1 Polar Codes

Last time we gave the sketch of the analysis of polar codes. Today we are going to analyze how polar codes correct erasures, which will give us intuition in the case of errors. Recall the picture of channel capacity martingale, we want to show that

$$\mathbb{P}[X_t \in (\gamma^t, 1-\gamma^t)] \leq \alpha^t.$$

We know that X_t forms a martingale, i.e. $E[X_{t+1}|X_t] = X_t$ and $E[\sqrt{X_{t+1}(1-X_{t+1})}]$ decays geometrically. This implies that taking t steps forward, $\mathbb{P}[X_t \in (\gamma^t, 1-\gamma^t)]$ falls geometrically. However, this is not enough as the decaying factor would be close to 1.

In the case of BEC, all Λ's in the tree are also BEC. Recall that if $\Lambda = (A; B)$ and $(A_1; B_1), (A_2; B_2)$ iid $\sim \Lambda$, then

$$\Lambda_1 = (A_1 + A_2; B_1, B_2)$$
$$\Lambda_2 = (A_2, A_1 + A_2; B_1, B_2)$$

Thus if $\Lambda = \text{BEC}(\lambda)$, then $\Lambda^+ = \text{BEC}(2\lambda - \lambda^2)$ and $\Lambda^- = \text{BEC}(\lambda^2)$ and we can have the explicit expression

$$X_{t+1} = \begin{cases} 2X_t - X_t^2 & \text{with probability } 1/2, \\ X_t^2 & \text{with probability } 1/2. \end{cases}$$

where $X_t = H(\Lambda_i^{(t)})$. The case with errors is more complicated. Unfortunately, we do not have a closed form for X_{t+1} in terms of X_t for the BSC channel. The original proof by Arikan [Ari09] and other analysis of polar codes [ZSW11] uses Bhattacharyya parameter $Z(\Lambda)$. The Bhattacharyya parameter is a way to measure how good a channel is which satisfies the following properties:

- $H(\Lambda^{(t)}) \leq Z(\Lambda^{(t)}) \leq \sqrt{H(\Lambda^{(t)})}$
- $Z(\Lambda^+) = Z(\Lambda)^2$
- $Z(\Lambda)$ is a supermartingale

So to show that a fraction of $H(\Lambda_i^{(t)})$ is small, it is suffice to show a fraction of $Z(\Lambda_i^{(t)})$ is small.

2 Intro to List Decoding

We have seen that a random linear code of rate $1 - h(\delta)$ has relative distance of δ and these codes:

- Can correct up $\delta/2$ worst case errors.
- Can achieve capacity of BSC(δ).

If we want to correct p fraction of errors, by the GV bound, potentially we would have an optimal rate of $1 - h(2p)$ for worst case errors which is the best-known rate. However, in the Shannon case, the optimal rate is equal to $1 - h(p)$, achievable by random linear codes. This leads us the following situation:

- Worst case – Hamming view – pessimistic
- BSC(p) – Shannon view – optimistic

A natural question is: "can we do as well as Shannon model against worst case erasures?" The answer is "NO" if unique/unambiguous decoding is required. In the worst-case model, we want the Hamming balls of radius pn to be disjoint. In the Shannon model, the Hamming balls of radius pn are almost disjoint. List decoding allows the decoding algorithm to have multiple outputs, i.e. a list of codewords. If the algorithm receives y which is close to multiple codewords, it outputs them all. If every $y \in \Sigma^n$ is in $\leq L$ balls with L small ($O(1)$ or $poly(r)$), then every y can be decoded to list of $\leq L$ codewords. A capacity achieving code for BSC does not necessarily satisfy the requirement for list decoding as there can be a small number of y which are in many Hamming balls. However, if you back of from the optimal rate by $1/L$, then list decodable codes exist with rate $1-h(p)-1/L$.

For a code achieving Shannon capacity, it corrects every pattern of pn errors but simply gives up in some of the pathological cases (which happens with small probability), where there are > 1 answers. On the other hand, list decoding:

- Corrects every pattern of pn errors
- Pathological cases, get > 1 answers but list is not too long.
- Many other cases, correct pattern when $p > \frac{\delta}{2}$.
- Corrects all patterns correct by Shannon

Ideally, we want the code to be list decodable, but if the errors happen randomly then we get a unique answer. However, these things need not coexist for a specific code. For a random linear code, the two properties hold with high probability but for a specific code it may be the case that it is list decodable but most of the time you have a list of size 2. For code like RS, we can prove that if the errors happen randomly then the list size would be 1. Now we will prove the information theoretic limit of list decodable codes.

3 List Decoding Capacity

Definition 3.1. A code $C \subset \Sigma^n$ which is (p, L)-list decodable if $\forall y \in \Sigma^n$, $|B(y, pn) \cap C| \leq L$.

Note $(p, 1)$ list decodable codes is equivalent to the distance to the code $\delta(C) \geq 2p$. One may wonder if it is possible if for $p = \delta/2 + \epsilon$ there are too many codewords, i.e. is there are sharp thresholds such that the list size changes dramatically? We want L to be constant or grow with polynomial speed.

Fortunately, this does not happen by Johnson bound. For any $C \subset \{0,1\}^n$, $\delta(C) = \delta$, we know that C is $(J(\delta), O(n))$-list decodable where $J(\delta)$ is defined by $2J(\delta)(1-J(\delta)) = \delta$. If you plot $J(\delta)$, when δ goes to $1/2$, $J(\delta)$ also goes to $1/2$. One might still worry that when you exceed $J(\delta)$, the list side grows exponentially. In fact, there are codes where this happens. For low rates and large δ, $J(\delta)$ is almost $\delta(\delta/2)$. We saw in the homework, just as a function of distance, this is optimal in the sense that if the fraction of errors exceeds $J(\delta)$, then there is a code such that the list size grows exponentially. In principle, every code of distance δ can be list decoded up to $J(\delta)$ fraction of errors. For a typical code of rate $1 - h(\delta) - \epsilon$:

- The code has distance $\sim \delta$.
- List decoding radius $\sim \delta$ (GV bound) with $L = O(1/\epsilon)$.

For a typical code (random linear code), you can correct almost up to the distance of errors and have not too many answers.

Theorem 3.1 (List decoding Capacity)**.** A random code $C \subset \{0,1\}^n$ of rate $1 - h(p) - \frac{1}{L}$ is (p, L)-list decodable with high probability.

The converse of the theorem also holds

Theorem 3.2 (List decoding Capacity Converse)**.** A random code $C \subset \{0,1\}^n$ of rate $1 - h(p) + \epsilon$ is not $(p, 2^{\Omega_\epsilon(n)})$-list decodable with high probability.

Proof. Pick center $y \in \{0,1\}^n$ at random, then

$$\begin{aligned}
&E\left[|B(y, pn) \cap C|\right] \\
=&E\left[\sum_{c \in C} 1\{c \in B(y, pn)\}\right] \\
=&E\left[\sum_{c \in C} 1\{y \in B(c, pn)\}\right] \\
=&\sum_{c \in C} \mathbb{P}\left[y \in B(c, pn)\right] \\
\geq&\sum_{c \in C} \frac{2^{h(p)n - o(n)}}{2^n} \\
\geq&\frac{2^{(1-h(p)+\epsilon)n}}{2^{(1-h(p))n+o(n)}} \\
\geq&2^{\epsilon n/2}
\end{aligned}$$

So there exists a y such that list size exceeds $2^{\epsilon n/2}$. □

Proof. (Capacity Theorem) For a random code C, which is chosen $c_1, ..., c_n$ iid at random from $\{0,1\}^n$ (codewords are allowed to repeat). Fix a $y \in \Sigma^n$, we have by union bound

$$\begin{aligned}
&\mathbb{P}\left[\exists > L \text{ codewords of C in} B(y, pn)\right] \\
=&\sum_{S \subset [m], |S| = C+1} \mathbb{P}\left[c_j, c \in S, c_j \in B(y, pn)\right] \\
\leq&\sum_{S \subset [m], |S| = C+1} (\frac{2^{h(p)}n}{2^n})^{L+1} \\
\leq&2^{(L+1)Rn}(\frac{2^{h(p)}n}{2^n})^{L+1} \\
\leq&2^{(L+1)Rn}2^{-(1-h(p))n(L+1)}
\end{aligned}$$

So

$$\begin{aligned}&\mathbb{P}\left[C \text{ is not (p,L)-list decodable}\right]\\ \leq& 2^n 2^{(L+1)Rn} 2^{-(1-h(p))n(L+1)}\\ \leq& 2^{-\Omega(n)}\end{aligned}$$

□

4 History of List Decoding

List decoding was introduced by Elias, Wozencraft [Eli57] in the 1950s. The motivation was not quite algorithmic. For a rate $R < C$, they want to find the best value of the exponent $E(R, C)$ such that the miscommunication probability is $\leq 2^{-E(R,C)n}$ for Shannon models. If we do not allow list decoding, we only have bounds of $E(R, C)$ and optimal values are not found. Elias found out that if we allow list decoding, then we can determine the optimal value $E(R, C)$.

A more algorithmic revival of list decoding happened in 1989 which is related to cryptography called Goldreich-Levin hardcore predicate [GL89].

Definition 4.1. One way function is a function f such that given x, computing $f(x)$ is easy but finding y such that $f(y) = m$ is very hard on average.

An example is multiplication of two large primes. A hardcore predicate for $f(x)$ is $b : X \to \{0, 1\}$ such that $b(x)$ easy but is very hard to compute $b(f^{-1}(z))$. Goldreich and Levin [GL89] showed that for $b(x, r) = x \cdot r$ is a hardcore predicate for $g(x, r) = (f(x), r)$, i.e. any randomized polynomial time algorithm can only predict $g(x, r)$ with prob at most $1/2 + \epsilon$ by where ϵ is exponentially small. Given algorithm A to predict $b(x, r)$ with $(f(x), r)$ with small error. We can think of the algorithm having query access to the Hadmard encoding of x. As the algorithm runs in polynomial time, it can have only polynomial number of queries. Assuming such algorithm A exists, Goldreich and Levin showed how to list decode Hadmard code with a small list size of $O(1/\epsilon^2)$, which is a contradiction to the converse of the capacity theorem. This shows that ϵ is exponentially small, proving the hardcore predicate property of g.

References

[Ari09] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on information Theory*, 55(7):3051–3073, 2009. 1

[Eli57] Peter Elias. List decoding for noisy channels. 1957. 4

[GL89] Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, 1989. 4, 4

[ZSW11] Shengmei Zhao, Peng Shi, and Bei Wang. Designs of bhattacharyya parameter in the construction of polar codes. In *2011 7th International conference on wireless communications, networking and mobile computing*, pages 1–4. IEEE, 2011. 1