CS 6815 Pseudorandomness and Combinatorial Constructions **Fall 2019**

Lecture 14: October 17

Lecturer: Eshan Chattopadhyay *Scribe: Ke Wu*

14.1 Preliminary and Notation

Definition 14.1 (Shannon entropy) *Shannon Entropy of random variable X is defined as:*

$$H(X) := \sum_{x \in supp(x)} p(x) \log(\frac{1}{p(x)}), p(x) := \Pr[X = x]$$

Definition 14.2 (Conditional entropy) *The conditional Entropy of random variable Y given X is defined as:*

$$H(Y|X) := \sum_{x \in supp(x), y \in supp(y)} p(x,y) \log\left(\frac{p(x)}{p(x,y)}\right), p(x) := \Pr[X = x], p(x,y) = \Pr[X = x, Y = y]$$

Denote the Shannon entropy of a Bernoulli random variable X as $H(p)$, where X equals to 1 with probability p, and 0 with probability $1-p$. Then $H(p) = p\log(\frac{1}{p}) + (1-p)\log(\frac{1}{1-p})$.

Theorem 14.3 (Chain rule of Shannon entropy)

$$H(X,Y) = H(X) + H(Y|X)$$

Proof: Let $p(x) = \Pr[X = x], p(x,y) = \Pr[X = x, Y = y]$ and $p(y|x) = \Pr[Y = y|X = x]$. Then we have:

$$\begin{aligned}
H(X,Y) &= \sum p(x,y) \log\left(\frac{1}{p(x,y)}\right) \qquad \text{(by definition)} \\
&= \sum p(x)p(y|x) \left(\log\left(\frac{1}{p(x)}\right) + \log\left(\frac{1}{p(y|x)}\right)\right) \\
&= \sum_x p(x) \left(\sum p(y|x)\right) \log\left(\frac{1}{p(x)}\right) + \sum_x p(x) \left(\sum p(y|x)) \log\left(\frac{1}{p(y|x)}\right)\right) \\
&= H(X) + H(Y|X)
\end{aligned}$$

■

Corollary 14.4 (Chian rule of n random variables)

$$H(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i | X_{<i})$$

where $X_{<i} = (X_1, ..., X_{i-1})$.

Proof: By induction on n. ■

14.2 Existence of Codes

What kinds of codes exist? We will analyze binary codes. Informally, we want to show that

Theorem 14.5 (Informall) *$\exists(n,k,d)_2$ codes where $k=\Omega(n), d=\Omega(n)$. That is, there exists binary codes with constant relative rates and constant relative distances.*

More formally,

Theorem 14.6 (Gilbert-Varshamov bound) *$\forall 0<\delta<\frac{1}{2}, 0<\varepsilon\le 1-H(\delta)$, there exists an $(n,k,d)_2$ code where $\frac{k}{n}\ge 1-H(\delta)-\varepsilon, \frac{d}{n}\ge\delta$.*

Proof: We will use a greedy algorithm to show the existence of claimed binary codes:

Greedy Algorithm

- $\mathcal{C}\leftarrow\emptyset$
- while $\exists v\in\{0,1\}^n$ s.t. $\Delta(v,\mathcal{C})\ge d$, add v to $\mathcal{C}$.
- Output $\mathcal{C}$.

Definition 14.7 (Hamming Ball) *$\forall v\in\{0,1\}^n, r\ge 0$, define $Ball(v,r) := \{w|\Delta(v,w)\le r\}$.*

Definition 14.8 (Volume of Ball) *$\forall n\in\mathbb{N}, r\le n$, define $Vol(n,r) := |Ball(0^n,r)|$. Notice that $\forall v\in\{0,1\}^n, |Ball(v,r)| = |Ball(0^n,r)|$.*

Claim 14.9

$$\bigcup_{c\in\mathcal{C}} Ball(c,d-1) = \{0,1\}^n$$

Proof of:[14.9]

Suppose $v\in\{0,1\}^n$ is not in this union, then $\Delta(v,\mathcal{C})\ge d$. The greedy algorithm would thus add v to $\mathcal{C}$, which indicates that v should then be included in this union. And we get a contradiction. $\square$

Using claim 14.9 we know that $Vol(n,d-1)\cdot|\ell|\ge 2^n$, which means that $Vol(n,d-1)\cdot 2^k\ge 2^n$. Now we are left with the evaluation $Vol(n,d-1)$. We'll show the following bound.

Claim 14.10 $Vol(n,r)\le 2^{nH(\frac{r}{n})}, \forall r\le\frac{n}{2}$

Proof of:[14.10]

Let $X=(X_1,...,X_n)$ be a random variable that is uniform over $Ball(0^n,r)$. Then the entropy of $H(X) = \log(Vol(n,r))$. Notice that

$$\begin{aligned} H(X_1,...,X_n) &= \sum_{i=1}^n H(X_i|X_{<i}) \\ &\le \sum_{i=1}^n H(X_i) = nH(X_1) \\ &\le n\cdot H(\frac{r}{n}) \end{aligned}$$

The last step holds because $n \cdot \mathbb{E}[X_1] = \mathbb{E}[\sum_{i=1}^{n} X_i] \leq r$, which means that $\mathbb{E}[X_1] = \Pr[X_1 = 1] \leq \frac{r}{n}$. Thus $Vol(n, r) \leq 2^{nH(\frac{r}{n})}$. $\square$

Therefore we have $2^{n-k} \leq Vol(n, d-1) \leq 2^{n \cdot H(\frac{d}{n})}$, which means that $1 - \frac{k}{n} \leq H(\frac{d}{n})$. Thus we have $\frac{k}{n} \geq 1 - H(\delta) - \varepsilon$. $\blacksquare$

14.3 Unique Decoding of RS Codes

Definition 14.11 *Define $Poly_{\leq w}$:= univariate polynomials over $\mathbb{F}_q$ of degree less than or equal to w.*

Then the message space of RS code is $Poly_{\leq k-1} : p(x) = \sum_{i=0}^{k-1} \alpha_i x^i$ where $(\alpha_0, ..., \alpha_{k-1}) \in \mathbb{F}_q^k$. The encoding is the evaluation of this polynomial on n distinct points: $(p(\beta_1), ..., p(\beta_n))$. Observe that this is a linear code.

Decoding question: let $(f(\beta_1), f(\beta_2), ..., f(\beta_n))$ for some $f : \mathbb{F}_q \mapsto \mathbb{F}_q$ be a corrupted codeword. If the number of errors is small, can we recover p?

Information theoretically, we can recover up to $\lfloor \frac{d-1}{2} \rfloor$ errors. The intuition is simple: if we have e, where $\lfloor \frac{d-1}{2} \rfloor$, then the corrupted codeword f might be within e distance from two codewords, which makes it impossible for the decoder to uniquely decode from f. As in the figure, f is within distance e to p_1 and p_2 if $e > \lfloor \frac{d-1}{2} \rfloor$:

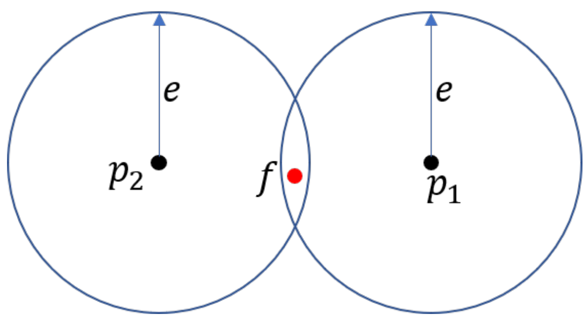

Figure 14.1: $\Delta(p_1, p_2) = d, e > \lfloor \frac{d-1}{2} \rfloor, \Delta(f, p_1) \leq e, \Delta(f, p_2) \leq e$, so the decoder doesn't know which one is being modified.

Formally, the decoding problem is defined as:

Problem 14.12 *given a function $f : \mathbb{F}_q \mapsto \mathbb{F}_q$, and the promise that $\exists p \in Poly_{\leq k-1}$ s.t. the $\frac{e}{q} = \Pr_{x \in \mathbb{F}_q}[f(x) \neq p(x)] \leq \lfloor \frac{d-1}{2} \rfloor \frac{1}{q}$, can we recover $p(x)$ in poly(n) time?*

14.3.1 Welch-Berlekamp Algorithm

For convenience, define $T = \{\beta_i : f(\beta_1) \neq p(\beta_i)\}$ to be the set of all corrupted positions. Then $|T| \leq e$.

Definition 14.13 (Error-locator polynomial) *$E(x)$ is a polynomial over $\mathbb{F}_q$ such that $E(\beta_i) = 0$ iff $\beta_i \in T$.*

For example, the error locator polynomial can be defined as $E(x) = \prod_{\beta_i \in T}(x - \beta_i)$.

Observation 14.14

$$\forall x \in \mathbb{F}_q, E(x)p(x) = E(x)f(x) \tag{$*$}$$

As $E(x)$ is of degree e and the degree of $p(x)$ is smaller than or equal to $k-1$, $(*)$ is a system of n equations on $e+k-1 < n$ variables which are the unknown coefficients of $E(x)$ and $p(x)$. Thus, solving this system would give us the error locator polynomial $E(x)$ as well as the correct codeword $p(x)$. However, $(*)$ is a quadratic system, which is a NP-hard problem in general. We need to figure out an alternative way of decoding in the next lecture.