduction

There are several different point of views on error correcting codes. We will use the one of coding theory, which is different, for example, from the information theoretic point of view.

Definition 1 *Given an alphabet Σ and two integers $k, n > 0$, a* **code** *is a function $C : \Sigma^k \to \Sigma^n$.*

The basic settings are that we have an input of size k over Σ, and we want the code to generate an output of size n over Σ, such that the following hold.

- The code is **decodable**, i.e., given a code word we can easily find the source of it. This implies that the code must be one-to-one (thus $n \geq k$).
- The code is **error-correcting**, i.e., even if a code word is changed to some extent (by error or maliciously), we can still decode it. This implies that the code words should be as different from each other as possible, since if, for example, there are two code words that are different in only one letter, then if this letter is changed, the decoding is wrong, thus the code is not even single-error-correcting.

Typically, we use $\Sigma = \{0, 1\}$, thus we use *bits* rather then *letters*. Note that in order for a code to be single-error-correcting, every two code words must be different in ≥ 3 bits. Otherwise, if there are two code words that are different only in one bit, then one error can turn one of them to the other and change the decoding. Even if there are two code words that are different in two bits, if one of these bits changes in one of these words, we cannot know which one of the two code words was the output. But if every two code words are different in 3 bits or more, then if one error occurs, all the code words except the one in which the error occurred are different from this word in ≥ 2 bits.

Definition 2 *A p-***repetition code** *is a code $C : \Sigma^k \to \Sigma^{(p \cdot k)}$ such that for every $\mathbf{a} = (a_1 a_2 \dots a_k) \in \Sigma^k$ we have $C(\mathbf{a}) = a_1^p a_2^p \dots a_k^p$, where a_i^p stands for p repetitions of a_i.*

Clearly, in the 3-repetition code, every two code words are different in ≥ 3 bits. Thus, this code is single-error-correcting. Nevertheless, the 3-repetition code is extremely inefficient, reducing throughput by three times, and the efficiency drops drastically as we increase the number of times each bit is duplicated in order to detect and correct more errors. [The last sentence is cited from wikipedia: http://en.wikipedia.org/wiki/Hamming_code]

Definition 3 *The* **information ratio** *of a code is the size of the input (k) divided by the size of the output (n).*

For example, the information ratio of the 3-repetition code is 1/3. As the information ratio approaches 1, it means that less bits are 'wasted', thus the code is more efficient in a sense.

1.2 Hamming Code

Definition 4 *The* **Hamming code** *is the code* $C : \{0,1\}^4 \to \{0,1\}^7$ *such that for every* $\mathbf{a} \in \{0,1\}^4$ *we have* $C(\mathbf{a}) = \mathbf{a}G$ *(the product of the vector* $\mathbf{a}$ *and the matrix* G*), where*

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

G *is called the* **code generating matrix**.

Definition 5 *For* $x, y \in \{0,1\}^m$*, we define the* **weight** *of* x *as* $\mathrm{wt}(x) = |\{1 \le i \le m : x_i = 1\}|$*, and the* **Hamming distance** *between* x *and* y *as* $\mathrm{dist}(x,y) = |\{1 \le i \le m : x_i \ne y_i\}|$.

Claim 6 *For* $a, b \in \{0,1\}^4$*, if* $a \ne b$ *then* $\mathrm{dist}(aG, bG) \ge 3$ *(the code words for* a *and* b *differ by at least 3 bits).*

Proof Note that for $x, y \in \{0,1\}^m$, $\mathrm{dist}(x,y) \equiv \mathrm{wt}(x-y)$, where $x-y$ is taken (mod 2). Thus,

$$\mathrm{dist}(aG, bG) = \mathrm{wt}(aG - bG) = \mathrm{wt}((a-b)G).$$

So, we have to show that for every $c \ne \bar{0}$ we have $\mathrm{wt}(cG) \ge 3$.

To show that, we prove the following propositions first.

Proposition 7 *For every matrix* G *of size* 4×7*, there exists a matrix of size* 7×3 *with full column rank such that* $GH = 0$*. Such* H *is called the* **parity-check matrix** *of the code.*

Proof Since G has 4 rows, they span a subspace of $\{0,1\}^7$ of dimension 4 at most. Thus, due to dimension considerations, there exist 3 vectors in $\{0,1\}^7$, which are orthogonal to each other and to the rows of G. Let the columns of H be 3 such vectors, so clearly $GH = 0$. ■

Proposition 8 *Let* G *be the code generating matrix and let* $H \in M_{7\times 3}$ *be as defined above. Then,*

$$\left\{aG \in \{0,1\}^7 : a \in \{0,1\}^4\right\} = \left\{y \in \{0,1\}^7 : yH = \bar{0}\right\}.$$

Proof First we show that the left set is a subset of the right set. Let $x \in \{0,1\}^7$ be a member of the left set, that is, $x = aG$ for some $a \in \{0,1\}^4$. Then, $xH = aGH = a \cdot 0 = \bar{0}$; thus x is a member of the right set too.

Now note that both sets form subspaces of $\{0,1\}^7$ of dimension 4. The left set is the image of a linear transformation with a domain of dimension 4 and a kernel of dimension 0 (since the 4 left columns of G are the unit vectors of $\{0,1\}^4$, we get $aG = \bar{0}$ iff $a = \bar{0}$). The right set is the kernel of a linear transformation with a domain of dimension 7 and an image of dimension 3 (h has a full column rank). As the left set is a subset of the right one and both have the same dimension, the sets are equivalent. ■

The two latter propositions essentially say that the parity-check matrix H defines the code as well as G does. For the current Hamming code

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

Now we return to our proof. We have to show that for every $c \neq \bar{0}$ we have $\mathrm{wt}(cG) \geq 3$. Thus, we can show instead that for every $y \neq \bar{0}$ such that $yH = \bar{0}$ (that is, by Proposition 8, $y = cG$ for some $c \neq \bar{0}$) we have $\mathrm{wt}(y) \geq 3$.

Assume by way of contradiction that $\mathrm{wt}(y) < 3$. Since $y \neq \bar{0}$, $\mathrm{wt}(y)$ may equal 1 or 2. In case $\mathrm{wt}(y) = 1$, y is a vector of 0's, except a single 1 in the i'th position. Then yH is the i'th row of H. Since no one of the rows of H is $\bar{0}$, $yH \neq \bar{0}$, and we have reached a contradiction. Otherwise, $\mathrm{wt}(y) = 2$. In this case yH is the bitwise sum of the rows of H corresponding to the two positions in which y=1. Since all the rows of H are different, $yH \neq \bar{0}$, and we have reached a contradiction again. Thus the claim is proved. ■

Definition 9 *An $(n, k, d)_\Sigma$-code is a code $C : \Sigma^k \to \Sigma^n$ where Σ is the alphabet, n is the block size (the length of the code words), k is the code dimension (the length of the input words), and $d = \min_{x \neq y} \mathrm{dist}(C(x), C(y))$ (the minimal Hamming distance between two code words) is the code distance. In case that C is linear, it is called a* **linear code** *and denoted [n,k,d]-code.*
Sometimes we drop Σ, when it is understood from the context.

The Hamming code we saw is a $(7,4,3)_2$-code. What if we want to code words of greater length? There is a family of Hamming codes. For each $l > 1$, there is a Hamming $(2^l - 1, 2^l - l - 1, 3)$-code with a generating matrix $G \in M_{2^l - l - 1 \times 2^l - 1}$ and a parity-check matrix $H \in M_{2^l - 1 \times l}$. For example, for $l = 5$, there is a Hamming (31,26,3)-code. Note that as approaches ∞, the information ratio of Hamming code approaches 1, which is optimal, but on the other hand, the distance of the code is constant (3), which is bad, since as the code words grow longer, we may expect more errors to occur, and if the distance is 3 then no matter what the length of the code word is, the code is only single-error-correcting. In general, a code with distance d can correct t errors, where $2t < d$.

Definition 10 *For an (n, k, d)-code C, the* **rate** *of C is defined $r = k/n$ and the* **relative distance** *of C is defined $\delta = d/n$.*

An optimal code is a code in which both r and δ are maximized, but we cannot have both parameters close to 1, since there is a trade off between them. If we want the relative distance of the code to be large, so that we can correct a number of errors which is relative to the block length of the code, we should increase n relatively to k, but then we decrease r. In addition to optimization of these parameters, a good code should have a fast and easy encoding and decoding.
For example, in Hamming code, if the code word z does not have more than a single error, then there is an efficient algorithm to restore the code word. In this case we have $z = y + e$ where $\mathrm{wt}(e) \leq 1$ and $yH = \bar{0}$. Thus, we caculate $zH = (y + e)H = yH + eH = eH$, and if $\mathrm{wt}(e) = 1$ we get the row of H corresponding to the position of the 1 in e. Since every row of H is the binary representation of the row number, we have that zH is exactly the binary representation of the position in z that we have to flip to get y.
To summarize, in Hamming code we have $r = k/n \to 1$ but $\delta = d/n \to 0$.

1.3 Trade-offs between r and δ

Definition 11 *For a point $x \in \{0,1\}^n$, the* **ball** *of radius t around x is defined*

$$B(x, t) = \{z \in \{0,1\}^n : \mathrm{dist}(x, z) \leq t\}.$$

We denote the number of strings in this ball by $\mathrm{Vol}_2(n, t)$ (this number is equal for all $x \in \{0,1\}^n$).

Remark Since $\mathrm{Vol}_2(n, t) = \sum_{i=0}^{t} \binom{n}{i} \approx 2^{H(\frac{t}{n}) \cdot n}$, Where $H(p) = p \log_2(\frac{1}{p}) + (1 - p) \log_2(\frac{1}{1-p})$ for $0 \leq p \leq 1$ is the **entropy** function, we occasionally replace $\mathrm{Vol}_2(n, t)$ by $2^{H(\frac{t}{n}) \cdot n}$.

Proposition 12 (Hamming bound) *For an $(n,k,d)_2$-code we have $2^k \cdot \mathrm{Vol}_2(n, \lfloor \frac{d}{2} \rfloor) \leq 2^n$.*

Proof The domain of the code is of size 2^k. For each two different strings $x, y \in \{0,1\}^k$ the corresponding code words are in distance $\geq d$. Thus, for every string in $\{0,1\}^k$ there is a ball of radius $\lfloor d/2 \rfloor$ in $\{0,1\}^n$, such that all the balls are disjoint. The proposition immediately follows. ■

Corollary 13 *For an $(n,k,d)_2$-code we have $r + H(\frac{\delta}{2}) \leq 1$.*

Proof In Proposition 12, we replace $\mathrm{Vol}_2(n, \lfloor \frac{d}{2} \rfloor)$ by $2^{H(\frac{d}{2n}) \cdot n}$. Then we take log and divide by n and we get $\frac{k}{n} + H(\frac{d}{2n}) \leq 1$. This is, by definition, what we need to show. ■

We have shown a bound on r and δ. Since $H(x$ is symmetric around its maximum at $x = 1/2$, where $H(1/2) = 1$, and decreasing to both sides, where $H(x) = 0$ for $x = 0$ and for $x = 1$, we get on one hand that if r approaches 1, then δ must be very small, and on the other hand, if δ approaches 1 (and therefore $H(\frac{d}{2n})$ approaches 1 too), r must be very small.
Remark In Hamming code, the Hamming bound holds as an equality. For a given $l > 1$, the parameters of Hamming code are $n = 2^l - 1$, $k = 2^l - l - 1$, and $d = 3$. Thus, if we take log of the Hamming bound, on the left hand we have $k + \log(\mathrm{Vol}_2(n, \lfloor \frac{d}{2} \rfloor)) = 2^l - l - 1 + \log(\mathrm{Vol}_2(n,1)) = 2^l - l - 1 + \log(n+1) = 2^l - l - 1 + \log(2^l - 1 + 1) = 2^l - l - 1 + l = 2^l - 1 = n$, Which is equal to what we have on the right hand. In this sense, Hamming code is optimal.

Though the Hamming code is optimal in the sense that the hamming bound holds for it as an equality, it is far from optimal, as it allows only one error, independently of the block length, which may grow as we desire (as mentioned above, $r \to 1$, while $\delta \to 0$). Thus, we introduce below two other codes in which neither r nor δ equal 0.

1.4 Greedy code

Given n and d, we iteratively construct a subset $C \subseteq \{0,1\}^n$ (the image of the code) such that the pairwise minimal distance in C is $\geq d$. In addition to C, we maintain a subset $S \subseteq \{0,1\}^n$ which contains all the words that are not contained in any of the balls of radius $d-1$ around the words in C. Thus, we initiate the construction with $C = \emptyset$ and $S = \{0,1\}^n$. Then, at each iteration until S is empty, we pick some $x \in S$ at random, add x to C, and remove from S the ball of radius $d-1$ around x.
To analyze this code, first note that since d and n are parameters, we may choose them as we desire, and in particular, $\delta = d/n \geq \varepsilon > 0$. Secondly, note that the size of C determines the code dimension. To bound the size of C from below, observe that in the worst case all the balls around the words in C are pairwise disjoint. In this case, the size of C is the size of $\{0,1\}^n$ divided by the number of words in each ball. Thus,

$$|C| \geq \frac{2^n}{\mathrm{Vol}_2(n, d-1)} \geq \frac{2^n}{\mathrm{Vol}_2(n, d)} = \frac{2^n}{\mathrm{Vol}_2(n, \delta n)} \approx \frac{2^n}{2^{H(\delta)n}} = 2^{n(1-H(\delta))}.$$

Therefore, $k = \log|C| \geq n(1 - H(\delta))$, so $r = k/n \geq (1 - H(\delta))$. Since for reasonably small δ ($< \frac{1}{2}$) we have $H(\delta) < 1$, it implies $r > 0$, as required.
To conclude, the Hamming bound assures $r \leq 1 - H(\frac{\delta}{2})$, and here we have $r \geq 1 - H(\delta)$. Yet, while Hamming code is linear (uses a linear transformation for encoding), this code has no efficient encoding and decoding. Furthermore, this code is not practical, since there is no efficient algorithm for picking at random from a subset of a large set.

1.5 Random code

Given n and r (thus $k = rn$), pick in random a matrix $G \in M_{k\times n}$ over $\{0,1\}$, and let it be the generating matrix of the code, i.e., for $x \in \{0,1\}^k$, $C(x) = xG$.

Claim 14 *For δ such that $r + H(\delta) < 1$, there exists a code of the above form, whose distance is $\geq \delta n$.*

Proof We show that the probability over all $G \in M_{k\times n}$ of the event in which there exists some $a \in \{0,1\}^k$ such that $a \neq \bar{0}$ and $\mathrm{wt}(aG) < \delta n$ is < 1. Thus, there exists some $G \in M_{k\times n}$, for which for all $a \neq 0$ we have $\mathrm{wt}(aG) \geq \delta n$, as required.

Let X_a be an indicator variable, whose value equals 1 iff $\mathrm{wt}(aG) < \delta n$. Then

$$\Pr_G[\exists a \neq \bar{0} \text{ s.t. } \mathrm{wt}(aG) < \delta n] = \Pr_G[\bigcup_{a\neq\bar{0}} \mathrm{wt}(aG) < \delta n] = \Pr_G[\sum_{a\neq\bar{0}} X_a > 0] \overset{(*)}{\leq} \sum_{a\neq\bar{0}} \Pr_G[X_a > 0] \leq \sum_a \Pr_G[X_a > 0],$$

where (*) is valid due to the union bound. Now, for a fixed a, aG is the linear combination of the rows of G with the coefficients a_i. But since G is random, so is aG. Thus,

$$\Pr_G[X_a > 0] = \Pr_{y\in\{0,1\}^n}[\mathrm{wt}(y) < \delta n] \leq \Pr_{y\in\{0,1\}^n}[y \in B(\bar{0}, \delta n)] = \frac{\mathrm{Vol}(n, \delta n)}{2^n} \approx \frac{2^{H(\delta)n}}{2^n} = 2^{n(H(\delta)-1)},$$

and therefore

$$\sum_a \Pr_G[X_a > 0] \leq \sum_a 2^{n(H(\delta)-1)} = 2^k \cdot 2^{n(H(\delta)-1)}.$$

Now, this number is less then 1 iff (taking log) $k + n(H(\delta) - 1) < 0 \Leftrightarrow \frac{k}{n} + H(\delta) < 1$. Since $r = \frac{k}{n}$, the claim is proved. ■