A Theorist's Toolkit **(CMU 18-859T, Fall 2013)**

Lecture 10: Error-correcting Codes

October 9, 2013

Lecturer: Ryan O'Donnell *Scribe: Xiaowen Ding*

1 Overview

In information theory and coding theory, error detection and correction are techniques that enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error correction enables recovery of the original data.

Information theory and electrical engineering often focus on cases that errors are random, while computer scientists focus on worst case.

2 Basic definitions

Definition 2.1 (Error-Correcting Codes)**.** Error-correcting codes is an injecting map from k symbols to n symbols:

$$\textit{Enc}\colon\ \Sigma^k \to \Sigma^n$$

where Σ is the set of symbols.

- q: We say $q = |\Sigma|$ is the cardinality of alphabet set. In particular, for $q = 2$, the code is binary code.
- *Message*: The domain Σ^k is called the **message space** and elements in Σ^k, which would be transmitted are called **message**. And k is called message length or **message dimension**.
- *Block length*: Messages would be mapped to n-bit strings. Here, n is called **block length**. In general, $k < n$.
- *Code*: We write C for the image, C would have cardinality q^k because Enc is an injective function. We often call C the "**code**" and $y \in C$ a "**codeword**". For efficiency, we don't want n to be too big than k.
- *Rate*: We call $\frac{k}{n}$ the "**rate**", and higher rates indicates higher efficiency.

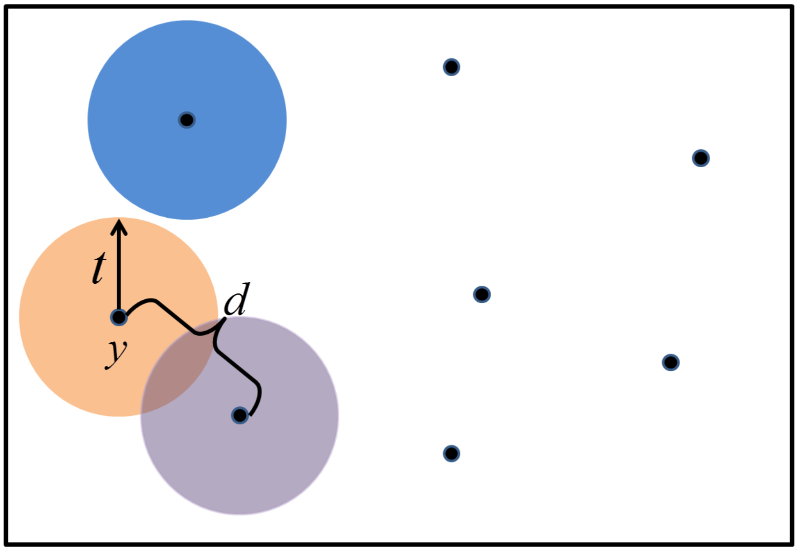
$$\begin{tikzcd}
x \arrow[r, "Enc"] & y \arrow[r, squiggly, "errors"] & z \arrow[r, "recover?"] & x \\
\Sigma^k \arrow[u] & \Sigma^n \arrow[u] & \Sigma^n \arrow[u] & \Sigma^k \arrow[u]
\end{tikzcd}$$

The above picture indicates the procedure of encoding and decoding. Here, errors mean up to t symbols of y are changed, and "change" here means switching to another symbol. The decoding is recovering the original x from z.

3 Hamming distance

Definition 3.1 (Hamming distance)**.** Hamming distance is the number of positions at which the corresponding symbols are different.

$$\Delta(y, z) = |\{i : y_i \neq z_i\}|$$



In the graph above, the box stands for all elements in Σ^n. Each vertex is a codeword and each ball around is the possible z the receiver gets.

t stands for the maximum number of errors. For codeword y here, everything is the Hamming ball has Hamming distance to y up to t. If there doesn't exist two Hamming balls that overlap each other, we can recover every message. In the graph, the orange ball overlaps the purple one. Then sometimes we cannot recover x that is associated with y correctly.

Here, d indicates the **minimum distance** between two vertices.

Definition 3.2. Minimum distance is the least distance between two distinct codewords:

$$d = \min_{y \neq y' \in C} \{\Delta(y, y')\}$$

Fact 3.3. *Unique decoding, which means for each z the receiver gets, there is a unique x he can recover, is possible if and only if $t \leq \lfloor \frac{d}{2} \rfloor$*

We usually want d to be large so that the area it can correct would be large. But by doing this, the number of vertices we can put in Σ^n would become small. In many ways, coding theory is about exploring the tradeoff.

`Question:` Why not just choose $C \subset \Sigma^n$ randomly?

`Answer:` Inefficient encoding and decoding because we might have to maintain a huge table.

4 Linear code

In coding theory, a linear code is an error-correcting code for which any linear combination of codewords is also a codeword. In fact, linear codes allow for more efficient encoding and decoding algorithms than other codes.

A linear code of length n and rank k is a linear subspace C with dimension k of the vector space $\mathbb{F}_q^n$ where $\mathbb{F}_q$ is the finite field with q elements.

Definition 4.1 (Linear code)**.** *Enc:* $\mathbb{F}_q^k \to \mathbb{F}_q^n$, linear map:

$$x \mapsto Gx$$

where x is a vector and G is a matrix

Here, G is called "*Generator matrix*" which is a full-rank $n \times k$ matrix that makes the linear map injective.

$C = Im(G)$ is the image of G which spans all linear combinations of rows.

Notation 4.2.

$$[n, k(, d)]_q$$

where n is the length of codeword, k is the length of message, and d is the minimum distance if known.

$[\cdot]$ indicates that it is a linear code. For codes that are not linear, we should use $(\cdot)$.

Remember that encoding is very efficient for given G.

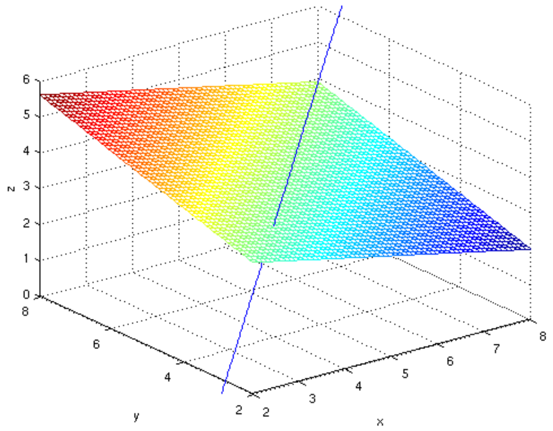
Then, let's focus on decoding. It's not obvious whether decoding is efficient. Given $y = Gx$, it is easy to recover x. But if we get a z that is not in the code space C, it might be NP-hard to find the nearest y that associated with a x. So we should design G s.t. decoding is efficient.

Although it might be hard to find nearest y for z, it is easy to check if y is in the code. One way to do it is to check whether it is a span of the row vectors, another way is to check whether it is orthogonal to a bunch of vectors.

Let $C^\perp = \{w \in \mathbb{F}_q^n : w^T y = 0, \forall y \in C\}$, every vector in $C^\perp$ are orthogonal to every codeword.

Since C's dimension is k, $C^{\perp}$ is a subspace of dimension $n-k$.

In our mind, in 3-dimension space, two subspaces that are orthogonal would be like a line and a plane in the following graph:



But in finite fields, thing would like different.

Example 4.3. *$q=2$, $n=2$,*

$C=\{(0,0),(1,1)\}$,

$C^{\perp}=\{(0,0),(1,1)\}$

$C^{\perp}$ is same with C.

$C^{\perp}$ gives rise to a $[n,n-k]_q$ code, $Enc^{\perp}:\mathbb{F}_q^{n-k}\rightarrow\mathbb{F}_q^{n}$ maps w to H^Tw, where H is a $(n-k)\times n$ matrix like $\begin{bmatrix} \quad & H & \quad \end{bmatrix}$

Here H is called parity check matrix of the original code C.

$C^{\perp}$ is rowspan of H, so $z\in C \Leftrightarrow Hz=0$.

When $q=2$, each row of H is a binary string, and a string z is in the original code C iff $Hz=\overrightarrow{0}$. Since every bit in H is either 0 or 1, it's like checking the parity of some subsets of z.

In general, for a linear code that has some nice properties, we would usually look at the dual and find another nice properties.

Definition 4.4 (Hamming weight)**.** Hamming weight of w is $wt(w)=\Delta(0,w)$.

Fact 4.5. *$d(c)$ is the least Hamming weight of a nonzero codeword. This is because $\Delta(y,y')$ is Hamming weight of $y-y'$.*

Fact 4.6. *$d(c)$ is the minimum number of columns of H which are linearly dependent.*

Proof.

$$\begin{aligned} d(c) &= \min\{wt(z): z\in C, z\neq 0\} \\ &= \min\{wt(z): Hz=0, z\neq 0\} \end{aligned}$$

The columns of H that associated with the non-zero entries of z are linearly dependent. □

5 Hamming Code

Hamming code [Ham50] is defined by the case of linear code that $q = 2$:

$$H = \begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & \cdots & 1 \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 1 & \cdots & \cdots & \cdots & 1 \\ 1 & 0 & 1 & \cdots & \cdots & \cdots & 1 \end{bmatrix}$$

where H is a $r \times (2^r - 1)$ matrix and columns span all possible binary of length r except the all-zero column.

H is full-rank because it has the identity matrix. And distance is 3 because from fact 4.6, the minimum number of linearly dependent columns is 3.

Ham_r is $[2^r - 1, 2^r - 1 - r, 3]_2$, or we can say it's $[n, n - \log_2(n + 1), 3]_2$ for $n = 2^r - 1$. This code has excellent rate $\frac{k}{n} \approx 1$. Basically, we just need to add $\log_2 n$ bits to the message and transit it. And since the distance is 3, it is uniquely decodable from up to $\lfloor \frac{3}{2} \rfloor = 1$ error.

In fact, we can correct one error easily. First, we can check if $Hz = 0$. If it is, y is not modified. Otherwise, one bit changed, we know that for some i, $z = y + e_i$, so $Hz = H(y + e_i) = Hy + He_i = He_i$, which is just the ith column of H. Actually, the ith column of H is equally to i written in binary. So we can easily find the corresponding i here.

This code is called perfect code:

Definition 5.1 (perfect code)**.** A perfect code may be interpreted as one in which the balls of Hamming radius t centered on codewords exactly fill out the space.

This code is perfect because for each $z \in \Sigma_2^n$ such that $Hz \neq 0$, there exists a column of H: H_i such that $Hz = H_i$ because the columns of H span all possible binary string of length r, and $z + e_i$ is the center of the Hamming ball containing z.

In 1973, it was proved that any non-trivial perfect code over a prime-power alphabet has the parameters of a Hamming code or a Golay code [Gol49].

6 Hadamard Code

The Hadamard code is a code with terrible rate and excellent distance. It is always used for error detection and correction when transmitting messages over very noisy or unreliable channels.

First, let's look at dual generator matrix of Hamming code:

$$H^T = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

Then let's add in an all-zero extra row:

$$G = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

The matrix G above is the generator matrix of Hadamard code.

Definition 6.1 (Hadamard code)**.** Hadamard encoding of x is defined as the sequence of all inner products with x:

$$x \mapsto (a \cdot x)_{a \in \mathbb{F}_2^r}$$

Definition 6.2. Given $x \in \mathbb{F}_2^r$, define r-variate linear polynomial $L_x : \mathbb{F}_2^r \to \mathbb{F}_2$ as

$$a \mapsto x^T a = \sum_{i=1}^{r} x_i a_i$$

Actually, for each x, it is a linear mapping. When x is fix, we can view x_i as coefficients and a_i as variables.

This is like mapping x to the "truth table" of L_x because $(a \cdot x)_{a \in \mathbb{F}_2^r} = (L_x(a))_{a \in \mathbb{F}_2^r}$

Fact 6.3. *Hadamard code is a* $[2^r, r, 2^{r-1}]_2$ *code. Let* $n = 2^r$*, it's a* $[n, \log_2 n, \frac{1}{2}n]_2$ *code.*

For any two distinct "truth tables", they differ in at least half of coordinates. That is, $\forall x \neq 0$, $\mathbf{Pr}_{a \sim \mathbb{F}_2^n}[L_x(a) \neq 0] \geq \frac{1}{2}$ by *Schwartz-Zippel Lemma* [Zip79, Sch80].

Actually, we can view $L_x(a)$ as $x_1 a_1 + x_2 a_2 + \cdots + x_r a_r$. For non-zero x, say $x_i \neq 0$, when fix other coordinates, exactly one of the two cases $a_i = 1$ and $a_i = 0$ would make $L_x(a)$ to be 0. So for half of a's, $L_x(a)$ is non-zero. So $\mathbf{Pr}_{a \sim \mathbb{F}_2^n}[L_x(a) \neq 0] = \frac{1}{2}$.

In general, for alphabet with q symbols, it is a $[q^r, r, (1 - \frac{1}{q})q^r]_q$ code.

7 Reed-solomon Codes

In coding theory, Reed-Solomon(RS) codes are non-binary cyclic error-correcting codes invented by Irving S. Reed and Gustave Solomon. [RS60] Reed-Solomon codes have since found important applications from deep-space communication to consumer electronics. They are prominently used in consumer electronics such as CDs, DVDs, Blu-ray Discs, in data transmission technologies such as DSL and WiMAX, in broadcast systems such as DVB and ATSC, and in computer applications such as RAID 6 systems.

Definition 7.1 (Reed-solomon codes)**.** For $1 \leq k < n$, $q \geq n$, select a subset of symbols of cardinality n, $S \subseteq \mathbb{F}_q$, $|S| = n$. We define *Enc:* $\mathbb{F}_q^k \to \mathbb{F}_q^n$ as following:

For message $m : (m_0, m_1, \cdots, m_{k-1}) \in \mathbb{F}_q^k$,

$$m \mapsto (P_m(a))_{a \in S}$$

where $P_m(x) \in \mathbb{F}_q[x]$ is $m_0 + m_1 x + ... + m_{k-1} x^{k-1}$

Remember the following:

- Usually $q = n$, $S = \mathbb{F}_q$, and P_m is the "truth table" of polynomial
- It is linear code. We can easily check it by definition. We add two messages m and m', $Enc(m+m') = Enc(m)+Enc(m')$ because it's like adding the coefficients of polynomial.
- Generator matrix is a Vandermonde matrix that each row is $[1, \alpha, \alpha^2, \cdots, \alpha^{k-1}]$ for an $\alpha \in S$
- *min-dist* $\geq n - (k-1) = n - k + 1$. The number of different roots of a polynomial of degree $k-1$ is at most $k-1$. So in $[P_m(\alpha_0), P_m(\alpha_1), ..., P_m(\alpha_n)]$, at most $k-1$ entries are 0.

RS code has the above great properties except q has to be at least as large as n. To summarize, it's a $[n, k, n-k+1]_q$ code. In particular, $q = n$, it's a $[n, k, n-k+1]_n$ code. If we set k to be half of n, then the rate would be half and the minimum distance is also half of n.

One thing that might be confusing is that why larger alphabet size would help. One reason is that if the alphabet size is large, say 2^{10}, then each symbol's length is 10 bits. It's like "packing" each 10 bits, and once an error occurs, it would be in a "package" instead of random 10 bits.

In fact, the parameters of this code is optimal. And we will see it in next section.

8 Good codes exist

Theorem 8.1 (Singleton bound)**.** *For a* $[n, k, d]_q$ *code,* $k \leq n - d + 1$ *[Sin64]*

Proof. Assume for the sake of contradiction, $|C| > q^{n-d+1}$, by pigeonhole principle, some two elements in C have same first $n-d+1$ coordinates.

Then Hamming distance $\leq d-1 < d$ □

We observe that this bound doesn't rely on alphabet size. So one problem here is for fixed q (say $q=2$), whether there exist codes with both good rates and good distances.

Fixed q, let $C = \{C_n\}$ be a sequence of codes $[n, k(n), d(n)]_q$

For the codes we learned above:

- Hamming Code: $[n, n - \log_2(n+1), 3]_2$
- Hadamard Code: $[n, \log_2 n, \frac{n}{2}]_2$
- Reed-Solomon Code: It's not a code family since the alphabet size is growing. But if we set $q = n$, it's $[n, k, n-k+1]_n$

We always want the code to be good for infinite many n instead of some particular ones. So we define the following things:

Definition 8.2 (Asymptotic rate)**.**

$$R(C) = \lim_{n\to\infty} \inf \frac{k(n)}{n}$$

Definition 8.3 (Asymptotic relative distance)**.**

$$\delta(C) = \lim_{n\to\infty} \inf \frac{d(n)}{n}$$

Here are asymptotic rate and asymptotic relative distance of codes we learned:

- Hamming Code: $R = 1, \delta = 0$
- Hadamard Code: $R = 0, \delta = \frac{1}{2}$
- Reed-Solomon Code: $R = R_0, \delta = 1 - R_0$ because it depends on how we set k. But it doesn't really count because q is not fixed.

Definition 8.4 (Asymptotically good code family)**.** Asymptotically good code family is a code family such that

$$R(C) \geq R_0 > 0$$
$$\delta(C) \geq \delta > 0$$

Fact 8.5. *Good codes exist.*

There is a natural greedy approach to construct a code of distance at least d. We can randomly add a codeword and erase the part within distance d and repeat adding until we cannot proceed.

Theorem 8.6 (Gilbert-Varshamov bound)**.** *$\forall q \geq 2$, $\forall \delta \in [0, 1-\frac{1}{q}]$, $\exists C$, $\forall n$, with $\delta(C) = \delta$,*

$$R(C) \geq 1 - h_q(\delta) > 0$$

where $h_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$

In fact, The Gilbert-Varshamov bound was proved in two independent works. Gilbert proved this bound for greedy aproach and Varshamov proved this bound by using the probabilistic method for linear code. [Gil52, Var57]

Justesen codes [Jus72] form a class of error-correcting codes that have a constant rate, constant relative distance, and a constant alphabet size.

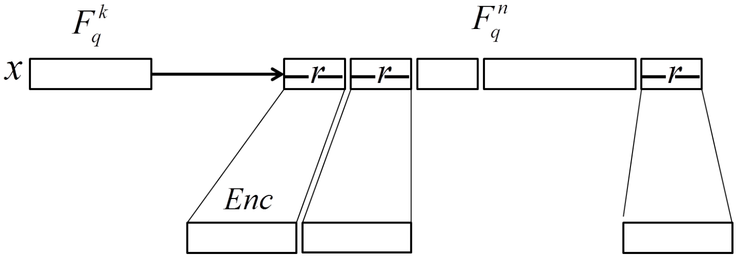
Theorem 8.7 (Justesen'72)**.** *For $q = 2$, $\exists$ polynomial time constructible, encodable, and decodable good code family. $\forall 0 < R < 1$, $\delta \approx \frac{1}{2}(1-R)$ decode from up to $\lfloor \frac{\delta n}{2} \rfloor$ errors.*

Sipser and Spielman [SS96] use expander graphs to construct a family of asymptotically good, linear error-correcting codes.

Theorem 8.8 (Sipser-Spielman'96)**.** *They construct asymptotic good code family ($q = 2$) that can be encodable and decodable in linear time and can be decoded from $\Omega(1)$ fraction of error.*

Concatenated codes are error-correcting codes that are constructed from two or more simpler codes in order to achieve good performance with reasonable complexity. Originally introduced by Forney in 1965 to address a theoretical issue, they became widely used in space communications in the 1970s. [For66]

For example, if we concatenate Reed-Solomon code and Hadamard code:



It has horrible rate and excellent minimum distance.

Theorem 8.9 (Forney'66)**.** *Concatenated codes could be used to achieve exponentially decreasing error probabilities at all data rates less than capacity, with decoding complexity that increases only polynomially with the code block length.*

References

[For66] G David Forney. *Concatenated codes*, volume 11. Citeseer, 1966.

[Gil52] Edgar N Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31(3):504–522, 1952.

[Gol49] Marcel JE Golay. Notes on digital coding. *Proc. ire*, 37(6):657, 1949.

[Ham50] Richard W Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *Information Theory, IEEE Transactions on*, 18(5):652–656, 1972.

[RS60] Irving S Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial & Applied Mathematics*, 8(2):300–304, 1960.

[Sch80] Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980.

[Sin64] R Singleton. Maximum distance-nary codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964.

[SS96] Michael Sipser and Daniel A Spielman. Expander codes. *Information Theory, IEEE Transactions on*, 42(6):1710–1722, 1996.

[Var57] RR Varshamov. Estimate of the number of signals in error correcting codes. In *Dokl. Akad. Nauk SSSR*, volume 117, pages 739–741, 1957.

[Zip79] Richard Zippel. *Probabilistic algorithms for sparse polynomials*. Springer, 1979.