CS 70 Discrete Mathematics and Probability Theory
Spring 2016 Rao and Walrand Discussion 6A Solution

1. **Polynomial intersections** Find (and prove) an upper-bound on the number of times two distinct degree d polynomials can intersect. What if the polynomials' degrees differ?

   Answer:

   Solutions: They can intersect up to d times. This is because we can specify d points to be the same for both polynomials, then the $(d+1)$th point to be different. Then the polynomials will be distinct, but still agree at d points. If $d+1$ points agree, the polynomials will be identical.

   If the polynomials have degrees d_1 and d_2, with $d_1 > d_2$, then they can intersect up to d_1 times. Pick d_1 points on the polynomial of degree d_2, and another point not on this polynomial. Then a unique degree d_1 polynomial will go through these d_1+1 points.

2. **Lagrange or Linear System**

   In this exercise we will try to find out a polynomial $P(x)$ of degree at most 2 with coefficients in $0,\ldots,4$ such that $P(1) = 2 \pmod 5$, $P(2) = 4 \pmod 5$, and $P(3) = 3 \pmod 5$.

   (a) Find out the polynomials $\Delta_i(x)$ for $i \in \{1,2,3\}$.

   Answer:

   $$\begin{aligned}
   \Delta_1(x) &= \frac{(x-2)(x-3)}{(1-2)(1-3)} = 3x^2+3 \\
   \Delta_2(x) &= \frac{(x-1)(x-3)}{(2-1)(2-3)} = 4x^2+4x+2 \\
   \Delta_3(x) &= \frac{(x-1)(x-2)}{(3-1)(3-2)} = 3x^2+x+1
   \end{aligned}$$

   (b) Combine Δ_i's with the right coefficients to find the polynomial $P(x)$.

   Answer: We have $P(x) = 2\Delta_1(x) + 4\Delta_2(x) + 3\Delta_3(x) = x^2+4x+2$.

   (c) Now we will try a different approach. Write the polynomial $P(x)$ as $c_0 + c_1x + c_2x^2$. Treating c_i's as variables what do the equations $P(1) = 2 \pmod 5$, $P(2) = 4 \pmod 5$, and $P(3) = 3 \pmod 5$ tell about c_i's?

   Answer: They give us a system of linear equations.

   $$\begin{aligned}
   P(1) = 2 &\implies c_0 + c_1 + c_2 = 2 \pmod 5 \\
   P(2) = 4 &\implies c_0 + 2c_1 + 4c_2 = 4 \pmod 5 \\
   P(3) = 3 &\implies c_0 + 3c_1 + 4c_2 = 3 \pmod 5
   \end{aligned}$$

(d) Solve the system of equations you get from the last part to solve for c_i's. What is the resulting polynomial $P(x)$?

Answer: The answer given by the system of linear equations is the same as the one gotten by Lagrange; i.e. $c_0 = 2, c_1 = 4, c_2 = 1$.

3. **Hamming codes** Here is an example of a simple single-error correcting code that works without having to invoke polynomials and just deals with bits. In real life, these kinds of codes are often used in conjunction with Reed Solomon codes. In this problem we will see how to send 4 bits (each either 0 or 1) using Hamming codes. This will be accomplished by using 3 parity bits (check bits) so that 1 error in the message can be detected and corrected. To encode a 4-bit message m, we will multiply it by a code-generator matrix G:

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(a) Show how to encode the message $m = 1101$. What is the resulting 7-bit transmitted codeword x?

Answer: **Solutions:** $Gm = x = 1010101$.

(b) To detect potential errors, we can use a parity check matrix H:

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

What do you notice about the form of H? If you multiply H by x, what do you get? Will this be true for any message m?

Answer: **Solutions:** The columns of H spell out the numbers 1-7 in binary (read bottom to top). $Hx = (0,0,0)$, which will be true for the encoding of any message m with no errors.

(c) Suppose that there was an error (flipped bit) in the 6th bit of x, yielding x'. To find out where it occurred, we can multiply H by x'. What is the result base 10? How can you correct x?

Answer: **Solutions:** We actually received $x' = 1010111$. Computing Hx', we get the sixth column of H, which corresponds to 6 in binary, so we know the 6th bit should be flipped to correct x.

(d) Now we wish to recover the original message m. To do this, we can multiply the corrected x by a recovery matrix R that pulls out the 4 original bits:

$$R = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Do you get m back?

Answer: **Solutions:** After correcting x, we compute $Rx = m$, and get back $m = 1101$.

(e) Do you think that a code like this could exist that could fix a single binary error in a 4-bit message without using at least 7 bits for the codeword? Why or why not?

Answer: **Solutions:** With 7 bits, we need to encode 8 possible error "locations" (1-7, and then the no-error option). And to send 4 bits there are $2^4 = 16$ possible messages. So there are $8 \cdot 16 = 128 = 2^7$ possibilities, which requires 7 bits. Any shorter and it wouldn't all fit.

4. **Secret Sharing**

(a) Suppose that the teaching staff of a course involves three professors and two TAs. The solutions to the next homework are encrypted by an encryption key shared by all five. The three professors together should be able to access the solutions, or any one TA with one professor, or both TAs. Suggest a secret-sharing scheme that achieves this. (*Hint:* Try weights.)

(b) Suppose now that the class is taught by three professors, each with *her* own two TAs. Any two professors can access the data, as long as one TA from each is also present. Now what?

Answer:

(a) Pick a random degree-2 polynomial $P(x)$ such that the secret is $P(0)$. The encryption key consists of the 7 pairs $(i, P(i))$ for $1 \leq i \leq 7$. Two pairs should be given to each TA and one to each professor.

(b) If you read the problem to mean that one TA of each of the *three* professors should be present, this is one possible solution:

Take a degree 5 polynomial $P(x)$, and let $P(0)$ be the secret. Give $P(1), P(2)$ to prof.A , $P(2), P(3)$ to prof.B, $P(3), P(1)$ to prof.C and $P(4)$ to A's TAs, $P(5)$ to B's TAs, $P(6)$ to C's TA's. All 6 values are needed to find the polynomial. These can be put together when any two professors are present, and at least one TA from each.

If you read the problem to mean that one TA of each of the *present* professors should be present, this is one possible solution:

First we pick a random degree-1 polynomial $P(x)$ such that the secret is $P(0)$. We then pick 3 random degree-3 polynomials $Q_1(x), Q_2(x)$ and $Q_3(x)$ such that the value of $Q_i(0) = P(i)$ for each $1 \leq i \leq 3$. In other words, the secret of each of the $Q_i(x)$ polynomials is part of the encryption key of the $P(x)$.

Now for each of the $Q_i(x)$, we divide the encryption key into 5 parts and give 3 pairs to the professor and 1 to each of her TA's. Then to recover the message, each professor i gets together with one of her TA's and retrieves the secret of the $Q_i(x)$ polynomial, which is her share of the encryption key for the $P(x)$ polynomial. Finally, two professors can get together and recover $P(0)$.

5. **Why work with primes?**

In class, you learned about erasure codes and error correcting codes, and prime numbers played a central role in both kinds of codes – since all calculations were supposed to be done modulo a *prime number*. In this problem, we will see why this is a crucial requirement, and explore what happens if this requirement is relaxed in a naive manner.

For this problem, assume that Alice wants to send n packets to Bob, across an "erasure channel" modeled as discussed in class. Let us say all calculations are done modulo $N = 12$ (note that this is *not* a prime number).

As discussed in class, let us say Alice sends $n+1$ packets to Bob, and Bob receives at least n of these packets intact. That is, the channel can erase at most 1 packet, and if it does so, Bob gets to know which packet was erased (although he does not know the contents of the erased packet).

a) Suppose $n = 1$. That is, Alice wants to send only 1 packet to Bob (plus one redundant packet to compensate for erasure). Would the scheme discussed in class work with $N = 12$? What are all the possible 2-packet lists that Alice could transmit? In each case, would Bob be able to recover Alice's message in spite of a possible erasure? Would Alice or Bob face any problems because they are doing their calculations modulo 12?

Answer: If Alice wants to send only 1 packet to Bob, then her "message" is just a single integer m, drawn from the set $\{0, 1, 2, \ldots N-1\}$. Let's call this set $\texttt{range}(N)$. Then, following the scheme discussed in class, Alice will now try to find a polynomial $P(x)$ of degree 0 that evaluates to m at $x = 1$. This polynomial, of course, has to be the constant polynomial $P(x) = m$. Now Alice will transmit both $P(1)$ and $P(2)$ to Bob (to compensate for possible erasure). So her transmission will be of the form (m, m), i.e. , a symbol drawn from $\texttt{range}(N)$ that is repeated twice. Clearly, even if one packet is erased, Bob can retrieve Alice's message from the other. So N being composite has no adverse effect when $n = 1$.

b) Now suppose $n = 2$. That is, Alice now wants to send 2 packets to Bob (plus one redundant packet to compensate for erasure). Now, would there be any problems because $N = 12$?

Answer: If Alice wants to send 2 packets to Bob, then her "message" consists of a pair of integers (m_1, m_2), each drawn from $\texttt{range}(N)$. Now Alice will try to find a polynomial $P(x)$ of degree at most 1 that evaluates to m_1 at $x = 1$ and m_2 at $x = 2$. Let's say this polynomial is $P(x) = ax + b$ (the highest power of x is 1 because the degree of the polynomial is at most 1). Then we have:

$$\begin{aligned} a + b &= m_1 \pmod N, \text{ and} \\ 2a + b &= m_2 \pmod N \end{aligned}$$

The above equations always have a solution, and what is more, this solution is unique and is given by $a = (m_2 - m_1) \bmod N$, and $b = (2m_1 - m_2) \bmod N$, regardless of whether N is prime or composite. So there is one, and only one, polynomial $P(x)$ that Alice can construct, before she transmits $(P(1), P(2), P(3))$.

However, that does *not* mean that this is the only polynomial that Bob will come up with. To see why, consider this simple example. Let us say Alice wants to send the message $(3, 5)$. Her polynomial, therefore, will be $P(x) = 2x + 1$. So far, this is unique and well-defined. The integers Alice transmits will then be $(3, 5, 7)$. Now, what if the channel erases the second packet, which leaves Bob with $(3, \square, 7)$. Bob dutifully attempts to find a polynomial $Q(x) = cx + d$ such that $Q(1) = 3$ and $Q(3) = 7$. This gives him the following 2 equations:

$$\begin{aligned} c + d &= 3 \pmod{12}, \text{ and} \\ 3c + d &= 7 \pmod{12} \end{aligned}$$

The problem is that the above equations don't have a unique solution. Indeed, there are 2 solutions that work, $(c, d) = (2, 1)$ and $(c, d) = (8, 7)$. Thus, as far as Bob can tell, Alice's polynomial might have been either $2x + 1$ or $8x + 7$. This means Alice's message could have been either $(3, 5)$ or

$(3,11)$; Bob can narrow it down to these two possibilities, but beyond that, he has no clue as to which of these is the actual message that Alice had intended.

Note: if Bob had tried Lagrange interpolation to find $Q(x)$, that would not have given him a solution either. Indeed, applying Lagrange's method, the "solution" obtained by Bob would have been:

$$\begin{aligned} Q(x) &= 3 \cdot \frac{x-3}{1-3} + 7 \cdot \frac{x-1}{3-1} \\ &= 3 \cdot \frac{x-3}{10} + 7 \cdot \frac{x-1}{2} \\ &= 3 \cdot (x-3) \cdot 10^{-1} + 7 \cdot (x-1) \cdot 2^{-1} \end{aligned}$$

But neither 10 nor 2 has an inverse in mod 12 arithmetic, so the above formula becomes meaningless.

The root cause of the above problems is that N is composite. If N had been prime, Bob's system of equations above would have had a unique solution. Alternatively, every integer in `range`(N) would have had an inverse mod N, and therefore Lagrange interpolation would have succeeded.

So N being composite has serious adverse effects when $n = 2$.

c) Now let $n = 3$ (3 packets plus one additional packet to compensate for erasure). Assume that Alice wants to encode messages into "systematic" codewords (with the first few evaluations of the polynomial being the message itself). Prove that Alice can no longer send arbitrary messages of her liking to Bob, by showing that it would be impossible for Alice to send the message $(11,6,2)$. Find 2 other examples of messages that Alice cannot send to Bob.

Answer: In the previous ($n = 2$) case, at least the polynomial $P(x)$ that Alice would need was unique and well-defined. All the problems that arose were at Bob's end, so to speak. Now, with $n = 3$, that is no longer the case (now there can be problems at Alice's end as well). Let's say Alice wants to send the message $(11,6,2)$. Thus, she has to find a polynomial $P(x) = ax^2 + bx + c$, such that

$$\begin{aligned} a + b + c &= 11 \pmod{12}, \\ 4a + 2b + c &= 6 \pmod{12}, \text{ and} \\ 9a + 3b + c &= 2 \pmod{12} \end{aligned}$$

Eliminating c from the first two equations, and then from the last two equations, we obtain:

$$\begin{aligned} 3a + b &= 7 \pmod{12}, \text{ and} \\ 5a + b &= 8 \pmod{12} \end{aligned}$$

Subtracting the first equation above from the second, we obtain:

$$2a = 1 \pmod{12}$$

However, the equation above is impossible. No matter what the value assigned to a, $2a$ never leaves remainder 1 when divided by 12. This is because 2 has no multiplicative inverse in mod 12 arithmetic.

Thus, it is impossible for Alice to send the message $(11, 6, 2)$.

Note: As in part (b) above, if Alice had tried Lagrangian interpolation, that would have failed as well, giving rise to meaningless multiplicative inverses, just like those that came up in part (b) above.

Two other messages that Alice cannot possibly send are $(0, 0, 1)$ and $(0, 1, 1)$.

d) Why does Lagrange interpolation fail when Alice tries to send the messages in part (c) above? Does Lagrange interpolation fail for any message in part (a) or part (b)? Why or why not?

Answer: The answer to this is already contained in the solutions for parts (b) and (c) above. In a nutshell, the problem has to do with the non-existence of multiplicative inverses for some integers (like 2) in mod 12 arithmetic. When Alice tries to send the messages in part (c) above, her Lagrangian interpolation requires division by a number that has no multiplicative inverse in mod 12 arithmetic. In part (a), this problem does not arise because both Alice and Bob had to deal with only constant polynomials, and multiplicative inverses did not come up in their calculations. In part (b), Alice's calculations did not require impossible multiplicative inverses, but Bob's did.

e) Suppose Alice chooses a value for n, and restricts herself to only the messages that are possible to send in modulo 12 arithmetic. For any such n, and any such message, would Bob be able to recover Alice's message in spite of a possible packet erasure? If so, how? If not, provide a counter-example.

Answer: No, Bob would not be able to recover Alice's messages even if Alice restricted herself only to messages that were possible to send in mod 12 arithmetic. A simple counter-example was described in the solution to part (b) above, where Alice sends the message $(3, 5)$ (which is very much possible to send), but Bob is unable to recover it when the second packet is erased.