5 Errors, Error-Correcting Codes, and Intro to Counting

Computer Science Mentors 70

Mar 01 - Mar 05
Spring 2021

Amogh Gupta, Sylvia Jin, Aekus Bhathal, Abinav Routhu, Debayan Bandyopadhyay
Roast us here: `https://tinyurl.com/csm70-feedback20`

1 Erasures

1. Sylvia would like to send a secure message to Amogh over a channel of size n. This means Sylvia can send at most n packets at a time across the channel. However, the channel is not very reliable.

   Assume the channel behaves as follows: of the first batch of n packets, it corrupts none; of the second batch of n packets, it corrupts exactly 1; of the third batch of n, it corrupts exactly 2; and so on, until for the $(n+1)^{\text{th}}$ batch of n packets (and thereafter), it corrupts all of them.

   Suppose we use error correcting codes for each batch of packets in order to recover the original messages which were sent through the channel. What is the maximum size message (in terms of packets) that we can send? Your final answer should be a closed-form expression, but keeping it as a summation is also acceptable.

   Assume n is even.

Solution: In a batch with k corruptions, the maximum number of packets m_k we can send using error correcting codes must satisfy $m_k + 2k = n$, or $m_k = n - 2k$ (NOTE: the n here stands for a different quantity than the $n + 2k$ for general errors mentioned in the note). The total number of packets sent is then

$$\sum_{k=0}^{n/2} m_k = \sum_{k=0}^{n/2} n - 2k$$

From here, simplifying further requires knowing the sum identity $\sum_{i=0}^{n} i = n(n+1)/2$:

$$\begin{aligned}
\sum_{k=0}^{n/2} m_k = \sum_{k=0}^{n/2} n - 2k &= \sum_{k=0}^{n/2} n - \sum_{k=0}^{n/2} 2k \\
&= \left(\frac{n}{2} + 1\right) n - 2\sum_{k=0}^{n/2} k \\
&= \left(\frac{n^2}{2} + n\right) - 2\left(\frac{(n/2)(n/2+1)}{2}\right) \\
&= \frac{n^2}{2} + n - \left(\frac{n}{2} \cdot \frac{n+2}{2}\right) \\
&= \frac{n^2}{2} + n - \frac{n^2+2n}{4} \\
&= \left(\frac{n^2}{2} - \frac{n^2}{4}\right) + \left(n - \frac{n}{2}\right) \\
&= \boxed{\frac{n^2}{4} + \frac{n}{2}}
\end{aligned}$$

2 Errors/Corruptions

1. Leanne is playing Among Us with 9 of her other friends.

   (a) Leanne wishes to send a message to her friends to tell her friends the room code, such that if all 9 of her friends join together, they can determine the room code. What kind of scheme could Leanne use?

   Solution: A polynomial of degree 8 is uniquely determined by the values of the polynomial at 9 points, so Leanne can create a degree 8 polynomial P with $P(0) = m$, where m is the room code, and send $P(1), P(2), \ldots, P(9)$ to her 9 friends. The value of m can only be determined if all 9 friends combine their values and use Lagrange Interpolation to compute the polynomial P.

   (b) There are two imposters, who are working together to not get caught. Leanne is not an imposter, and she wishes to send a message to her friends to confirm this fact with her friends. She uses the same scheme as in part (a), where her message is 0 if Leanne is not an imposter and 1 if Leanne is an imposter. How could the two imposters work together to make Leanne seem like an imposter?

   Solution: Suppose that person k is an imposter. By Lagrange interpolation, there is a unique degree 8 polynomial Q such that $Q(0) = 1$, $Q(1) = P(1), \cdots, Q(k-1) = P(k-1), Q(k+1) = P(k+1), \cdots, Q(9) = P(9)$. Thus, person k should claim that they received the point $Q(k)$. When all 9 friends use Lagrange Interpolation, they will get the polynomial Q, and conclude that since $Q(0) = 1$, Leanne is the imposter.

   (c) Leanne now knows that the imposters will do what they did in part (b). How should Leanne change her scheme to make sure that the message is sent correctly and determine at least one imposter?

   Solution: Leanne wishes to send 9 points of a polynomial, with at most 2 general errors (corresponding to the imposters claiming a different value than the one they receive). This corresponds to a "message" of length $n = 9-2\cdot 2 = 5$; in other words, her polynomial should be of degree $n-1 = 4$.
   Thus, consider the scheme where Leanne creates a degree 4 polynomial P with $P(0) = 0$ and sends $P(1), P(2), \ldots, P(9)$ to her 9 friends. By the Berlekamp-Welch Algorithm, her friends will be able to determine P and E; E will allow them to know who at least one imposter is, and after computing P, her friends will look at $P(0)$ to determine that Leanne is not an imposter.

2. In this problem, we explore why we need $n + 2k$ points to correct for k general errors. Suppose Alice is trying to send a message of length n to Bob, but she knows that if she sends a message, k packets will be corrupted. Alice knows that by Berlekamp-Welch, she should send $n + 2k$ packets to ensure that the message can be decoded. By considering an adversary, Eve, who can corrupt k packets of her choice, show that $n + 2k$ is the minimal number of packets to send to be able to decode the message; in other words, if Alice sends fewer than $n + 2k$ packets, then the message could potentially not be decoded.

Solution: Alice will use a polynomial P of degree $n - 1$ to send a message of length n. Suppose that Alice sends $n + \ell$ packets. Then when the message is sent, there will be $n + \ell - k$ packets that are correct and k packets that are corrupted. To find the correct polynomial P, Bob looks for $n + \ell - k$ packets that lie on a degree $n - 1$ polynomial and concludes that that polynomial is correct.

Suppose that Eve wishes to corrupt the packets in a way to trick Bob. What Eve can do is take $n + \ell - 2k$ of the correct packets, construct a different polynomial Q of degree $n - 1$ that passes through these $n + \ell - 2k$, then corrupt k packets so that they lie on this polynomial Q. Bob would not be able to determine whether or not P or Q is correct, and so would not be able to decode the message.

This is only possible if $n + \ell - 2k < n$, since n points determine a degree $n - 1$ polynomial. Thus, Bob could potentially be unable to decode the polynomial if $\ell < 2k$, as desired.

3 Berlekamp-Welch

(a) Alice sends Bob a message of length 3 on the Galois Field of 5 (modular space of mod 5). Bob receives the following message: $(3, 2, 1, 1, 1)$. Assuming that Alice is sending messages using the proper general error message sending scheme, set up the linear equations that, when solved, give you the $Q(x)$ and $E(x)$ needed to find the original $P(x)$.

Solution: To determine $Q(x)$ and $E(x)$, set up 5 equations for the five values of x that we have, such that

$$Q(x) = r_x * (x - b) \mod 5$$

where

$$Q(x) = a_3 * x_i^3 + a_2 * x_i^2 + a_1 * x_i + a_0$$

(where r_i = ith received number). We then set up the linear equations as follows:

$$\begin{aligned}
x = 1 &: a_3 + a_2 + a_1 + a_0 = 3(1 - b) \pmod 5 \\
x = 2 &: 3a_3 + 4a_2 + 2a_1 + a_0 = 2(2 - b) \pmod 5 \\
x = 3 &: 2a_3 + 4a_2 + 3a_1 + a_0 = 1(3 - b) \pmod 5 \\
x = 4 &: 4a_3 + 1a_2 + 4a_1 + a_0 = 1(4 - b) \pmod 5 \\
x = 5 &: 0 + 0 + 0 + a_0 = 1(0 - b) \pmod 5
\end{aligned}$$

The solution to these equations is: $b = 3$, $a_3 = 1$, $a_2 = 3$, $a_1 = 3$, $a_0 = 2$.

(b) What is the encoded message that Alice actually sent? Which packet(s) were corrupted?

Solution: This means that $Q(x) = x^3 + 3x^2 + 3x + 2$ and $E(x) = x - 3$.
To find the actual encoded message we use $P(x) = \frac{Q(x)}{E(x)}$. We use long division to find that $P(x) = x^2 + 6x + 21 + \frac{65}{x-3} = x^2 + x + 1 \pmod 5$. We plug in the values 1, 2, 3, 4, 5 to find the encoded 5 packet message.

$$\begin{aligned}
P(1) &= 1^2 + 1 + 1 = 3 \pmod 5 \\
P(2) &= 2^2 + 2 + 1 = 7 = 2 \pmod 5 \\
P(3) &= 3^2 + 3 + 1 = 13 = 3 \pmod 5 \\
P(4) &= 4^2 + 4 + 1 = 21 = 1 \pmod 5 \\
P(5) &= 5^2 + 5 + 1 = 31 = 1 \pmod 5
\end{aligned}$$

The actual message is $(3, 2, 3, 1, 1)$. We see that the 3rd packet doesn't match the initial message of $(3, 2, 1, 1, 1)$, and is therefore the corrupted one (we could also have seen this by looking at the value of b).

4 SUPERmutations

2. How many ways are there to arrange the letters of the word "SUPERMAN"...

(a) ...on a straight line?

Solution: 8!

(b) ...on a straight line, such that "SUPER" occurs as a substring?

Solution: 4! Treat "SUPER" as one character.

(c) ...on a circle? Note: If we arrange elements on a circle, all permutations that are "shifts" are equivalent (i.e. SUPERMAN and UPERMANS).

Solution: 7! Anchor one element, arrange the other 7 around in a line.

(d) ...on a circle, such that "SUPER" occurs as a substring? Reminder: SUPER can occur anywhere on the circle!

Solution: 3! Treat "SUPER" as a single character, anchor one element, and arrange the other 3 around in a line.

3. Now how many ways are there to arrange the letters of the word "SUPERMAN"...

(a) ...on a straight line, such that "SUPER" occurs as a subsequence (S U P E R appear in that order, but not necessarily next to each other)?

Solution: $3! \times \binom{8}{3}$ There are two ways to think about the problem.

We can arrange the letters of SUPERMAN 8! ways, but divide by 5! because we have arranged SUPER in any of 5! ways, when we only want one way. This gives us $8!/5!$.

Alternatively, we can think about picking three slots for "MAN" and then permute them. Then "SUPER" should fill in the other 5 slots with S going first and then U, P, E, R as they have to appear as a subsequence. This gives us $3! \times \binom{8}{3}$.

You're encouraged to check that the above two methods give the same answer.

(b) ...on a circle, such that "SUPER" occurs as a subsequence (S U P E R appear in that order, but not necessarily next to each other)?

Solution: $\binom{7}{3} \times 3! \times 2 = \binom{7}{2} \times 2! \times 5 \times 2 = 420$.

There are two methods. Method 1: anchor one of {S, U, P, E, R}. Choose which 3 places to put the M, A, and N (7 choose 3) and allow them to be shuffled (3!). Then the U, P, E, R must fill in the remaining slots in order. Finally, choose whether SUPER is placed clockwise or counterclockwise around the circle. We get $\binom{7}{3} \times 3! \times 2$.

Method 2: anchor one of {M, A, N}. Choose which of the 2 remaining 8 spots to place the A and N, allowing shuffles ($\binom{7}{2} \times 2!$). Then, choose which of the 5 remaining spots to place the S (the other letters must follow in order after the S). Finally, choose whether SUPER is placed clockwise or counterclockwise around the circle. We get $\binom{7}{2} \times 2! \times 5 \times 2$.

You are encouraged to check the two answers above are equivalent.

4. How many 5-digit sequences have the digits in ...

(a) strictly increasing order?

Solution: This is equivalent to choosing five digits without replacement and order doesn't matter, which corresponds to $\binom{10}{5}$ ways.

(b) non-decreasing order?

Solution: This can be framed as a stars and bars problem. We have 9 bars representing the separation of 10 types of digits $(0, 1, \ldots, 9)$ and 5 stars representing the 5 digits. The location of a star represents the value of its associated digit. For example, a star placed before the first bar represents 0, between the first bar and second bar represents 1, etc. There are $\binom{14}{9} = 2002$ ways to arrange 9 bars and 10 stars, which gives us the answer.