6.897 Algorithmic Introduction to Coding Theory October 9, 2002

Lecture 10

Lecturer: Madhu Sudan *Scribe: Fu Liu*

1 Overview

- Algorithmic issues
- Algorithm for decoding Reed-Solomon codes

2 Algorithmic tasks

Broadly, it's relatively easier to give an encoding function and construction of a code than to find a decoding algorithm.

2.1 Encoding

If given a family of encoding functions $\{E_n | E_n : \{0,1\}^k \to \{0,1\}^n\}_n$, we would like to ask that how efficiently they can be computed.

It turns out that it is pretty hard when code is specified as part of the input. Fortunately, it works well for linear codes in polytime. Also, if the encoding problem is given as "given an circuit E, for the ith member of a family of a code, and a message m, compute its encoding." The this problem is trivial to solve in linear time in the size of E.

2.2 Construction of Code

- Code $\mathcal{C} \subseteq \{0,1\}^n$
- $|\mathcal{C}| = 2^k$
- Encoding circuit $E : \{0,1\}^k \to \mathcal{C}$.

  Circuit for E maybe specifictation of $\mathcal{C}$.
-

$$Decide(x) = \begin{cases} 1, & \text{if } x \in \mathcal{C} \\ 0, & \text{otherwise} \end{cases}$$

  Circuit for *Decide* is specification of $\mathcal{C}$.

2.3 Decoding

Now we come to the hard part. The problem can be stated informally as "if message c is sent, how can we compute c from the received message r? Also, we we consider the specification of the encoding function E, when/how is it specified" and "how much time it takes to design the decoding algorithm"... Unfortunately, these problems are so hard that we do not know any efficient algorithm. So we will have to focus on some specific codes.

2.3.1 Popular definitions in decoding

1. Maximum Likelihood Decoding (MLD)

   If we are given the channel, the code $\mathcal{C}$, and a received vector $r \in \Gamma^n$, find a codeword $c \in \mathcal{C}$, which

   $$\text{maximizes } \Pr_{\text{channel}} [r \text{ seen} \mid c \text{ sent}].$$

2. Nearest Codeword Problem (NCP)

   Given channel and the code $\mathcal{C}$, and received vector $r \in \Sigma^n$, find a codeword $c \in \mathcal{C}$, which

   $$\text{minimizes } \Delta(c, r).$$

   If we look back the MLD problem and give the channel as a q-ary symmetric channel, i.e. if $\Sigma = \{x_1, x_2, \ldots, x_q\}$, for $\forall i : 1 \leq i \leq q$, the probability of x_i seen when x_i sent is $1 - p$, and the probablity of $x_j (j \neq i)$ seen when x_i sent is $\frac{p}{1-q,}$ where $1 - p > \frac{1}{q}$ and $|\Sigma| = q$. Then this problem is just a NCP. This is a connection between MLD and NCP.

3. Soft-decision Decoding Problem

   Given the channel, the code $\mathcal{C}$ and a Σ by n matrix M, whose entries are nonnegative reals, find a $c \in \mathcal{C}$, that

   $$\text{minimizes} \Sigma_{i=1}^{n} M_{c_i,i}$$

   If M is given as a 0/1 matrix with one 1 in each column. Then it is the NCP problem.

   If Assume the given channel is i.i.d.. Then

   $$\begin{aligned} & \Pr[r \text{ seen} \mid c \text{ sent}] \\ = \; & \Pi_{i=1}^{n} \Pr[r_i \text{ seen} \mid c_i \text{ sent}] \\ = \; & \text{EXP}(\Pi_{i=1}^{n} \underbrace{-log \Pr[r_i \text{ seen} | c_i \text{ sent}]}_{M_{c_i,i}}) \end{aligned}$$

   By the equation above, soft-decision decoding solves MLD for i.i.d. channel.

2.3.2 Reasonable decoding problems

For a decoding problem, we should consider both the number of errors we're trying to correct and number of errors the code designed to handle. Let's look at the following three questions:

1. Unambiguous/unique decoding Given a code $\mathcal{C}$ with minimum distance d, and $r \in \Sigma^n$, find codeword c such that $\Delta(r, c) < \frac{d}{2}$ if it exists. Sometimes it is also called Bounded Distance Decoding.

2. Relatively Near Codeword (RNC) This is a parameterize problem with parameter $\gamma > 0$. Given a code $\mathcal{C}$ with minimum distance d, and $r \in \Sigma^n, e < \gamma d$, find $c \in \mathcal{C}$, such that $\Delta(r, c) \leq e$.

   Note when pick $\gamma = \frac{1}{2}$, RNC is Unambiguous Decoding Problem.

   When pick $\gamma = \infty$, RNC becomes NCP.

   And the problem is reasonable up to when $\gamma < 1$.

3. List-decoding problem The definition of this problem is similar to RNC except that here we want a list of all codewords c such that $\Delta(r, c) < \gamma d$.

   Therefore, Unambiguous decoding is List-decoding also, for $\gamma = \frac{1}{2}$.

   For each γ, RNC reduces to List-decoding Problem. And the List-decoding can be very hard if the list size is (potentialy) super polynomial. Hence, we should only attempt to solve this up to the list-decoding radius of a code, which is the radius of a ball with guarantee that the number of codewords in the ball is polynomial.

2.3.3 Easy problems for linear codes

For a linear code, because a generator matrix is given, the following three problem is "easy". Here "easy" means that it takes poly time.

- Encoding Clearly, code can be easily generated by the generator matrix.
- Error-detection We can get the parity matrix from the given generator matrix, and use it to detect whether errors occur.
- Erasure correction
  - Erasure-decoding problem: Given an k by n generator matrix G, the code $\mathcal{C}$ generated by G and vector $r \subseteq (\Sigma \cup \{?\})^n$, find $c \in \mathcal{C}$, such that $c_i = r_i$, for $i : 1 \leq i \leq n$ with $ri \neq ?$.
  - Decoding Algorithm: If there're s symbols of r are $?'s$, i.e. erased, let r' be a vector obtained by removing the $s?'s$ from r. And let G' obtained by removing the corresponding columns. Then solving the linear system $mG' = r'$ will give the solution $c = mG$.

    If $mG' = r'$ has no solution, they our problem does not have solution; if it has one solution, our problem have one solution; if it has more than one solution, they our problem has a solution space in the form of $Ax + b$.
  - Another fact about the erasure-decoding problem is that when the number of erasures, s, is smaller than d, the minimum distance of the code, then the solution is unique.

2.3.4 Syndrome Decoding

- Definition: Given a linear code $\mathcal{C}$ with parity check matrix H and $e \in F_q^n$, $e \cdot H$ is the *syndrome* of e.
- Note that if given a vector $r = c + e$, where $c \in \mathcal{C}$ and e is the error, the $r \cdot H = e \cdot H$ depends only on the error e but not on the codeword c.
- Syndrome decoding can be interpreted in two ways:
  - Given $e \cdot H$, compute e.
  - Build up a table of map $e \cdot H \to e$, and look up the table by index $e \cdot H$ to compute e. This is a brute-force algorithm.

3 Unambiguous Decoding of Reed-Solomon Codes

3.1 Problem statement

If given n distinct elements $\alpha_1, \alpha_2, \ldots, \alpha_n \in F_q, r_1, r_2, \ldots, r_n \in F_q$ and a parameter k, find a polynomial $p \in F_q[x]$ of degree no more than k, such that $p(\alpha_i) = r_i$ for $\underbrace{\text{many values}}_{> \frac{n+k}{2}}$ of i.

3.2 Convoluted history

At first glance, this problem is not trivial. Let's look at the history of this problem first.

- In 1958, BC + H discovered binary BCH¿
- Peterson 60' gave polytime decoding algorithm for binary BCH in 1960.
- Reed-Solomon found RS code, but did not notice the connection between it and BCH.
- In 1963, Gorenstein-Zierler discovered q-ary BCH codes and noticed that RS codes are special cases of q-ary BCH codes and that Peterson's algorithm generalizes to q-ary BCH codes.