CS 70 Discussion 5B

October 4, 2024

Error-Correcting Codes

Problem: We want to send a message of length n (n numbers $v_0, v_1, \ldots, v_{n-1}$) across an unreliable channel, but still ensure that the receiver can reconstruct the original message.
Solution: Redundancy! Send more than n packets to account for possible errors. But, what should these extra packets be? How does the receiver reconstruct the original message using the extra packets?

Erasure Errors

Problem: We want to tolerate up to k of our packets being erased (i.e. not making it to the destination).

Solution:

- Sender: Create a degree $\leq n-1$ polynomial P with our n packets. We use points $(0, v_0), (1, v_1), \ldots, (n-1, v_{n-1})$ to construct the polynomial with Lagrange Interpolation.
- Sender: Send $n+k$ unique points on the polynomial. In particular, $(0, P(0)), (1, P(1)), \ldots, (n+k-1, P(n+k-1))$ across the channel
- Receiver: Use Lagrange Interpolation on n of the received points to get the original polynomial P
- Receiver: Evaluate P at the n "packet positions" to get the original message: $v_0 = P(0), v_1 = P(1), \ldots, v_{n-1} = P(n-1)$

General Errors

Problem: We want to tolerate up to k of our packets being corrupted (i.e. their values are modified over the channel).
Solution:

- Sender: Create a degree $\leq n-1$ polynomial P with our n packets. We use points $(0, v_0), (1, v_1), \ldots, (n-1, v_{n-1})$ to construct the polynomial with Lagrange Interpolation.
- Sender: Send $n+2k$ unique points on the polynomial. In particular, $(0, P(0)), (1, P(1)), \ldots, (n+2k-1, P(n+2k-1))$ across the channel
- Receiver: Gets packets $(0, r_0), (1, r_1), \ldots, (n+2k-1, r_{n+2k-1})$ where at most k points are corrupted (i.e. $r_i \neq P(i)$)
- Receiver: Use Berlekamp-Welch Algorithm to get P
- Receiver: Evaluate P at the n "packet positions" to get the original message: $v_0 = P(0), v_1 = P(1), \ldots, v_{n-1} = P(n-1)$

Berlekamp-Welch Algorithm

Error Polynomial: $E(i) = (i - e_0)(i - e_1)...(i - e_{k-1})$, where e_j = a *potential* i-value where $P(i) \neq r_i$ (i.e. packet i is corrupted)
Product Polynomial: $Q(i) = P(i)E(i) = r_i E(i)$

- If $P(i) \neq r_i$, then $E(i) = 0$, so $P(i)E(i) = r_i E(i)$
- If $P(i) = r_i$, then $P(i)E(i) = r_i E(i)$

Important: $\deg(E) \leq k$, $\deg(P) \leq n-1$, and $\deg(Q) \leq k+n-1$
Solution: Solve

$$(\forall i \in \{0, 1, ..., n+2k-1\}) \left[\sum_{j=0}^{n+k-1} a_j i^j = r_i \left(x_k + \sum_{j=0}^{k-1} b_j i^j \right) \right]$$

$n + 2k$ linear equations and $n + 2k$ unknowns (r_is are already known), so use row reduction to solve for all a_js and b_js. Now, we know E and Q, and we know $P(x) = \frac{Q(x)}{E(x)}$, so use polynomial division to get P.