30 Jan

List Decoding

We have seen unique decoding: i.e. for each received vector y, we want to return a single codeword as the estimate of the transfered codeword.

Flipped picture

What if we want unique decoding in the ball of radius t around any received vector y, $\exists$ only one $c \in \mathcal{C}$. Then how large can t be?

$\rightarrow t \leq \frac{d-1}{2}$

What generalization is possible? RS basically "solves" everything (except for the field size issue).

What if we don't need unique decoding?

Lets define $B_e(y)$ to be the ball of radius e around y.

Notice $|B_e(y) \cap \mathcal{C}| \leq O(q^n) \quad \forall e$ (trivial)

Also for unique decoding, $|B_e(y) \cap \mathcal{C}| = 1$ for $e \leq \frac{d-1}{2}$

For what values of e can $|B_e(y) \cap \mathcal{C}| = \text{poly}(n)$?

Potential Application: Transmitted codeword is one among the "list" of potential words in the ball.

i) Now among this list you can look for the transmitted code. The search space is now polynomial. One way is for the transmitter to retransmit the index of the correct transmitted vector. This needs extra $\underset{\text{size of list}}{\underset{\downarrow}{\log L}}$ information.

Coded nearest neighbour?

<u>Definition</u>: Let $0 < \rho < 1$, and L be some +ve integer. A code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is said to be (ρ, L) list decodable if for every received word $y \in \Sigma^n$,

$$\left| \{ c \in \mathcal{C} \mid d(y,c) \leq \rho n \} \right| \leq L$$

Johnson Bound

If $\rho < J_q\left(\frac{d}{n}\right)$, then $\mathcal{C}$ is (ρ, qdn) decodable.

where $J_q(\delta) \triangleq \left(1 - \frac{1}{q}\right)\left(1 - \sqrt{1 - \frac{q\delta}{q-1}}\right)$

<u>Note</u>: $d \leq n-k+1$

$\Rightarrow L = qdn \leq qn^2 = O(n^2)$ if q is constant.

$\downarrow$ polynomial

<u>Proof Outline</u> -

Assume $|B_e(y) \cap \mathcal{C}| = M$. Suppose $q = 2$,

Show: $M \leq 2dn$, for $e = \rho n$ where $\rho < \frac{1}{2}\left(1 - \sqrt{1 - \frac{2d}{n}}\right)$

Let $B_e(y) \cap \mathcal{C} = \{c_1, c_2, \ldots, c_M\}$

We know, $d(c_i, c_j) \geq d$

Also let $c_i' = c_i - y$

Notice $w_H(c_i') = w_H(c_i - y) \leq e$

Also $d_H(c_i', c_j') = d_H(c_i, c_j) \geq d$

Let $S = \sum_{\substack{i,j \\ i \neq j}} d_H(c_i', c_j') \geq \binom{M}{2} d$

Lets get an upperbound for S.

Consider a Matrix - $\left[\begin{array}{c|c|c|c} c_1' & c_2' & \cdots & c_M' \end{array}\right]_{n \times M}$

Suppose row 1 has m_1 non-zeros. $\Rightarrow$ There are $(M - m_1)$ zeros. Only these contribute to the sum S.

$\Rightarrow$ row contributes $m_1(M - m_1)$ to S

In general, row i contributes $m_i(M - m_i)$.

$$\Rightarrow S = \sum_i m_i(M - m_i) = M \sum_i m_i - \sum_i m_i^2$$

For an upper bound for S, upper bound $M \sum_i m_i$ and lower bound $\sum_i m_i^2$

Let $\bar{e} \triangleq \frac{\sum_{i=1}^{n} m_i}{M} = \frac{\text{Total weight of all } M \text{ vectors}}{M}$ (Average weight of vector)

Take $a \triangleq (m_1, \dots, m_n)$ (some real vector of length n)
$b \triangleq \left(\frac{1}{n}, \dots, \frac{1}{n}\right)$

Now $\langle a, b \rangle^2 \leq \|a\|^2 \|b\|^2$ by Cauchy Schwarz.

$$\Rightarrow \left(\sum_i \frac{m_i}{n}\right)^2 \leq \left(\sum_i m_i^2\right)\left(\sum_i \frac{1}{n^2}\right) = \frac{1}{n}\sum_i m_i^2$$

$$\Rightarrow \sum_i m_i^2 \geq \frac{\left(\sum_i m_i\right)^2}{n} = \frac{(M\bar{e})^2}{n}$$

$$\Rightarrow S = M\left(\sum_i m_i\right) - \sum_i m_i^2$$

$$\leq M(M\bar{e}) - \frac{M^2\bar{e}^2}{n}$$

$$\leq M^2\left(\bar{e} - \frac{\bar{e}^2}{n}\right)$$

$$\Rightarrow \binom{M}{2} d \leq S \leq M^2\left(\bar{e} - \frac{\bar{e}^2}{n}\right)$$

$$\Rightarrow \frac{(M-1)}{2} d \leq M\left(\bar{e} - \frac{\bar{e}^2}{n}\right)$$

$$\Rightarrow M\left(\frac{d}{2} - \bar{e} + \frac{\bar{e}^2}{n}\right) \leq \frac{d}{2}$$

$$\Rightarrow M\left(\frac{dn - 2\bar{e}n + 2\bar{e}^2}{n}\right) \leq d$$

$$\Rightarrow M \leq \frac{dn}{dn - 2\bar{e}n + 2\bar{e}^2}$$

Multiplying and dividing by 2,

$$= \frac{2dn}{2dn - 4\bar{e}n + 4\bar{e}^2}$$

Completing the square,

$$= \frac{2dn}{2dn + n^2 - n^2 - 4\bar{e}n + 4\bar{e}^2}$$

$$\Rightarrow M \leq \frac{2dn}{-n(n-2d) + (n-2\bar{e})^2}$$

We want M to be poly in n. Suppose $M \leq 2dn$

Then, $(n-2\bar{e})^2 - n(n-2d) \geq 1$

We know $\bar{e} \leq e$

$$\Rightarrow (n-2e)^2 - n(n-2d) \geq 1$$

$$\Rightarrow (n-2e)^2 \geq 1 + n(n-2d)$$

$$\Rightarrow (n-2e) \geq \sqrt{1 + n(n-2d)}$$

$$\Rightarrow n - 2e > \sqrt{n(n-2d)}$$

$$\Rightarrow e < \frac{1}{2}\left(1 - \sqrt{1 - \frac{2d}{n}}\right)$$

Feb 6

Every code $\mathcal{C}$ with $d = d$ is $\left(\frac{\frac{d-1}{2}}{n}, 1\right)$ list decodable.

Recall: (ρ, L) - list decodable $\Rightarrow$ There are atmost L codewords in a ball of radius ρn

As $q \to \infty$, Johnson's bound gives -

$$\lim_{q \to \infty} \rho < \left(1 - \sqrt{1 - \frac{d}{n}}\right) = 1 - \sqrt{\frac{n-d}{n}}$$

Fraction of correctable errors

(singleton)

Now $d \leq n-k+1$

$\Rightarrow n-d \geq k-1$

$$\approx 1 - \sqrt{\frac{k-1}{n}}$$

ρn = radius = e

$\Rightarrow \rho = \frac{e}{n}$

$$\approx 1 - \sqrt{R}$$ where R is the rate.

Notion of 'correct' in (ρ, L) list decoding: If $d_H(c, y) \leq \rho n$ (c: tx codeword, y: rx vector)

Then c should belong to the list of decoded o/p vectors.

Welch - Berlekamp -

Want to generalize W-B to List Decoding.

$\rho = 1 - 2\sqrt{R} \Rightarrow$ List size is polynomial. (by Johnson Bound).

This is worse than what Johnson's bound allows

A way to ensure correctness is by making sure that all possible codewords which are at distance $\leq \rho n$ from y should be in the list.

Recasting B-W algorithm

Polynomial Interpolation

1. Define $Q(x,y) = yE(x) - N(x)$.
   where $\deg(E(x)) = t$
   $\deg(N(x)) = k+t-1$

Step 1 of B-W $\Rightarrow$ finding Q such that $Q(\alpha_i, y_i) = 0$.
for each $i = 1, \ldots, n$.

Factorization

2. Find a polynomial of the form $(y - \hat{M}(x))$ such that
   a) $\deg(\hat{M}(x)) \leq k-1$ $\rightarrow$ message length is k.
   b) $(y - \hat{M}(x)) \mid Q(x,y)$
   c) $d_H(y, (\hat{M}(\alpha_1), \ldots, \hat{M}(\alpha_n))) \leq t$. $\rightarrow$ codeword should be in ball.

Intuition for (b)

suppose $\hat{M}(x) = \dfrac{N(x)}{E(x)}$, then $\hat{M}(x)E(x) - N(x) = 0$.

consider $Q(x,y) = yE(x) - N(x)$.
For the above condition to hold, $\hat{M}(x)$ must be a root.
$\Rightarrow (y - \hat{M}(x))$ must be a factor.

Alternatively if $(y - \hat{M}(x)) \mid yE(x) - N(x)$

$\Rightarrow (y - \hat{M}(x)) \cdot q(x) = yE(x) - N(x)$

$\Rightarrow yq(x) - \hat{M}(x)q(x) = yE(x) - N(x)$

$$\Rightarrow \hat{M}(x) = \frac{y(q(x) - E(x)) + N(x)}{q(x)}$$

For $q(x) = E(x)$, $\hat{M}(x) = \dfrac{N(x)}{E(x)}$.

Generalizing for List Decoding

1. Interpolate $Q(X,Y)$ given $Q(\alpha_i, y_i) = 0$, $i = 1, \ldots, n$.

2. Find $\boxed{\text{ALL}}$ $\hat{M}(x)$ which satisfy –
   a) $\deg(\hat{M}(x)) \le k-1$
   b) $(y - \hat{M}(x)) \mid Q(x,Y)$
   c) $d_H(y, (M(\alpha_1), \ldots, M(\alpha_n))) \le \delta n$

Notice finding all such $\hat{m}(x)$ ensures that the original message appears because $M(X)$ will satisfy all the above conditions.

When does step 1 work?

The polynomial is of the form-

$$Q(X,Y) = \sum_{i=0}^{a} \sum_{j=0}^{b} q_{ij} x^i y^j$$

$$\Rightarrow Q(\alpha, y) = \sum_{i=0}^{a} \sum_{j=0}^{b} q_{ij} \alpha^i y^j = 0.$$

Thus we have n equations and $\underbrace{(a+1)(b+1)}_{r}$ unknowns.

In matrix form –

$$\underset{n \times r}{\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix}} \underset{r \times 1}{\begin{bmatrix} q_{00} \\ \vdots \\ q_{ab} \end{bmatrix}} = \underset{n \times 1}{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}$$

When will there be a non-zero solution?

$\rightarrow$ when $n \le r$.

If we don't want a unique solution, $n < r$.

$\Rightarrow \quad n < (a+1)(b+1)$.

i.e. $(\deg_x(Q)+1)(\deg_y(Q)+1) > n$.

Suppose $\deg_x(Q) = l$ and $\deg_y(Q) = \left\lceil \frac{n}{l} \right\rceil$

This satisfies the above condition. Now we want to find another condition for l so that the conditions for step 2 holds.

i.e. $(Y - \hat{M}(X)) \mid Q(X,Y)$

Define $R(X) = Q(X, M(X))$

Then $(Y - M(X)) \mid Q(X,Y)$ iff $R(X) = 0$.
(since $M(X)$ is a root of Q).

Notice, $R(X) = \sum_i^l \sum_j^{n/l} q_{ij} X^j (\underbrace{M(X)}_{\deg = k-1})^j$

$\therefore \deg(R(X)) \leq l + (k-1)\frac{n}{l}$

At every agreeing position, $y_i = M(\alpha_i)$, evaluating R,
(there are at least $n-e$)

$R(\alpha) = \sum_i^l \sum_j^{n/l} q_{ij} \alpha^i y^j = Q(\alpha, y) = 0$.
(by definition)

$\Rightarrow$ R has at least $n-e$ distinct roots.

$\Rightarrow R(X) = 0$ if $n-e > l + (k-1)\frac{n}{l}$

For the tightest bound, differentiating RHS we get,

$$1 - \frac{(k-1)n}{l^2} = 0$$

$$\Rightarrow l^2 = (k-1)n$$

$$l = \sqrt{n(k-1)}$$

$$\Rightarrow n-e > 2\sqrt{n(k-1)}$$

$$\Rightarrow e < n - 2\sqrt{n(k-1)}$$

$$\Rightarrow f = \frac{e}{n} = 1 - 2\sqrt{\frac{(k-1)}{n}} = 1 - 2\sqrt{R}$$

(10th and 13th cancelled due to quiz)

Feb 17

Johnson's bound allows us $1-\sqrt{R}$ but our algorithm gives only $1-2\sqrt{R}$.

Can we do better? Turns out we can do $1-\sqrt{2R}$.

Recall decoding is correct when the list contains the transmitted codeword.

This is guaranteed to happen when all possible codewords that are at distance $\leq \rho n$ from y are in the list.

Essential Idea: Define $Q(x,y)$ more intelligently.

Observation: To prove that every $M(X)$ with $\deg(M(X)) \leq k-1$ and $d_H(y, (M(\alpha_1) \cdots M(\alpha_n))) \leq e = \rho n$

we used a degree argument on $R(X) = Q(X, M(X))$.

Recall $\deg(R(x)) \leq \deg_x(Q) + (k-1)\deg_y(Q)$

For the new algorithm, assume a different structure for $Q(X,Y)$ such that $D = \deg(R(X))$

$= \max\{(i+(k-1)j) \mid x^i y^j$ exists in Q with non zero coefficient$\}$

is smaller than # roots of $R(X) = n-e$.

i.e. $n-e > D$ (so that R is zero polynomial)

$\Rightarrow e < n-D$

Smaller the D, larger e. However D is restricted by Q since to interpolate it the product of degree wrt X and Y should be more than n.

∴ Goal — Choose Q so that

a) $\frac{e}{n} < \frac{n-D}{n} = 1 - \sqrt{2R}$

b) # coeffs of $Q(X,Y) > n$.

Algorithm

1. Define $Q(X,Y) = \sum_{\substack{i,j \\ i+(k-1)j \leq D}} q_{ij} X^i Y^j$

Now $\deg(R(X)) \leq D$ by definition.
We will fix the value of D later when it becomes apparent

We have to still make sure that # coeffs $> n$.

Observations —

i) For $i + (k-1)j \leq D$, $i \geq 0$, $j \geq 0$

Then $j \leq \left\lfloor \frac{D}{(k-1)} \right\rfloor$

Let $l = \left\lfloor \frac{D}{(k-1)} \right\rfloor$

$$\Rightarrow \sum_{j=0}^{l} \sum_{i=0}^{D-(k-1)j} 1 = \text{\# coefficients.}$$

$$= \sum_{j=0}^{l} D - (k-1)j + 1$$

$$D = \frac{D}{(k-1)} \cdot (k-1)$$

$$\Rightarrow D \geq \left\lfloor \frac{D}{k-1} \right\rfloor (k-1)$$

$$= (l+1)(D+1) - (k-1)\sum_{j=0}^{l} j$$

$$= (l+1)(D+1) - (k-1)\frac{l(l+1)}{2}$$

$$= \frac{(l+1)}{2}\Big[2D + 2 - \underbrace{(k-1)l}_{\leq D}\Big]$$

$$\geq \frac{(l+1)}{2}\Big[2D + 2 - D\Big]$$

Also $l+1 > \frac{D}{k+1}$

$$\Rightarrow \quad \geq \frac{D}{2(k+1)}[D+2]$$

$\Rightarrow$ # coeffs $\geq \dfrac{D(D+2)}{2(k-1)}$

For interpolation we need $\dfrac{D(D+2)}{2(k-1)} > n$.

$\therefore$ pick $D = \sqrt{2n(k-1)}$

2. Find all $\hat{m}(x)$ satisfying the 3 conditions.

Show $\deg(R(x)) < n-e$.

Now $n - e > D = \sqrt{2n(k-1)}$

$\Rightarrow e < n - \sqrt{2n(k-1)}$

$\Rightarrow \rho < 1 - \sqrt{\dfrac{2(k-1)}{n}} = 1 - \sqrt{2R}$

Class 10
(Recorded)

Recall Johnson's bound guarantees that polynomial sized list upto a radius of $1-\sqrt{R}$.
But our previous two efficient algorithms didn't reach this.

Algo 2 Recap

1) Interpolation: Find $Q(X,Y)$ s.t. $Q(\alpha_i, y_i) = 0$, $i=1,\dots,n$
(To find this we imposed some degree constraints).

2) Factorization: Include all $\hat{M}(x)$ that satisfy –
i) $\deg(\hat{M}(x)) \leq k-1$
ii) $(y - \hat{M}(x)) \mid Q(X,Y)$
iii) $d_H((\hat{M}(\alpha_1), \dots, \hat{M}(\alpha_n)), y) \leq e = \delta n$

Checking correctness

Correct $\Rightarrow$ transmitted message polynomial M is within the outputted list. (given that y is at distance at most e)

We can ensure this by seeing if (i) and (iii) are satisfied then (ii) is also satisfied.
i.e., for any $M(X)$ satisfying (i) and (iii),
$R(X) = Q(X, M(X))$ is the zero polynomial.
Notice that if this is true $(y - M(x)) \mid Q(X,Y)$.

Note: $\deg(R(X)) = (1, k-1)$ degree of $Q(X,Y)$
$$= \max_{i,j} \{ i + j(k-1) \mid \exists \text{ non zero coefficient } q_{ij} \text{ in } Q(X,Y) \}$$
this is because $M(X)$ itself has degree $(k-1)$.

construction motivation for algorithm 2.

Motivated by step 2, define $Q(X,Y)$ as –

$$Q(x,y) = \sum_{\substack{i,j \\ i + j(k-1) \leq D}} q_{ij} \, x^i y^j$$

How to choose D?

$\#$roots of $R(x) \geq n-e$. (which would imply zero poly)

$\Rightarrow n - e > D$

$\therefore e < n - D$ we want to choose D so that $e < 1 - \sqrt{2R}$

$\Rightarrow \#$ coeffs of $Q(X,Y) > \#$ equation satisfied by the coefficients of $Q(X,Y)$.

Recall LHS was $\geq \dfrac{D(D+1)}{2(k-1)}$

RHS was n.

so if $\dfrac{D(D+2)}{2(k-1)} > n$, then we're done.

Algorithm 3

1. Interpolation: Find non-zero $Q(X,Y)$ s.t. $(1,k-1)$ deg of $Q \leq D$ such that (α_i, y_i) is a root of Q with multiplicity r.

   we'll define this later.

Increasing the multiplicity increases the no. of constraints.

Claim: There are now $\frac{n(r+1)r}{2}$ constraints.

We want $\left\{ \# \text{coefficients} \geq \frac{D(D+2)}{2(k-1)} > \frac{n(r+1)r}{2} \right.$

Thus if $D = \sqrt{nr(r+1)\cdot(k-1)}$,

we'll see that $\frac{e}{n} < 1 - \sqrt{R}$ will work.

Formalizing Claim 1: If (α_i, y_i) is a root of multiplicity r, for each $i = 1, \ldots, n$, then the # constraints satisfied by the coefficients of $Q(X,Y)$ is $\frac{nr(r+1)}{2}$.

2. Same as the previous two algorithms.

Again correctness is ensured by a degree argument on $R(X) = Q(X, M(X))$.

i.e. # roots of $R(X)$ (counted with multiplicity) $> \deg(R(X)) = D$.

$\Rightarrow (n-e)r > D$.

Claim: $\frac{e}{n} < 1 - \sqrt{R}$ given that $D = \sqrt{nr(r+1)(k-1)}$

Proof: $(n-e)r > D$

$$\Rightarrow e < n - \frac{D}{r} = n - \sqrt{\frac{n(r+1)(k-1)}{r}}$$

$$\Rightarrow \frac{e}{n} < 1 - \sqrt{\frac{(r+1)(k-1)}{nr}}$$

Now if we choose $r = k-1$,

$$\frac{e}{n} < 1 - \sqrt{\frac{k}{n}} = 1 - \sqrt{R}$$

Claim 2: If $M(x)$ is a poly such that (a) $\deg(M(x)) \leq k-1$
(c) $d_H(y, M|_{\alpha_1 \ldots \alpha_n}) \leq e$
and $Q(X,Y)$ is a poly obtained from step 1, then $R(X)$ has at least $n-e$ roots in $\{\alpha_1, \ldots, \alpha_n\}$ each with multiplicity r.

i.e. If α_i is a root of $R(X)$, $(X-\alpha_i)^r \mid R(X)$.

Multiplicity of roots

$f(x)$ has a root at 0, with multiplicity r, if $x^r \mid f(x)$ (or) $f(x)$ does not have monomials of degree $< r$.

$f(x)$ has a root at α, with multiplicity r, if $(x-\alpha)^r \mid f(x)$.

Note: $f(x)$ has root α with multiplicity r iff $f(x+\alpha)$ has root 0 with multiplicity r. i.e. $x^r \mid f(x+\alpha)$ or $f(x+\alpha)$ has no monomial of $\deg < r$.

$Q(x,y)$ has a root at $(0,0)$ with multiplicity r if $Q(x,y)$ contains no monomial of total degree $< r$.

(or) $\sum_{i,j} q_{ij} x^i y^j$, $q_{ij} = 0$ for $0 \leq i+j \leq r-1$.

For example - $Q(X,Y) = (X+Y)(X-Y)$ has multiplicity 2.

Note: $Q(X,Y)$ has a root at (α, y) with multiplicity r, if
$Q(X+\alpha, Y+y)$ has root $(0,0)$ with multiplicity r.

$\boxed{\text{Claim 1}}$: If (α_i, y_i) is a root of multiplicity r, for each $i = 1, \dots, n$ then the # constraints satisfied by the coefficients of $Q(X,Y)$ is $\frac{nr(r+1)}{2}$.

Proof: (α_i, y_i) being a root of multiplicity $r \Rightarrow$
$Q(X+\alpha_i, Y+y_i)$ has root at $(0,0)$ with multiplicity r

$$\Rightarrow Q(X+\alpha, Y+y) = \sum_{i,j} q_{ij} (X+\alpha)^i (Y+y)^j$$

$$= \sum_{ij} \tilde{q}_{ij} X^i Y^j$$

We have $\tilde{q}_{ij} = 0$, $\forall\ i + j \leq r-1$ (by multiplicity)
For every $j \in [0, r-1]$, we have $i \in [0, r-1-j]$.

$$\Rightarrow \#\, i,j \text{ is } \sum_{j=0}^{r-1} (r-j) = r^2 - \frac{r(r-1)}{2}$$

$$= \frac{2r^2 - r^2 + r}{2} = \frac{r(r+1)}{2}$$

$\Rightarrow \frac{r(r+1)}{2}$ coeffs of $Q(X+\alpha, Y+y)$ are zero.
(each of which are constraints).

Let's try to find q_{ij} in terms of $\tilde{q}_{i'j'}$.

$$\sum_{\substack{i,j \\ i+j(k-1) \leq D}} q_{ij} (X+\alpha)^i (Y+y)^j = \sum_{\substack{i',j' \\ i'+j'(k-1) \leq D}} \tilde{q}_{i'j'} X^{i'} Y^{j'}$$

LHS:
$$\sum_{i,j} q_{ij} \left[\sum_{i'=0}^{i} \binom{i}{i'} x^{i'} \alpha^{i-i'}\right]\left[\sum_{j'=0}^{j} \binom{j}{j'} y^{j'} y^{j-j'}\right]$$

$$= \sum_{i,j} q_{ij} \sum_{i',j'} \binom{i}{i'}\binom{j}{j'} x^{i'} y^{j'} \alpha^{i-i'} y^{j-j'}$$

$$= \sum_{i'j'} \underbrace{\left(\sum_{\substack{i \geq i' \\ j \geq j'}} q_{ij} \binom{i}{i'}\binom{j}{j'} \alpha^{i-i'} y^{j-j'}\right)}_{\tilde{q}_{i'j'}} x^{i'} y^{j'}$$

Thus each (α_i, y_i), $i = 1, \cdots, n$ gives $\frac{r(r+1)}{2}$ equations.

$\boxed{\text{Claim 2:}}$ If $M(x)$ is a poly such that (a) $\deg(M(x)) \leq k-1$

(c) $d_H(y, M|_{\alpha_1 \cdots \alpha_n}) \leq e$

and $Q(X,Y)$ is a poly obtained from step 1, then $R(X)$ has at least $n-e$ roots in $\{\alpha_1, \cdots, \alpha_n\}$ each with multiplicity r.

i.e. If α_i is a root of $R(X)$, $(X - \alpha_i)^r \mid R(X)$.

Proof: Again $y = M(\alpha_i)$ at at least $n-e$ positions, hence

$R(X) = Q(X, M(X))$ is 0 at at least $n-e$ positions.

Let (α, y) be at one such position, we want to show that $R(X)$ has root at α with multiplicity r.

Equivalently, $R(X+\alpha)$ is divisible by X^r.

$$R(X+\alpha) = Q(X+\alpha, M(X+\alpha))$$
$$= Q(X+\alpha, M(X+\alpha) - y + y)$$

$$= Q(X+\alpha,\ \tilde{M}(x+\alpha)+y)$$

We know that $Q(X+\alpha, Y+y)$ does not have a monomial of degree $< r$.

Note: $\tilde{M}(X+\alpha) = M(X+\alpha) - y$, we have
$$\tilde{M}(0+\alpha) = M(0+\alpha) - y = M(\alpha) - y = 0.$$
$$\Rightarrow X \mid \tilde{M}(X+\alpha) \Rightarrow \tilde{M}(X+\alpha) = X \cdot g(X)$$
(for some $g(X)$)

Now $R(X+\alpha) = Q(X+\alpha,\ \tilde{M}(X+\alpha)+y)$

$$= \sum_{i,j} \tilde{q}_{ij}\, X^i\, \tilde{M}(X+\alpha)^j$$

$$= \sum_{i,j} \tilde{q}_{ij}\, X^i X^j \cdot g(X)^j$$

$$= \sum_{\substack{i,j \\ i+j \geq r}} \tilde{q}_{ij}\, X^{i+j} g(X)^j \Rightarrow X^r \mid R(X+\alpha)$$
($\because$ $i+j \geq r$)