Decoding Reed-Solomon codes beyond $d/2$ using Sudan's Algorithm

Emanuele Viterbo

Dipartimento di Elettronica
Politecnico di Torino
Torino, Italy

Tutorial presentation

Introduction

- Reed-Solomon codes (1960) are among the most succesful forward error correcting codes in current applications.
- An algebraic decoding algorithm with polynomial complexity is available to decode up to packing radius of the code $d/2$. (Peterson, Berlekamp-Massey)
- Sudan (1997,1999) recently proposed a polynomial complexity algorithm capable of correcting up to $n - \sqrt{n(k-1)}$ errors.
- Current research is headed towards algebraic soft-decoding of RS codes to achieve significant coding gains.

Basic definitions

- $\mathbb{F}_q$ *finite field* of order q
- $\mathbb{F}_q^*$ non zero elements of finite field
- $\mathbb{F}_q[X]$ ring of polynomials in X with coefficients over $\mathbb{F}_q$
- *Reed-Solomon codes* can be defined as *evaluation codes*:

  given a set $\mathcal{P} = \{x_1, \dots x_n\} \subseteq \mathbb{F}_q$

  $RS_q(n,k) = \left\{\mathbf{x} = (f(x_1), \dots, f(x_n)), \quad f(X) \in \mathbb{F}_q[X], \quad \deg f < k\right\}$
- Usually $\mathcal{P} = \mathbb{F}_q^*$ and $q = 2^8$ and shortening is applyed.

Properties of Reed-Solomon codes

- The set of *information polynomials*

  $$\left\{f(X) \in \mathbb{F}_q[X], \quad \deg f < k\right\}$$

  forms a vector space hence the code is linear.
- Minimum distance (= minimum weight codeword)

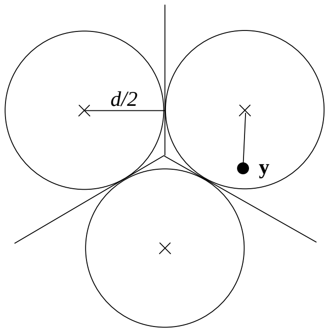
  $$d = n - k + 1$$

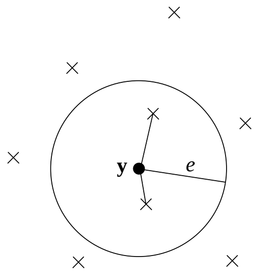
  since $f(X) = 0$ in at most $k-1$ positions.

Hard decision decoding

- Assuming bitwise transmission over a memoryless BSC or $\mathbb{F}_q$-symbol transmission over q-ary symmetric channel, the receved word is
$$\mathbf{y} = \mathbf{x} + \mathbf{e}$$
where $\mathbf{x}$ is the codeword and $\mathbf{e}$ is the error vector
- *Maximum-likelihood decoding* requires to find
$$\min_{\mathbf{x}} \; d_H(\mathbf{y}, \mathbf{x})$$
- The Berlekamp-Massey algorithm is a *bounded distance decoder* and finds the above minimum under the constraint
$$e = w_H(\mathbf{e}) = d_H(\mathbf{y}, \mathbf{x}) < d/2$$
otherwise it outputs a *decoding failure.*

Decoding beyond $d/2$

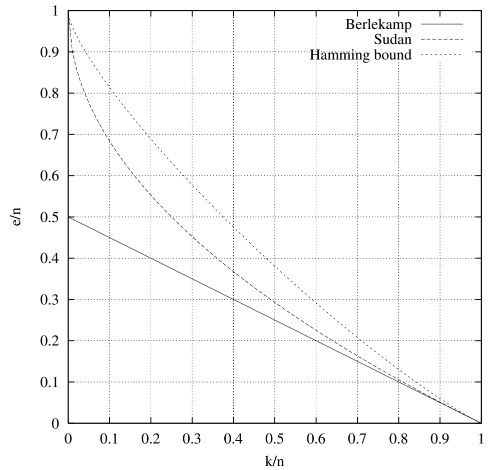




Decoding beyond $d/2$

- Sudan's algorithm "corrects" errors beyond $d/2$.
- If $e < d/2$ the closest codeword is unique.
- If we let $e \geq d/2$ we use a different strategy: the so called *list decoding* which outputs a list of all the codewords within a Hamming ball of radius e centered at the received word.
- The final decision is selected from the list using soft information to resolve *ex-aequo* cases.
- Further processing from the list can also be performed in concatenated schemes.

Error correcting capacity



The Polynomial reconstruction problem

- **INPUT:**
  k, s and $n(\geq s)$ points $\mathcal{P} = \{(x_p, y_p)\}_{p=1}^{n}$ where $x_p, y_p \in \mathbb{F}$
- **OUTPUT:**
  all univariate polynomials f with $\deg f < k$ such that

$$y_p = f(x_p) \quad \text{for at least } s \text{ values of } p \in \{1, \ldots, n\}$$

The above curve fitting algorithm is equivalent to list decoding of Reed-Solomon codes if we let $e = w_H(\mathbf{e}) = n - s$.

The Berlekamp-Massey RS decoder

Given the received word $\mathbf{y} = \mathbf{x} + \mathbf{e} = (y_1, \ldots, y_n)$ let $\mathcal{P} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

- If $\mathbf{e} = 0$ then $y_p = x_p = f(x_p)$ for a unique $f(X) \in \mathbb{F}_q[X]$ with $\deg f < k$.
  All points lie on the curve $Q(X,Y) = Y - f(X) = 0$
- If $\mathbf{e} \neq 0$ and $\mathbf{e} \neq \mathbf{x}$ then $Q(X,Y)$ does not have the form $Y - f(X)$ with $\deg f < k$.

We define the *error locator polynomial* $\Lambda(X,Y)$ such that $\Lambda(x_p, y_p) = 0$ when $e_i \neq 0$

- If $w_H(\mathbf{e}) < d/2$ it is possible to write

$$Q(X,Y) = \Lambda(X,Y)(Y - f(X)) \qquad \text{with} \qquad \deg f < k$$

  passing through all the points in $\mathcal{P}$.
- In this case f gives the ML transmitted codeword $\mathbf{x}$.

The key idea of Sudan

- **STEP 1:** *Interpolation*
  Find $Q(X,Y)$ of sufficiently high degree passing through all points in $\mathcal{P}$ with some multiplicity m.
- **STEP 2:** *Factorization*
  Find all polynomials $f_i(X)$ with $\deg f_i < k$ such that $Q(X, f_i(X)) = 0$ or equivalently

$$Q(X,Y) = \Lambda(X,Y) \prod_i (Y - f_i(X))$$

Some more definition

- Let $A(X,Y) = \sum_i \sum_j a_{ij} X^i Y^j$ and $w_X, w_Y \geq 0$ the (w_X, w_Y) *weighted degree* of A is

$$\deg_{(w_X, w_Y)}(A) = \max(i w_X + j w_Y) \quad \text{such that } \; a_{ij} \neq 0$$

- The number of monomials of (w_X, w_Y) weighted degree $\leq \delta$ is

$$N_{w_X, w_Y}(\delta) = |\{X^i Y^j : i, j \geq 0 \text{ and } i w_X + j w_Y \leq \delta\}|$$

- We say that a point (α, β) *lies on* $A(X,Y)$ if $A(\alpha, \beta) = 0$
- We say that $A(X,Y)$ *passes through* a point (α, β) with multiplicity m if $A(X + \alpha, Y + \beta)$ has lowest degree monomial of degree m.

Relation between $A(X,Y)$ and $A(X+\alpha,Y+\beta)$

Let $B(X,Y) = A(X+\alpha, Y+\beta) = \sum_h \sum_\ell b_{h\ell} X^h Y^\ell$ then

$$\begin{aligned} A(X+\alpha, Y+\beta) &= \sum_i \sum_j a_{ij} (X+\alpha)^i (Y+\beta)^j = \\ &= \sum_i \sum_j a_{ij} \sum_{h=0}^{i} \binom{i}{h} X^h \alpha^{i-h} \sum_{\ell=0}^{j} \binom{j}{\ell} Y^\ell \beta^{\ell-j} \end{aligned}$$

using the following identities

$$\begin{aligned} \sum_{i=0}^{\infty} \sum_{h=0}^{i} \binom{i}{h} \alpha^{i-h} X^h &= \sum_{h=0}^{\infty} \sum_{i=h}^{\infty} \binom{i}{h} \alpha^{i-h} X^h \\ \sum_{j=0}^{\infty} \sum_{\ell=0}^{j} \binom{j}{\ell} \beta^{j-\ell} Y^\ell &= \sum_{j=0}^{\infty} \sum_{j=\ell}^{\infty} \binom{j}{\ell} \beta^{j-\ell} Y^\ell \end{aligned}$$

We get

$$b_{h\ell} = \sum_{i=h}^{\infty} \sum_{j=\ell}^{\infty} a_{ij} \binom{i}{h} \binom{j}{\ell} \alpha^{i-h} \beta^{j-\ell}$$

Interpolation

- $A(X,Y)$ passes through (α,β) with multiplicity m iff

$$b_{h\ell} = 0 \quad \text{for} \quad 0 \le h + \ell < m$$

- This imposes $\frac{m(m+1)}{2} = \binom{m+1}{2}$ linear constraints on the coefficients a_{ij}.
- **Example** – Let $A(X,Y) = a_{10}X + a_{01}Y + a_{11}XY$

$A(X+\alpha, Y+\beta) = (a_{10}\alpha + a_{01}\beta + a_{11}\alpha\beta) + (a_{10} + a_{11}\beta)X + (a_{01} + a_{11}\alpha)Y + a_{11}XY$

  - $A(X,Y)$ passes through (α,β) with multiplicity $m = 1$ iff

$$a_{10}\alpha + a_{01}\beta + a_{11}\alpha\beta = 0$$

  - $A(X,Y)$ passes through (α,β) with multiplicity $m = 2$ iff

$$\begin{cases} a_{10}\alpha + a_{01}\beta + a_{11}\alpha\beta = 0 \\ a_{10} + \qquad + a_{11}\beta = 0 \\ \qquad a_{01} + a_{11}\alpha = 0 \end{cases}$$

The polynomial reconstruction algorithm

- **INPUT:** n, k, s and the set of points $\mathcal{P} = \{(x_p, y_p)\}_{p=1}^{n}$ where $x_p, y_p \in \mathbb{F}$.
- **STEP 0:** compute m and δ such that

$$ms > \delta \quad \text{and} \quad N_{1,k-1}(\delta) > n\binom{m+1}{2}$$

- **STEP 1:** find $Q(X,Y)$ such that: $\deg_{(1,k-1)}(Q) \le \delta$ i.e., find the coefficients $\{q_{ij}\}$ with $i,j \ge 0$ and $i + (k-1)j \le \delta$ such that
  1. at least one $q_{ij} \neq 0$ (non trivial solution)
  2. the coefficients $q_{h\ell}^{(p)}$ of $Q^{(p)}(X,Y) = Q(X+x_p, Y+y_p)$ are zero for $0 \le h+\ell < m$

$$q_{h\ell}^{(p)} = \sum_{i=h}^{\delta-(k-1)\ell} \sum_{j=\ell}^{\lfloor \frac{\delta-i}{k-1} \rfloor} q_{ij} \binom{i}{h} \binom{j}{\ell} x_p^{i-h} y_p^{j-\ell} = 0 \qquad \begin{cases} p = 1 \dots n \\ 0 \le h + \ell < m \end{cases}$$

The polynomial reconstruction algorithm (cont.)

- **STEP 2:**
  1. Find all polynomials $f \in \mathbb{F}_q[X]$ with $\deg f < k$ such that $(Y - f(X))$ is a factor of $Q(X,Y)$
  2. Check if $y_p = f(x_p)$ for at least s points out of n
  3. If so add to the list
- **END**

Correctness of the Algorithm (factorization)

Lemma 1 – If (x_p, y_p) is an input point and f a polynomial such that $y_p = f(x_p)$, then $(x - x_p)^m | g(X) = Q(X, f(X))$.

Proof.

- Let $f'(X) = f(X - x_p) - y_i$
- Since $f'(0) = 0$ we have $f'(X) = Xf''(X)$ for some polynomial $f''(X)$
- Let $g'(X) = Q^{(p)}(X, f'(X))$ then $g(X) = g'(X - x_p)$, in fact

$$g(X) = Q^{(p)}(X - x_p, f(X) - y_p) = Q^{(p)}(X - x_p, f'(X - x_p)) = g'(X - x_p) \qquad [*]$$

- $Q^{(p)}$ has no monomials of degree less than m
- If we substitute $Y = Xf''(X)$ in $Q^{(p)}(X, Y)$ to get $g'(X)$ we see that $x^m | g'(X)$
- Shifting back we have $(x - x_p)^m | g'(X - x_p)$ and for $[*]$ we have $(x - x_p)^m | g(X)$

Dip. Elettronica
Politecnico di Torino

Correctness of the Algorithm (factorization)

Lemma 2 – If $f(X)$ is a polynomial with $\deg f < k$ and $y_p = f(x_p)$ for at least s values of p in $\{1, \ldots, n\}$ and $ms > \delta$ then $Y - g(X)$ divides $Q(X, Y)$.

Proof.

- Since $g(X) = Q(X, f(X))$ and $\deg_{(1,k-1)}(Q) \le \delta$ then $\deg g \le \delta$
- By Lemma 1 for all $p \in S = \{p | y_p = f(x_p)\}$ we have $(x - x_p)^m | g(X)$ then

$$\prod_{p \in S} (x - x_p)^m | g(X) = Q(X, f(X)) \qquad [*]$$

- Since $|S| \ge s$ we have $\deg(\prod_{p \in S} (x - x_p)^m) \ge ms > \delta = \deg g(X)$
- Relation $[*]$ is possible only if $g = 0$ thus $f(X)$ is a root of $Q(X, Y)$ as a polynomial in Y with coefficients in $\mathbb{F}_q[X]$. Alternatively $Y - g(X)$ divides $Q(X, Y)$.

Dip. Elettronica
Politecnico di Torino

Correctness of the Algorithm (interpolation)

Lemma 3 – If $N_{1,k-1}(\delta) > n\binom{m+1}{2}$ then the polynomial $Q(X, Y)$ of STEP 1 exists and can be found in polynomial time by solving a linear system.

Proof.

- $n\binom{m+1}{2}$ is number of linear equations in the $N_{1,k-1}(\delta)$ unknowns q_{ij}
- The homogeneous linear system has non-trivial solution giving the coefficients of $Q(X, Y)$
- Let us compute explicitly the number of unknowns knowing that $\ell + (k-1)h \le \delta$

$$\begin{aligned} N_{1,k-1}(\delta) &= \sum_{h=0}^{\lfloor \delta/(k-1) \rfloor} \sum_{\ell=0}^{\delta-(k-1)h} 1 = \sum_{h=0}^{\lfloor \delta/(k-1) \rfloor} (\delta + 1 - (k-1)h) = \\ &= (\delta+1)\left(\left\lfloor \frac{\delta}{k-1} \right\rfloor + 1\right) - \frac{k-1}{2} \left\lfloor \frac{\delta}{k-1} \right\rfloor \left(\left\lfloor \frac{\delta}{k-1} \right\rfloor + 1\right) = \\ &= \left(\left\lfloor \frac{\delta}{k-1} \right\rfloor + 1\right)\left(\delta + 1 - \frac{k-1}{2} \left\lfloor \frac{\delta}{k-1} \right\rfloor\right) \ge \frac{\delta(\delta+2)}{2(k-1)} \end{aligned}$$

Dip. Elettronica
Politecnico di Torino

Error correcting capability

Lemma 4 – If $e < n - \sqrt{n(k-1)}$ (or equivalently $s^2 > n(k-1)$) then the conditions imposed by STEP 0 are satified for

$$\delta = ms - 1 \quad \text{and} \quad m = 1 + \left\lfloor \frac{n(k-1) + \sqrt{n^2(k-1)^2 + 4[s^2 - n(k-1)]}}{2[s^2 - n(k-1)]} \right\rfloor$$

Proof.

- Since we impose $ms > \delta$ we can choose $\delta = ms - 1$ and replace it in the second constraint

$$n\binom{m+1}{2} < \frac{\delta(\delta+2)}{2(k-1)} = \frac{(ms-1)(ms+1)}{2(k-1)}$$

- This gives the quadratic inequality $m^2[s^2 - n(k-1)] - mn(k-1) - 1 > 0$ which is satisfied for the above m.

Dip. Elettronica
Politecnico di Torino

Length of the list

Proposition – The number of codewords M that lie within a Hamming ball of radius $e < n - \sqrt{n(k-1)}$ of an (n,k) RS code is $O(\sqrt{n^3(k-1)})$.
Proof.

- By Lemma 2 $M < \deg_Y(Q)$ the degree in Y of $Q(X,Y)$.
- Since $\deg_{(1,k-1)}(Q) \le \delta$ then $\deg_Y(Q) \le \lfloor \delta/(k-1) \rfloor$
- Choosing $s^2 = n(k-1)+1$ or $e = n - \lfloor \sqrt{n(k-1)} \rfloor - 1$ (max radius) we find from Lemma 4 $m = O(n(k-1))$ and

$$M = O\left(\frac{\delta}{k-1}\right) = O\left(\frac{n(k-1)s}{k-1}\right) = O\left(\sqrt{n^3(k-1)}\right)$$

Complexity

- **Proposition** – Sudan's algorithm can be implemented to run using
$$O\left(\max\left\{\frac{n^6(k-1)^3 s^6}{[s^2 - n(k-1)]^6}, \frac{s^6}{(k-1)^3}\right\}\right)$$
field operations over $\mathbb{F}_q$, provided that $q \le 2^n$.
- **Corollary** – The polynomial reconstruction problem can be solved in time $O(n^{15})$, provided $s > \sqrt{n(k-1)}$, for any field $\mathbb{F}_q$ with $q < 2^n$.
Furthermore if $s^2 = (1+\gamma)n(k-1)$, then the problem can be solved in $O(n^3/\gamma^6)$.
- **Corollary** – RS codes of rate k/n can be list decoded in $O(n^{15})$ if $e = w_H(\mathbf{e}) = n - \lfloor \sqrt{n(k-1)} \rfloor - 1$.
The decoding time reduces to $O(n^3)$ if $e = w_H(\mathbf{e}) < n - \lfloor \sqrt{n(k-1)} \rfloor - 1$.

Conclusions

Sudan's algorithm can be adapted to

- decode algebraic-geometry codes beyond the packing radius.
- handle the combination of errors and erasures
- soft-decision decoding using the weighted version of the curve fitting problem

Bibliography

- M. Sudan: "Decoding Reed-Solomon codes beyond the error correcting diameter," *Proc. Allerton Conference*, 1997.
- M. Sudan: "Decoding Reed-Solomon codes beyond the error correcting bound," *J. Complexity*, vol. 13, no. 1, pp. 180–193, March 1997.
- V. Guruswami, M. Sudan: "Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, Sept. 1999.