CSM 70 Discrete Mathematics and Probability Theory

Spring 2016 Error Correcting, Counting? **Worksheet 6**

Key Terms

Erasure Errors	Berlekamp Welsh
General Errors	Countably infinite
Solomon Reed Codes	Halting

Goal: Alice wants to send Bob n packets over a noisy channel, where each packet is a number modulo q

Polynomial Facts:

1) There is a unique polynomial of degree _____ such that $P(i) = m_i$ for each packet $m_1, \ldots, m_n$
2) To account for errors we send $c_1=P(1), \ldots, c_{n+j}=P(n+j)$
3) If polynomial P(x) has degree n - 1 then we can uniquely reconstruct it from any n distinct points.
4) If a polynomial P(x) has degree n - 1 then it can be uniquely described by its n coefficients

I. Erasure Errors

We want to send n packets and we know that k packets could get lost.

How many more points does Alice need to send to account for k possible errors?

What degree will the resulting polynomial be?

How large should q be if Alice is sending n packets with k erasure errors, where each packet has b bits?

What would happen if Alice instead send n + k - 1? Why will Bob be unable to recover the message?

Exercises

1. Suppose A = 1, B = 2, C = 3, D = 4, and E = 5. Assume we want to send a message of length 3. Recover the lost part of the message, or explain why it can not be done.
    a. C_AA

    b. CE_ _

2. Suppose we want to send n packets, and we know p = 20% of the packets will be erased. How many extra packets should we send? What happens if p increases (say to 90%)?

II. General Errors (Berlekamp and Welch)

Now instead of losing packets, we know that k packets are corrupted. Furthermore, we do not know which k packets are changed. Instead of sending k additional packets, we will send an additional 2k.

Solomon-Reed Codes

1. Identical to erasure errors: Alice creates n - 1 degree polynomial P(x).
   $$P(x) = p_0 + p_1x + \ldots + p_{n-1}x^n$$
2. Alice sends P(1), …, P(n + 2k)
3. Bob receives R(1), …, R(n + 2k)

For how many points does R(x) = P(x)?

True or false: P(x) is the unique degree n - 1 polynomial that goes through at least n + k of the received points.

Write the matrix view of encoding the points P(1), …, P(n + 2k)

Berlekamp Welch

How do we find the original polynomial P(x)?

Suppose that $m_1, \ldots, m_k$ are the corrupted packets. Let $E(x) = (x - m_1) \ldots (x - m_k)$

Then $P(i)\ E(i) = r_i\ E(i)$ for any i greater than 1 and less than n + 2k. Why?

Let $Q(i) = P(i)\ E(i)$

So we have $Q(i) = P(i)\ E(i) = r_i\ E(i)$ where $1 \leq i \leq 2k + n$

What degree is Q(i)?

How many coefficients do we need to describe Q(i)?

What degree is E(i)?

How many unknown coefficients do we need to describe E(i)?

We can write $Q(i) = r_i\ E(i)$ for every i that is $1 \leq i \leq 2k + n$.

How many equations do we have? How many unknowns?

Once we have the above described equations, how do we determine what P(i) is?

Exercises

1) 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).

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

2) You want to send a super secret message consisting of 10 packets to the space station through some astronauts. You're afraid that some malicious spy people are going to tell the wrong message and make the space station go spiraling out of orbit. Assuming that up to 5 of the astronauts are malicious, design a scheme so that the group of astronauts (including the malicious ones) still find the correct message that you want to send. You can send any number of astronauts, but try to make the number that you have to send as small as possible.

3) Suppose that the message we want to send consists of 10 numbers. We find a polynomial of degree 9 which goes through all these points and evaluate it on 25 points. How many erasure errors can we recover from? How about general errors?

III. Countability

1. What does it mean for a set to be countably infinite?

2. Does N and Z^+ have the same cardinality? Does adding one element change cardinality?

Cantor-Bernstein Theorem: Suppose there is an injective function from set A to set B and there is an injective function from set B to set A. Then there is a bijection between A and B.

Exercises

1. Are these sets countably infinite/ uncountable infinite/ finite? If finite, what is the order of the set?
   A. Finite bit strings of length n.

   B. All finite bit strings of length 1 to n.

   C. All finite bit strings

   D. All infinite bit strings

   E. All finite or infinite bit strings.

2. Find a bijection between N and the set of all integers congruent to 1 mod n, for a fixed n.

3. Every infinite subset of a countable set is countable.

4. If A and B are both countable, then AxB is countable.

5. Every infinite set that contains an uncountable set is uncountable.

IIIB. Self Reference

The Halting Problem: Does a given program ever halt when executed on a given input?

$$\texttt{TestHalt(P,x)} = \begin{cases} \text{“yes”}, & \text{if program } P \text{ halts on input } x \\ \text{“no”}, & \text{if program } P \text{ loops on input } x \end{cases}$$

How do we prove that TestHalt doesn't exist? Let's assume that it does, and hope we reach a contradiction.

Define another program:

```
Turing(P):
      if TestHalt(P, P) = "yes" then loop
      else halt
```

What happens when we call Turing(Turing)?

How is this just a reformulation of proof by diagonalization?

Therefore the Halting Problem is unsolvable. We can use this to prove that other problems are also unsolvable. Say we are asked if program M is solvable. To prove it is not, we just need to prove the following claim:

If we can compute program M, **then** we could also compute the halting problem.

This would then prove that M can not exist, since the halting problem is not computable. This amounts to proof by contradiction.

Exercise

Say that we have a program M that decides whether any input program halts as long as it prints out the string “ABC” as the first operation that it carries out. Can such a program exist?

IV. Counting

How many ways are there to choose 3 cards from a deck? (Order matters)

How many distinct hands of 3 cards are there? (Order does not matter)

Rules of Counting:

1) If we make a sequence of choices, and there are n_1 ways to make the first choice, n_2 ways to make the second choice for every way of making the first choice, …, n_k ways to make the kth choice for every way of making all of the choices before it, then the total number of distinct objects we get will be $n_1 * n_2 * \ldots * n_k$
2) Assume that the order of the choices does not matter. Let A be the ordered objects and B be the unordered objects. If there is a mapping from A to B (k to 1) then we can count the number of ordered objects and divide by k to get the order of B.

Exercises

1) How many ways are there to order the following words?

MENTORS

CORRINA

MISSISSIPPI

2) When Order Matters
You have 15 chairs in a room and there are 9 people. How many different ways can everyone sit down?

How many ways are there to fill 9 of the 15 chairs? (We don't care who sits in them)

3)
The numbers 1447, 1005, and 1231 have something in common. Each of them is a four digit number that begins with 1 and has two identical digits. How many numbers like this are there?