(List-)
Decoding

CS591

Chun-Yun Hsiao
• Decoding: Find closest codeword to $Enc(m) + \eta$

• Linear Codes: Codewords form a vector space. $\exists$ generator matrix $G$ (or parity-check matrix $H$).

• Even decoding linear codes (given $G$ or $H$) is $NP$-hard.

• Read-Solomon code:
  
  – $msg = a$ polynomial $A(x)$ of degree $k - 1$ (where the coefficients are in $\mathbb{Z}_p = \{0, 1, \ldots, p - 1\}$)
  
  – $Enc(msg) = A(x_1), A(x_2), \ldots, A(x_n)$
  
  – $x_1, x_2, \ldots, x_n$ are fixed.

  – Distance of RS code is $n - k + 1$
• RS decoding:
  
  – Given: \( n \) pairs \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\), and \( k, e < \frac{n-k+1}{2} \)
  
  – Find: a polynomial \( A(x) \) of degree \(< k \) s.t. \( A(x_i) \neq y_i \) for at most \( e \) pairs

• The Welch-Berlekamp algorithm:
  
  – Error locating polynomial \( E(x) \):
    
    * \( E \) has degree \( e \)
    
    * \( E(x_i) = 0 \) if \( A(x_i) \neq y_i \)
  
  – Define \( N(x) \equiv A(x) \cdot E(x) \)
    
    * \( \forall i \in [n] \) \( N(x_i) = y_i \cdot E(x_i) \)
    
    * degree of \( N < k + e \)
\(- (N_0, N_1, \ldots, N_{k+e-1}) \) and \((E_0, E_1, \ldots, E_e)\) satisfy

\[
\sum_{j=0}^{k+e-1} N_j x_i^j = y_j \sum_{j=0}^{e} E_j x_i^j
\]

\(k + 2e + 1\) variables, \(n\) linear constraints, \(\exists\) solution if \(e < \frac{n-k+1}{2}\)

\(- \text{ Output } N(x)/E(x) \)

\(- O(n^3). \text{ Can be solved in } n \cdot poly(\log n) \)
• **List Decoding:**

  – Given: \( y = (y_1, y_2, \ldots, y_n) \) and \( t \)

  – Find: all codewords \( r \) that agree with \( y \) at \( \geq t \) positions \( (\triangle(r,y) \leq n - t) \)

• Johnson bound: there are “not too many” codewords

• List decoding for RS codes:

  – Given: \( n \) pairs \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\) and \( k, t \)

  – Find: a list of *all* polynomials \( A(x) \) of degree at most \( k \) s.t. \( A(x_i) = y_i \) for at least \( t \) pairs
• Outputing two polynomials:

\[(y - A_1(x))(y - A_2(x)) = 0\]

1. \(\forall i \in [n], y_i^2 - B(x_i)y_i + C(x_i) = 0\)

2. \(B(x) = A_1(x) + A_2(x)\)
   \(C(x) = A_1(x) \cdot A_2(x)\)

3. \(\deg(B) \leq k, \deg(C) \leq 2k\)

**Theorem.** \(B(x)\) and \(C(x)\) exist and if \(A(x)\) is any degree \(k\) polynomial s.t. \(A(x_i) = y_i\) for more than \(t \geq 2k + 1\) pairs, then
\(y - A(x) | y^2 - B(x) + C(x)\)

**Fact.** Factoring polynomial (in finite fields) can be done efficiently
• Generallization:

1. Find: $Q(x, y)$ of degree $l$ in $y$, and degree $D$ in $x$ s.t.
   
   $- Q(x_i, y_i) = 0, \forall i \in [n]$

   $- Q \neq 0$

2. Factor $Q(x, y)$ into $\prod_j(y - A_j(x))$

**Theorem.** Set $D = \sqrt{nk}$, $l = \sqrt{n/k}$. If $t > 2\sqrt{nk}$, the above is a RS-list-decoder.
• List-decoding Applications:

  – Complexity theory: Hardcore bit for one-way functions. The original construction [Goldreich, Levin] is a (implicit) list-decoding for Hadmard code. Better result can be achieved by RS-list-decoding.

  – Amplifying hardness of Boolean functions: Derandomizing \( \mathcal{BPP} \)

  – Extractors

  – Predicting witnesses for \( \mathcal{NP} \)-search problems

  – Direct product of \( \mathcal{NP} \)-complete languages

  – Permanent of random matrices