6.895 Essential Coding Theory	September 15, 2004
Lecture 3	
Lecturer: Madhu Sudan	*Scribe: Adi Akavia*

Today's Plan

- Converse coding theorem
- Shannon vs. Hamming theories
- Goals for the rest of the course
- Tools we use in this course

Shannon's Converse Theorem

We complete our exposition of Shannon's theory from last lesson in proving Shannon's Converse Theorem. Namely, we show that there exists no encryption and decryption algorithms (not necessarily efficient) such that transmit messages in rate exceeding $H(p)$.

Let us denote by $BSC_{p,n}$ the Binary Symmetric channel that transmits n bits in each time step, where each bit of the message is flipped w.p. $p \leq \frac{1}{2}$.

Theorem 1 (Shannon's Converse Theorem). *$\forall BSC_{p,n},\ n \geq n_0,\ \varepsilon > 0, R > 1 - H(p) + \varepsilon n$, and for all encoding and decoding $E\colon \{0,1\}^{Rn} \to \{0,1\}^n$ and $D\colon \{0,1\}^n \to \{0,1\}^{Rn}$,*

$$\Pr_{m \in \{0,1\}^{Rn}, \eta \in BSC_{p,n}} [D(E(m) + \eta) = m] \leq \exp(-n)$$

Proof. Informally speaking, the proof is based on two facts: On the one hand, there are roughly $Volume(Ball(p \pm \varepsilon), n)) \approx 2^{H(p)n}$ received words are likely to be a corrupted version of m, since w.h.p $np \pm \varepsilon$ errors falls in any transmitted message m. On the other hand, the average number of received words that decode to m is 2^{n-k}, since $D\colon \{0,1\}^n \to \{0,1\}^k$, for $k = Rn$. Now, since $2^{H(p)n} >> 2^{n-k}$, then each corrupted word is mapped back to its original with only a very small probability.

More formally, fix some encoding and decoding mappings E, D. Let $I_{m,\eta}$ be a *correct decoding indicator*, namely, $I_{m,\eta} = 1$ if $D(E(m) + \eta) = m$, and $I_{m,\eta} = 0$ o/w. We are interested in the sum $\frac{1}{2^k} \sum_{m,\eta} I_{m,\eta}$. To compute this sum, note that when we fix some received vector $r = E(m) + \eta$, we can write the summation as $\sum_r \sum_m I_{m,r-E(m)}$. Now, since there is a unique m_r s.t. $D(r) = m_r$, then $I_{m,r-E(m)} = 1$ iff $m = m_r$, and $I_{m,r-E(m)} = 0$ otherwise. Therefore,

$$\sum_{m,\eta} I_{m,\eta} = 2^n$$

To compute the probability of correct decoding, fix $p' = p - \varepsilon$ (note that p' still satisfies $R > 1 - H(p')$). We consider two types of error vectors η: (a) η has at most $p'n$ non-zero entries, and (b) η has more than $p'n$ non-zero entries. By Chernoff Bound, event (a) happens with very small probability $\exp(-n)$. Therefore,

$$\begin{aligned} \Pr[\text{correct decoding}] &\leq \exp(-n) + \Pr[\text{correct decoding} | \eta \text{ has more than } p'n \text{ non-zero entries}] \\ &= \exp(-n) + \sum_{\eta \notin \mathcal{B}(p'n,n)} \sum_m \Pr[\text{message } m, \text{ error } \eta, I_{m,\eta} = 1] \end{aligned}$$

As the choices of message, error and decoding algorithm are independent, and $\Pr[m] = 2^{-k}$, $\Pr[\eta | \eta \notin \mathcal{B}(p'n, n)] \leq \frac{1}{\binom{n}{p'n}} \approx 2^{-H(p')n}$, we have: $\Pr[\text{message } m, \text{ error } \eta, I_{m,\eta} = 1] \leq 2^{-k} \cdot 2^{-H(p')n}$. Therefore,

$$
\begin{aligned}
\Pr[\text{correct decoding}] &\leq \exp(-n) + \sum_{\eta \notin B(p'n,n), m} 2^{-k} \cdot 2^{-H(p')n} \cdot I_{m\eta} \\
&\leq \exp(-n) + 2^{-(k+H(p')n-n)} = \exp(-n)
\end{aligned}
$$

□

Generalizations of Shannon's Theorems

Shannon's paper profoundly influenced both theory and practice of communication, as well as other fields. The theorems we stated and proved so far cover only a small part of the paper. A few generalization we'd like to mention follows.

Shannon not only considers the Binary Symmetric channel but also channels where the input and output are taken from arbitrary alphabets Σ and Γ. E.g, $\Sigma = \{\pm 1\}$, $\Gamma = \mathbb{R}$. Shannon shows that as long as the error has some finite variance, then we might as well think of the channel as having some *finite* capacity (even when the alphabet in infinite as in $\mathbb{R}$!).

Shannon also considers more general probability distributions on the error of the channel. In particular, Shannon considers Markovian error model. Markovian models can capture situations where there are bursts of huge amounts of error, by considering a finite set of correlated states. This models influenced not only communication and coding theory but also other area such as Natural Language Processing, where it lead to the n-gram model.

Shannon vs. Hamming

We now contrast the works of Shannon and Hamming

- Both state formal mathematical models, and prove both feasibility and infeasibility.
- Constructive vs. Non-constructions. Shanon work deals with constructive settings of encoding and decoding algorithms, while Hamming consider codes as combinatorial objects $\{E(x)\}_x$. Yet, Hamming's proves constructive results (yielding the Hamming code), while the technique of the probabilistic method used by Shannon is very un-constructive.
- Worst-case vs. Probabilistic/Average case theory. Hamming deals with adversarial situations, namely, he analyzes worst-case scenarios; in contrast, Shanon analyzes average-case scenarios with a predefined error model of the channel, or message generation model for the source. Shannon average-case theory gives is far more complete than Hamming's worst-case theory, where there are still many open questions.

Course Goals

In this course we explore questions arising both from Hamming's and from Shannon's works.

Targets arising from Shannon's Theory

The main target arising from Shannon's theory is to explicitly find efficient encoding and decoding function. In particular, for Binary Symmetric channel, BSC_p, can we come up with polynomial-time encoding and decoding functions? Or, better yet, linear-time functions?

Shannon tells us that for any rate $R < 1 - H(p)$ we can encode/decode. However, wjem we are guaranteed $1 - H(p) - R = \varepsilon$, we are also interested in studying the complexity as a function of ε. In fact, a lot of the current research is concentrated on this type of questions.

Targets arising from Hamming's Theory Targets

To find efficient encoding and decoding algorithms we naturally require good codes. This leads us to targets of Hamming Theory of constructing "good codes", and associating with them efficient encoding/decoding functions. Let us remark that in the analysis of this codes we might consider both adversarial and probabilistic models.

To be more accurate, we must specify what are "Good Codes". Let us list the the interesting parameters and whether we want them to be maximized or minimized.

Parameter of Error-Correcting Codes n, k, d, q

- n is the block length, namely, the code is a subset Σ^n. We'd like to minimize n.
- k is the information length, that is the length of the messages. Note that the size of the code is $|Code| = \left|\Sigma^k\right|$. We'd like to maximize k.
- d is the minimum distance of code $d = \min_{x \neq y} \Delta_{Hamming}(x, y)$. We'd like to maximize d.
- q is the alphabet size $|\Sigma|$. It is not clear whether want q to be minimized of maximized. Nonetheless, we general assume that we want to minimize q, as empirical observation indicate that it's easier to design good codes with smaller alphabet.

To simplify the parameters, we usually consider normalized parameters: $R = \frac{k}{n}$ (to be maximized), $\delta = \frac{d}{n}$ (to be minimized), and study R vs. δ. To further simplify the parameters, we also often restrict ourselves to $q = 2$.

Tools

The tools we use in this class usually comes from either Probability Theory or Algebra of Finite Fields. A summary of those tools follows.

Probability Theory

- Linearity of expectation
- Union bound
- Probability of product of independence variables is the product of their individual probabilities
- Tail bounds (*i.e.*, the probability that a random variable deviates from its mean):
  - Markov's Inequality: If $X > 0$, then $\Pr[X > kE[X]] \leq 1/k$
  - Checyshev's Bound: Mardov's inequality applied to $(X - E[X])^2)$
  - Chernoff Bound: if $X_1 ... X_n \in [0, 1]$ are independent random variables with expectations $E[X_i] = p$, then $\Pr[\left|\frac{\sum_i X_i}{n} - p\right| > \varepsilon] \leq e^{-\varepsilon^{2n/2}}$

Algebra of Finite Fields

Fields and Vector Spaces

Definition 2 (Field). *A* Field $(\mathbb{F}, +, \cdot, 0, 1)$ *is a set* $\mathbb{F}$ *with addition and multiplication operations* $+, \cdot$ *(respectively) and special elements* $0, 1$ *such that:*

- *addition forms a commutative group on the elements of* $\mathbb{F}$*,*
- *multiplication forms a commutative group on the elements of* $\mathbb{F} \setminus \{0\}$*, and*
- *there is a distribution-law of multiplication over addition.*

The important thing for us is that finite fields exists.

Theorem 3. *For any prime* p*, and* $m \in Z^+$*, there exists a field* $\mathbb{F}_{p^m}$ *of size* p^m*.*

Where $m = 1$ the field of p elements is $\mathbb{F}_p = \{0, ..., p-1\}$ with addition and multiplication modulo p. For $m > 1$, the field $\mathbb{F}_{p^m}$ is defined by an irreducible polynomial[1] $f(x)$ of degree m with coefficients from $\mathbb{F}_p$, and the operations are taken modulo p as well as modulo $f(x)$. The set of all polynomials over $\mathbb{F}_p$ modulo p and $f(x)$ has precisely p^m elements and it fulfills the requirements of a field. For example,

$$\mathbb{F}_{18} = \left\{\text{polynomials of deg} \leq 17 \text{ with } 0/1 \text{ coefficients, } + \text{ and } \cdot \text{ are mod 2 and mod } x^{18} + x^9 + 1\right\}$$

Definition 4 (Vector Space). *A* Vector Space V *over a field* $\mathbb{F}$ *is a quadruple* $(\mathbb{F}, V, +, \cdot)$*, where* $V = \mathbb{F}^n$*,* $+$ *is vector addition (i.e.,* $(v_1, ..., v_n) + (u_1, ..., u_n) = (v_1 + u_1, ..., v_n + u_n)$*), and* $\cdot$ *is a scalar product (i.e.,* $\alpha(v_1, ..., v_n) = (\alpha v_1, ..., \alpha v_n)$*).*

Linear Codes

We'd like the codes we construct to have nice properties such as succinct representation, efficient encoding, and efficient decoding. Linear codes (as defined below) is a family of codes that has two of these properties: succinct representation, efficient encoding.

Let us first define what is a *linear subspace*.

Definition 5 (Linear Subspace). *Let* V *be a vector space over a field* $\mathbb{F}$*. A subspace* $L \subseteq V$ *is a* linear subspace *if every* $v_1, v_2 \in L$ *and* $\alpha \in \mathbb{F}$ *satisfy:* $v_1 + v_2 \in L$ *and* $\alpha v1 \in L$*.*

Now we define linear codes.

Definition 6 (Linear Codes). *For* Σ *a field* $\mathbb{F}$*,* $C \subseteq \Sigma^n$ *is a* linear code *iff* C *is a linear subspace of* $\mathbb{F}^n$*.*

Linear codes have the two desirable properties of succinct representation and efficient encoding.
Succinct representation: a linear codes is defined by a *basis* (*i.e.*, a set $b_1, ..., b_k \in C$ s.t. $\forall \alpha_1, \alpha_k \in \mathbb{F}$, $\sum \alpha_i b_i = 0$ implies $\alpha_1, .., \alpha_k = 0$).
Efficient encoding: the basis representing the linear code also defines the generator matrix G by having b_i as the ith row of G. This yields an efficient encoding of each x by Gx.

An alternate specification of Linear subspace is by its *null space*:

$$C^{\perp} \left\{ y \mid \langle y, v \rangle = 0 \forall v \in C \right\}$$

(where the inner product $\langle x, y \rangle = \sum x_i y_i$ for any $x, y \in \mathbb{F}$). For linear codes C, the null space $C^{\perp}$ is also linear, and $dim(C^{\perp}) + dim(C) = n$.

Empirically, there seem to be no harm in restricting ourselves to linear code. Therefore the study of linear codes will be a big emphasis of this course. In particular we'll consider codes based on polynomials over finite fields.

[1] Irreducible polynomial over $\mathbb{F}_p$ is one that can't be factored over $\mathbb{F}_p$

Polynomials over finite fields

$\mathbb{F}[X]$ denotes the ring of polynomials over $\mathbb{F}$. The elements of $\mathbb{F}[X]$ are vectors $(c_0, ..., c_k)$ which are associated with the formal polynomial $\sum_{i=0}^{k} c_i x^i$ with addition and multiplication defined the standard way.

The following theorem proves to be incredibly useful. The proof is not hard.

Theorem 7 (Evaluation of polynomials). *Let $C(x) = \sum_{i=0}^{k} c_i x^i$, and let $C(\alpha_1), ..., C(\alpha_q)$ be evaluate of $C(x)$ over fields' elements $\alpha_1, ..., \alpha_q$. If $q > k$, then these evaluations specify the polynomial $C(x)$ uniquely.*

See more details in the algebra notes on web.