CS 70 Discrete Mathematics and Probability Theory
Fall 2017 Satish Rao and Kannan Ramchandran

HW 6

Sundry

Before you start your homework, write down your team. Who else did you work with on this homework? List names and email addresses. (In case of homework party, you can also just describe the group.) How did you work on this homework? Working in groups of 3-5 will earn credit for your "Sundry" grade.

Please copy the following statement and sign next to it:

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up.

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up. (*signature here*)

1 How Many Polynomials?

Let $P(x)$ be a polynomial of degree at most 2 over GF(5). As we saw in lecture, we need $d+1$ distinct points to determine a unique d-degree polynomial.

(a) Assume that we know $P(0) = 1$, and $P(1) = 2$. Now we consider $P(2)$. How many values can $P(2)$ have? How many distinct polynomials are there?

(b) Now assume that we only know $P(0) = 1$. We consider $P(1)$, and $P(2)$. How many different $(P(1), P(2))$ pairs are there? How many different polynomials are there?

(c) How many different polynomials of degree at most d over GF(p) are there if we only know k values, where $k \leq d$?

Solution:

(a) 5 polynomials, each for different values of $P(2)$.

(b) Now there are 5^2 different polynomials.

(c) p^{d+1-k} different polynomials. For $k = d+1$, there should only be 1 polynomial.

2 Secret Sharing with Spies

An officer stored an important letter in her safe. In case she is killed in battle, she decides to share the password (which is a number) with her troops. However, everyone knows that there are 3 spies among the troops, but no one knows who they are except for the three spies themselves. The 3 spies can coordinate with each other and they will either lie and make people not able to open the safe, or will open the safe themselves if they can. Therefore, the officer would like a scheme to share the password that satisfies the following conditions:

- When M of them get together, they are guaranteed to be able to open the safe even if they have spies among them.
- The 3 spies must not be able to open the safe all by themselves.

Please help the officer to design a scheme to share her password. What is the scheme? What is the smallest M? Show your work and argue why your scheme works and any smaller M couldn't work.

Solution:

The key insight is to realize that both polynomial-based secret-sharing and polynomial-based error correction work on the basis of evaluating an underlying polynomial at many points and then trying to recover that polynomial. Hence they can be easily combined.

Suppose the password is s. The officer can construct a polynomial $P(x)$ such that $s = P(0)$ and share $(i, P(i))$ to the i-th person in her troops. Then the problem is: what should the degree of $P(x)$ be and what is the smallest M?

First, the degree of polynomial d should not be less than 3. It is because when $d < 3$, the 3 spies can decide the polynomial $P(x)$ uniquely. Thus, n will be at least 4 symbols.

Let's choose a polynomial $P(x)$ of degree 3 such that $s = P(0)$. We now view the 3 spies as 3 general errors. Then the smallest $M = 10$ since n is at least 4 symbols and we have $k = 3$ general errors, leading us to a "codeword" of $4 + 2 \cdot 3 = 10$ symbols (or people in our case). Even though the 3 spies are among the 10 people and try to lie on their numbers, the 10 people can still be able to correct the $k = 3$ general errors by the Berlekamp-Welch algorithm and find the correct $P(x)$.

Alternative solution:

Another valid approach is making $P(x)$ of degree $M - 1$ and adding 6 public points to deal with 3 general errors from the spies. In other words, in addition to their own point $(i, P(i))$, everyone also knows the values of 6 more points, $(t+1, P(t+1)), (t+2, P(t+2)), \ldots, (t+6, P(t+6))$, where t is the number of the troops. The spies have access to total of $3 + 6 = 9$ points so the degree $M - 1$ must be at least 9 to prevent the spies from opening the safe by themselves. Therefore, the minimum M is 10.

3 Error-Correcting Codes

(a) Recall from class the error-correcting code for erasure errors, which protects against up to k lost packets by sending a total of $n+k$ packets (where n is the number of packets in the original message). Often the number of packets lost is not some fixed number k, but rather a *fraction* of the number of packets sent. Suppose we wish to protect against a fraction α of lost packets (where $0 < \alpha < 1$). At least how many packets do we need to send (as a function of n and α)?

(b) Repeat part (a) for the case of general errors.

Solution:

(a) Suppose we send a total of m packets (where m is to be determined). Since at most a fraction α of these are lost, the number of packets received is at least $(1-\alpha)m$. But in order to reconstruct the polynomial used in transmission, we need at least n packets. Hence it is sufficient to have $(1-\alpha)m \geq n$, which can be rearranged to give $m \geq n/(1-\alpha)$.

(b) Suppose we send a total of $m = n+2k$ packets, where k is the number of errors we can guard against. The number of corrupted packets is at most αm, so we need $k \geq \alpha m$. Hence $m \geq n+2\alpha m$. Rearranging gives $m \geq n/(1-2\alpha)$.

Note: Recovery in this case is impossible if $\alpha \geq 1/2$.

4 Error-Correcting Polynomials

(a) Alice has a length 8 message to Bob. There are 2 communication channels available. When n packets are fed through channel A, the channel will only deliver 5 packets (picked at random). Similarity, channel B will only deliver 5 packets (picked at random), but it will also corrupt (change the value) of one of the delivered packets. All channels can only work if at least 10 packets are sent through it. Using the 2 channels, how can Alice send the message to Bob?

(b) Alice wishes to send a message to Bob as the coefficients of a degree 2 polynomial P. For a message $[m_1, m_2, m_3]$, she creates polynomial $P = m_1x^2 + m_2x + m_3$ and sends 5 packets: $(0,P(0)), (1,P(1)), (2,P(2)), (3,P(3)), (4,P(4))$. However, Eve interferes and changes one of the values of a packet before it reaches Bob. If Bob receives

$$(0,3), (1,0), (2,3), (3,0), (4,3),$$

and knows Alice's encoding scheme and that Eve changed one of the packets, can he still figure out what the original message was? If so find it as well as the x-value of the packet that Eve changed, if not, explain why he can not. (Work in mod 11.)

(c) Alice decides that putting the message as the coefficients of a polynomial is too inefficient for long messages because the degree of the polynomial grows quite large. Instead, she

decides to encode the message as values in a degree 2 polynomial. For a 5 length message $[m_1, m_2, m_3, m_4, m_5]$, she creates a degree 2 polynomial P such that $P(0) = m_1, P(1) = m_2, P(2) = m_3, P(3) = m_4, P(4) = m_5$. She then sends the length 5 message directly to Bob as 5 packets: $(0, m_1), (1, m_2), (2, m_3, (3, m_4), (4, m_5)$. Eve again interfere and changes the value of a packet before it reaches Bob. If Bob receives $(0,0), (1,3), (2,0), (3,3), (4,0)$ and knows Alice's encoding scheme and that Eve changed one of the packets, can he still figure out what the original message was? If so find it as well as the x-value of the packet that Eve changed, if not, explain why he can not. (Work in mod 11.)

(d) After getting tired of decoding degree 2 polynomials, Bob convinces Alice to send messages using a degree 1 polynomial instead. To be on the safer side, Alice decides to continue to send 5 points on the polynomial even though it is only degree 1. She encodes and sends a length 5 message in the same way as part (c) (except using a degree 1 polynomial). Eve however, decides to change 2 of the packets. After Eve interferes, Bob receives $(0,-3), (1,-1), (2,x), (3,-3), (4,5)$. If Alice sent $(0,-3), (1,-1), (2,1), (3,3)$, $(4,5)$, for what values of x will Bob not be able to uniquely determine the Alice's message? (Assume Bob knows that Eve changed 2 of the packets and work in mod 13.)

Solution:

(a) Channel A will deliver 5 packets so we can send a message of length 5 encoded on a polynomial of degree 4 though it. If we send 10 points though channel A, it doesn't matter which 5 points bob gets, he will still be able to reconstruct our degree 4 polynomial. Since the channel B has 1 general error, we can only send a message of length 3 encoded on a degree 2 polynomial through it. If we send 10 points, Bob will get 5 points to calculate a degree 2 polynomial with 1 general error, which he is able to do. Thus to send our length 8 message, we can send the character 1 - 5 through a channel A and the characters 6 - 8 through channel B.

(b) Since we have 5 points, we have to find a polynomial of degree 2 that goes through 4 of those points. The point that the polynomial does not go through will be the packet that Eve changed. Since 3 points uniquely determine a polynomial of degree 2, we can pick 3 points and check if a 4th point goes through it.
We pick the points $(0,3), (1,0), (3,0)$. Lagrange interpolation can be used to create the polynomial be we can see that for the polynomial that goes through these 3 points, it has 0's at $x = 1$ and $x = 3$. Thus the polynomial is $(x-1)(x-3) = x^2 - 4x + 3$. We then check to see if the this polynomial goes through one of the 2 points that we didn't use. Plugging in 4 for x, we get 3. The packet that Eve changed is the point that our polynomial does not go through which has x-value 2. Alice's original message was $m_1 = 1, m_2 = -4, m_3 = 3$.
Alternative solution: Here is another valid approach for recovering a small degree (2 in this problem) polynomial:

We want to interpolate a polynomial through a single point (x_0, y_0). In this case, the first point $(0,3)$. The only constant function that passes through $(0,3)$ is $f_0(x) = 3$. Now say we wanted to define the polynomial $f_1(x)$ that passes through the points $(0,3)$ and $(1,0)$. We will write

$f_1(x) = f_0(x) + a_1(x - x_0)$, and solve for a_1, we get that

$$a_1 = \frac{y_1 - f_0(x_1)}{x_1 - x_0} = \frac{0-3}{1-0} = -3.$$

$$f_1(x) = -3x + 3.$$

Now say we want a polynomial $f_2(x)$ that passes through $(0,3)$, $(1,0)$, and $(3,0)$. If we write $f_2(x) = f_1(x) + a_2(x - x_0)(x - x_1)$, we just need a_2 such that

$$f_1(x_2) + a_2(x_2 - x_0)(x_2 - x_1) = y_2.$$

$$a_2 = \frac{y_2 - f_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} = 1.$$

Thus the polynomial is $P(x) = -3x + 3 + (x-0)(x-1) = x^2 - 4x + 3$

(c) To find the polynomial P, we can use Berlekamp and Welch. We have: $Q(x) = P(x)E(x)$. $E(x)$ has degree 1 since we know we have at most 1 error. $Q(x)$ is degree 3 since $P(x)$ is degree 2. We can write a system of linear equations and solve:

$$d = 0(0 - e)$$

$$a + b + c + d = 3(1 - e)$$

$$8a + 4b + 2c + d = 0(2 - e)$$

$$27a + 9b + 3c + d = 3(3 - e)$$

$$64a + 16b + 4c + d = 0(4 - e)$$

Since we are working in mod 11, this is equivalent to:

$$d = 0$$

$$a + b + c + d = 3 - 3e$$

$$8a + 4b + 2c + d = 0$$

$$5a + 9b + 3c + d = 9 - 3e$$

$$9a + 5b + 4c + d = 0$$

Solving yields:

$$Q(x) = -x^3 + 6x^2 - 8x, E(x) = x - 2$$

To find $P(x)$ we divide $Q(x)$ by $E(x)$ and get $P(x) = -x^2 + 4x$. To recover Eve's message, we evaluate the polynomial at: $0, 1, 2, 3, 4$ and get the message: $0, 3, 4, 3, 0$. The x-value of the packet that Eve changed is 2 (given by $E(x)$).

(d) Since Bob knows that Eve changed 2 of the points, the 3 remaining points will still be on the degree 1 polynomial that Alice encoded her message on. Thus if Bob can find a degree 1 polynomial that passes through at least 3 of the points that he receives, he will be able to uniquely recover Eve's message. The only time that Bob cannot uniquely determine Alice's message is if there are 2 polynomials with degree 1 that passes through 3 of the 5 points that he receives. Since we are working with degree 1 polynomials, we can plot the points that Bob receives and then see which values of x will cause 2 sets of 3 points to fall on a line. $(0,-3),(1,-1),(4,5)$ already fall on a line. If $x=-2$, $(1,-1),(2,-2),(3,-3)$ also falls on a line. If $x=-3$, $(0,-3),(2,-3),(3,-3)$ also falls on a line. If $x=1$, $(0,-3),(2,1),(4,5)$ falls on the original line, so here Bob can decode the message. If $x=2$, $(2,2),(3,-3),(4,5)$ also falls on a line. So if $x=-3,-2,2$, Bob will not be able to uniquely determine Alice's message.

5 Distance Properties

Imagine that you want to send a message x of length n over the binary alphabet $\{0,1\}$. However, you want to prepare for the possibility that one of your bits will be corrupted. You therefore send a codeword y of length $m > n$ given by an encoding function $E : y = E(x)$. The encoding function is one-to-one (or else there would be ambiguity in which message you sent), and furthermore the encoding function has the property that for any two distinct codewords y_1, y_2 in the range of E, y_1 and y_2 have a Hamming distance of at least 3 (this ensures that you can correct for a corruption of at most one bit).

Prove that $m \geq n + \log_2(m+1)$, that is, the number of extra bits that you have to send is at least logarithmic in your original message length.

Hint: Try a counting approach. For each codeword y, let B_y be the set of length-m strings with a Hamming distance of at most 1 away from y. Observe that for distinct codewords y_1 and y_2, B_{y_1} and B_{y_2} do not overlap.

Solution:

Let $\mathscr{C}$ be the set of codewords (i.e., the range of E). Since the B_y sets do not overlap and their union is a subset of the length-m bit strings, we obtain the bound $2^m \geq |\bigcup_{y\in\mathscr{C}} B_y| = \sum_{y\in\mathscr{C}} |B_y| = 2^n |B_y|$, since there are 2^n codewords. Also note that $|B_y| = m+1$ since there is one length-m string at distance 0 from y (namely y itself) and m strings at distance 1 from y. Hence $2^n(m+1) \leq 2^m$ and taking logarithms we obtain the result.