CS70 - Lecture 10 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Gerald Zhang

Polynomials

- **Fact**: Any $d+1$ points specifies a distinct degree d polynomial
- **Modular Fact**: Any $d+1$ points specifies a distinct degree d polynomial in mod p space when p is prime
- **Uniqueness Fact**: At most one degree d polynomial hits $d+1$ points

Uniqueness.

Uniqueness Fact. At most one degree d polynomial hits $d+1$ points.

Proof:

Roots fact: Any degree d polynomial has at most d roots.

Assume two different polynomials $Q(x)$ and $P(x)$ hit the points.

$R(x) = Q(x) - P(x)$ has $d+1$ roots and is degree d.
Contradiction.

- **Roots Fact**: Any degree d polynomial has at most d roots

Only d roots.

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0:
$P(x) = (x-a)Q(x)$.

Proof: $P(x) = (x-a)Q(x)+r$.
Plugin a: $P(a) = r$.
It is a root if and only if $r = 0$. □

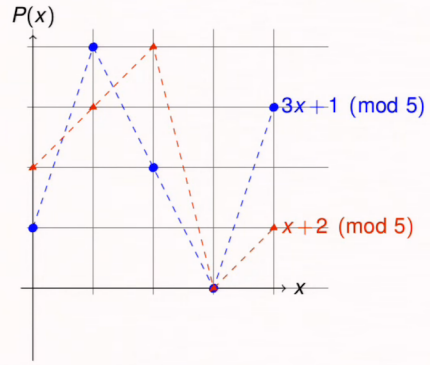
Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then
$P(x) = c(x-r_1)(x-r_2)\cdots(x-r_d)$.
Proof Sketch: By induction.

Induction Step: $P(x) = (x-r_1)Q(x)$ by Lemma 1. $Q(x)$ has smaller degree so use the induction hypothesis. □

$d+1$ roots implies degree is at least $d+1$.

Roots fact: Any degree d polynomial has at most d roots.

Polynomial: $P(x) = a_d x^4 + \cdots + a_0 \pmod p$



Finding an intersection.
$x+2 \equiv 3x+1 \pmod 5$
$\implies 2x \equiv 1 \pmod 5 \implies x \equiv 3 \pmod 5$
3 is multiplicative inverse of 2 modulo 5.
Good when modulus is prime!!

- Polynomials only map to f(x) at integer values of x
- f(x) is contained in the mod space
- Use delta functions to create meaningful polynomials in mod space

Shamir's k out of n scheme:
Secret $s \in \{0, ..., p-1\}$
Set $a_0 = s$, randomly assign $a_1, ..., a_{k-1}$
Let $P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + ... + a0$ with $P(0) = a_0 = s$
Share $(i, P(i) \bmod p)$ with i-th person
k shares gives secret (degree $= d = k-1$, Modular fact, $d+1 = k$ shares gives the polynomial which reveals $P(0) = s$
Solve for polynomial given $d+1$ coordinates

In general..

Given points: $(x_1, y_1); (x_2, y_2) \cdots (x_k, y_k)$.

Solve...

$$\begin{aligned} a_{k-1}x_1^{k-1} + \cdots + a_0 &\equiv y_1 \pmod p \\ a_{k-1}x_2^{k-1} + \cdots + a_0 &\equiv y_2 \pmod p \\ &\vdots \\ a_{k-1}x_k^{k-1} + \cdots + a_0 &\equiv y_k \pmod p \end{aligned}$$

Will this always work?

As long as solution **exists** and it is **unique!** And...

Modular Arithmetic Fact: Exactly 1 degree $\leq d$ polynomial with arithmetic modulo prime p contains $d+1$ pts.

- $d = k-1, d+1 = k$
- Solve system of linear equations to get a_0

Lagrange Interpolation

Delta Function

$$\Delta_i(x) = \begin{cases} 1, & x = x_i \\ 0, & x = x_j \text{ for } j \neq i \\ \text{doesn't matter}, & x = \text{anything else} \end{cases}$$

- 1 at one point (x-value), 0 everywhere else
- valid for a set of x values $x_1, ..., x_{d+1}$
- $y_i\Delta_i(x) = y_i$ because $\Delta_i(x)$ is 1 at x_i and 0 everywhere else
  - ⋆ $P(x) = y_1\Delta_1(x) + y_2\Delta_2(x) + ... + y_{d+1}\Delta_{d+1}(x)$ because at x_i you only get y_i (Δx_i is 0 at anything except x_i)

Formation of Delta Function:
Given points: $(x_1, y_1); (x_2, y_2); ...(x_{d+1}, y_{d+1})$

$$\Delta_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)} \tag{1}$$

CS70 - Lecture 11 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Gerald Zhang

Secret Sharing

Minimality

- Use mod p space where p is prime
- $p > n$ where n is the amount of shares you want to hand out
- $p > 2^b$ where b is the number of bits you want in your secret
- Uses **Theorem**(There is always a prime between n and $2n$). This strategy chooses a p that is within 1 bit of secret size (minimality).

Runtime

- Polynomial in terms of k, n, and $\log p$
- Evaluate $k-1$ degree polynomials n times as a system of linear equations, using $\log p$-bit numbers
- Reconstruct secret by solving system of k equations using $\log p$-bit arithmetic.

Counting

- m^{d+1}: $d+1$ coefficients must be $\in \{0, ..., m-1\}$
- m^{d+1}: $d+1$ points with y-values that must be $\in \{0, ..., m-1\}$

Erasure Codes

Solution

- n packet message, loses k packets in channel
- must send $n+k$ packets
- Use n point values to construct an $n-1$ degree polynomial

Erasure Coding Scheme:

1. n packet message: $m_0, m_1, ..., m_{n-1}$
2. Choose prime $p \approx 2^b$ for mod space where each packet has b bits
3. $p > n + k$
4. $P(x) = m_{n-1}x^{n-1} + ... + m_0 \pmod{p}$
5. Send, $P(1), ..., P(n+k)$

Any n of the $n+k$ packets gives polynomial and the entire message (all coefficients or y-values)

Erasure Coding Example:
Sending
Send message 1, 4, 4 (3 packets, 2 bits)
Make $P(x)$: $P(1) = 1, P(2) = 4, P(3) = 4$
Try mod5 because 5 is the closest prime to $2^b = 4$, but only gives 5 possible shares, so work mod7
Use Lagrange Interpolation
$P(x) = 2x^2 + 4x + 2 \bmod 7$
Send $(0, P(0))(1, P(1))...(6, P(7))$: 6 points
Receieving
Retrieve $P(x)$ using Lagrange or system of linear equations
Need to know which x-value the correct packets correspond to

Error Correction

- Need to recover information sent AND which packets are corrupted
- Send $n + 2k$ packets because if k errors exist, multiple original messages are possible if $< n + 2k$ packets sent.

Reed-Solomon Code:

1. Encoding polynomial $P(x)$ of degree n-1
   - $P(1) = m_1, ..., P(n) = m_n$
   - Can encode with packets as coefficients (check HW6)
2. Use **Lagrange Interpolation** to get $P(x)$
3. Send $(P1), ..., P(n + 2k)$
4. After noisy channel, receive $R(1), ..., R(n + 2k)$
5. $P(i) = R(i)$ for at least $n + k$ points i; $P(i) \neq R(i)$ for k points
6. Do not know where errors occurred
7. $P(x) =$ unique degree $n - 1$ polynomial

Error Locator Polynomial: $\boldsymbol{E(x) = (x - e_1)(x - e_2) \cdots (x - e_k)}$

- Errors at points $e_1, ...e_k$; E(i) = 0 iff $e_j = i$ for some j; $E(x)$ has degree k
- Idea: Multiply equation i by $E(x) = (x - i)$ iff $P(i) \neq R(i)$, but this creates $n + 2k$ **non-linear** equations with n_k unknowns.
- **Solution:** Let $Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots + a_0$
  - Now you have $n + 2k$ linear equations $Q(i) = R(i)E(i)$
  - **Find $E(x)$ and $Q(x)$**
    - $E(x) = x^k + b_{k-1}x^{k-1} \vdots b_0$ w/ k unknown coefficients
    - $Q(x) = a_{n+k-1}x^{n+k-1} + \cdots + a_0$ w/ $n + k$ unknown coefficients
    - Solve for coefficients of $Q(x)$ and $E(x)$; Total Unknowns: $n + 2k$
  - $\boldsymbol{P(x) = Q(x)/E(x)}$

Brute force: BAD

- Remove every possible combination of k received packets one at a time and form a degree $n + k - 1$ polynomial with remaining $n + k$ points. First consistent solution gives the corrupted packet.
- Runtime: $(n/k)^k$: exponential in k with $\binom{n+2k}{k}$ possibilities

RS Code Example:
Problem:

- Message 3,0,6 : tolerate $k = 1$ errors (send $n + 2k = 5$ packets)
- Lagrange Encoding $P(x) = x^2 + x + 1 \pmod 7$
- Send: $P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3$
- Receive: $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

Solution: Berklekamp-Welsh Algorithm

- $Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$
- $E(x) = x - b_0$
- $Q(i) = R(i)E(i)$

$$\begin{aligned}
a_3 + a_2 + a_1 + a_0 &\equiv 3(1 - b_0) \pmod 7 \\
a_3 + 4a_2 + 2a_1 + a_0 &\equiv 1(2 - b_0) \pmod 7 \\
6a_3 + 2a_2 + 3a_1 + a_0 &\equiv 6(3 - b_0) \pmod 7 \\
a_3 + 2a_2 + 4a_1 + a_0 &\equiv 0(4 - b_0) \pmod 7 \\
6a_3 + 4a_2 + 5a_1 + a_0 &\equiv 3(1 - b_0) \pmod 7
\end{aligned}$$

- Gaussian Elimnation: $a_3 = 1, a_2 = 6, a_1 = 6, a_0 = 5$; $b_0 = 2$
- $Q(x) = x^3 + 6x^2 + 6x + 5$
- $E(x) = x - 2$
- Polynomial Long Division: $P(x) = Q(x)/E(x) = x^2 + x + 1 \pmod 7$

$$\begin{array}{r|l}
x - 2 & \begin{array}{r} x^2 + 8x + 22 \\ \hline x^3 + 6x^2 + 6x + 5 \\ -x^3 + 2x^2 \\ \hline 8x^2 + 6x \\ -8x^2 + 16x \\ \hline 22x + 5 \\ -22x + 44 \\ \hline 49 \end{array}
\end{array}$$

- **Message = 3,0,6**
- RS Code: $P(x) = x^2 + x + 1 \pmod 7$ where $P(1) = 3, P(2) = 0, P(3) = 6$

CS70 - Lecture 12 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Gerald Zhang

Berklekamp-Welsh Algorithm

Existence:

- Exists because packets constructed using $P(x)$

Unique:

- Proved assuming $\frac{Q'(x)}{E(x)} = \frac{Q(x)}{E'(x)} = P(x$
- $E(x)$ and $E'(x)$ have at most k roots each
- Cross multiply assumption at n valid points, so claim is true for n points, which make $P(x)$ a unique $< n$ degree polynomial
- **Fact:** $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of x
  - Holds when $E(x)$ or $E'(x)$ are 0
  - Use above method of cross multiplication when not zero

Encoding/Decoding Polynomial Summary:

Summary: polynomials.

Set of $d+1$ points determines degree d polynomial.

Encode secret using degree $k-1$ polynomial:
Can share with n people. Any k can recover!

Encode message using degree $n-1$ polynomail:
n packets of information.

Send $n+k$ packets (point values).
Can recover from k losses: Still have n points!

Send $n+2k$ packets (point values).
Can recover from k corruptionss.
Only one polynomial contains $n+k$
Efficiency.
Magic!!!!
Error Locator Polynomial.

Relations:
Linear code.
Almost any coding matrix works.
Vandermonde matrix (the one for Reed-Solomon)..
allows for efficiency. Magic of polynomials.

Counting

- Counting Numbers: 0,1,2... all Natural Numbers $\mathbb{N}$
- **Countable if there is a bijection between S and some subset of $\mathbb{N}$**
- if subset of $\mathbb{N}$ = finite, S has finite **cardinality**
- if subset of $\mathbb{N}$ = infinite, S is **countably infinite**
  - Evens are countably infinite
  - Integers are countably infinite
  - Pairs of Natural Numbers are countably infinite
  - Rationals are countably infinite (subset of pairs of natural numbers with gcd of 1)
  - Reals are uncountable
- All countably infinite sets have the same cardinality

Isomorphism Principle:

- If there is $f : D \to R$ that is one to one and onto, (bijective) then $|D| = |R|$

Listing:

- A bijection with a subset of natural numbers
- The nth item in the list is a mapping $n \in \mathbb{N} \to f(n)$
- If you can list a set you can show a bijection
- Finite List: Bijection with subset of $\mathbb{N}\{0, ..., |S| - 1\}$
- Infinite List: Bijection with $\mathbb{N}$

Enumerating $\equiv$ Countability = Listing:

- Enumerating a set $\implies$ countability
- Corollary: Any subset T of a countable set S is also countable
- Each element of $x \in S$ has a specific, finite position in a list (ex. $\mathbb{Z} = \{0, 1, -1, 2, -2, ...\}$
- Fails for integers if you list positive integers before negative integers
  - $\mathbb{Z} = \{\{0, 1, 2, ..., \}$ and then $\{-1, -2, ...\}\}$
  - -1s position is not finite, because there are ∞ positive integers
  - So.. you must **interweave**

Diagonalization:

- **Diagonal Number** Number that is not in the list of $f(n)$
- Ex. Method to create diagonal number for Reals: Digit i is 7 if number i's ith digit is not 7, 6 otherwise. For every nth position on the list, at least the nth digit will be different than the diagonal number's nth digit. Contradiction because the diagonal number is real.
- Check if this creates contradicton. If diagonal number is in the questionable set, the list could not have existed and thes et is not countable.

CS70 - Lecture 13 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Gerald Zhang

Countability Summary:

- S is countable if there is a bijection between S and some subset of $\mathbb{N}$.
- If the subset of $\mathbb{N}$ is finite, S has finite cardinality.
- If the subset of $\mathbb{N}$ is infinite, S is countably infinite.
- Bijection with natural numbers $\implies$ countably infinite.
- Enumerable (listing) $\equiv$ countable.
- Subset of countable set is countable.
- All countably infinite sets have the same cardinality
- Natural numbers have finite number of digits

Digonalization:

- The set of all subsets S_i of $\mathbb{N}$ (powerset of $\mathbb{N}$ is not countable
  - Arbitrary Listing: L
  - Diagonal set D: For each index i of L, if $i \notin S_i$, put i in D, otherwise omit i
  - D is not in L by construction: D is different from each ith set S_i in L, for every i
  - D is a subset of N: every element in D is a natural number
  - L does not contain all subsets of N: Contradiction

Diagonalization Algorithm:

1. Assume that set S can be enumerated.
2. Consider an arbitrary list of all the elements of S.
3. Use the diagonal from the list to construct a new element t.
4. Show that t is different from all elements in the list $\implies$ t is not in the list.
5. Show that t is in S.
6. Contradiction.

Cardinalities:

Continuum Hypothesis:

- Goedel proved this hypothesis cannot be proven with math we currently know
- There is no infinite set whose cardinality is between the cardinality of an infinite set and its power set.

Uncountable Sets:

- Prove equivalence between cardinalities
- Show bijection exists between two sets: uncountable sets cannot be enumerated
- Create function $f : B \to A$ (can include multiple casesfor certain domains)
- Prove mapping is one to one by testing on arbitrary values: x, y (Need to validate for multiple cases)
  - Example: $|[0, 1]| \equiv |\mathbb{R}|$
  - $f : \mathbb{R}^+ \to [0, 1]$

Undecidability:

Russell's Paradox:

- Naive Set Theory: Any definable collection is a set.

$$\exists y \forall x (x \in y \iff P(x)) \tag{1}$$

- NST : $y =$ the set of elements that satifies $P(x)$
- Make statement: $P(x) = x \notin x$
- By NST: There exists a y that satisfies above statement for $P(x)$
- Plug in $x = y$ to NST

$$y \in y \iff y \notin y \tag{2}$$

- Methematicl system is broken, because conditions and statements are false and contradictions

HALT: DNE

- $HALT(P, I)$: P = program, I = input to program
  - Theoretically determines if $P(I)$ halts or loops forever

Halt Turing Proof:

- Assume $HALT(P, I)$ exists
- Set $P = Turing(P)$
- Use Diagonalization

```
def Turing(P):
    if(HALT(P,P)): #halts
        go into infinite loop
    else
        halt immediately
```

- Assume $Turing(Turing)$ halts
- Run $HALTS(Turing, Turing)$
  - if 'halts', Turing(Turing) 'goes into infinite loop'
  - if loops forever, Turing(Turing) 'halts immediately'
- Contradiction, so $HALT(P, P)$ does not exist

Halt Diagonlization Proof:

- Program and input are both enumerable (fixed length strings)
- Program either halts or loops on any input
- Create list: $P_i \to P_j(P)$ where $i, j \in \mathbb{N}$
- Each entry of list is arbitrarily $HALT$ or $LOOP$
- Diagonal exists, so create $Turing()$ s.t. it returns opposite values along the diagonal
- This means $Turing()$ is not in the list $\implies$ $Turing()$ is not a prgoram
  - $Turing$ is a simple function constructed from $HALT$
  - $\therefore$ $Turing()$ DNE $\implies$ $HALT()$ DNE

Undecidable Problems

-

CS70 - Lecture 14 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Gerald Zhang

Counting

Tree Counting: Slow

- Build up string by bits, total amount of leaves is total possibilities

First Rule of Counting: Product Rule:

- If objects constructed from a sequence of choices $n_1, n_2, ..., n_k$
- Total number of objects $= n_1 \times n_2 \times \cdots \times n_k$

Counting Functions/Polynomials

- There are $|T|^{|s|}$ functions $f : S \to T$
  - $|T|$ choices for mapping of $f(s_i)$ (Use product rule)
- p^{d+1} polynomials of degree d mod p
  - p choices for each of the $d + 1$ coefficients

Permutations

- Derived from the first rule of counting (product rule)
- Choose from less items each step
- Permutations of n objects: number of orderings of n objects (no replacements)
  - $n \times (n-1) \times (n-2) \times \cdots \times 1 = n!$
- Number of one to one functions $|S| \to |S|$
  - Decreasing choices every step: $|S| \times |S| - 1 \times \cdots \times 1 = |S|!$

Permutation Formula

- Number of different samples of saize k from n numbers **without replacement**

$$nPk = n \times (n-1) \times (n-2) \times \cdots \times (n-(k-1)) = \frac{n!}{(n-k)!} \tag{1}$$

Counting Sets: When order doesn't matter

Second Rule of Counting: Order Doesn't Matter (Combination):

- If order doesn't matter, count the number of ordered objects (permutations) and divide by number of orderings
- Choose k out of n possibilities

$$\binom{n}{k} = nCk = \frac{n!}{k!(n-k)!} \tag{2}$$

Sampling:

- Sample k items out of n
- Without replacement:
  - If order matters (first rule): $\frac{n!}{(n-k!)}$
  - If order does not matter (second rule): $\frac{n!}{k!(n-k!)}$
- With replacement:
  - If order matters (first rule): n^k
  - **see Stars and Bars formula (3)**

Anagrams:

- First rule on total number of letters N: $N!$ total permutations
- Divide by the number of duplicate permutations generated due to D duplicate letters: First rule: $D!$
- total distinct permutations = $\frac{N!}{A!B!\cdots D!}$ (can have multiple duplicate sets of letters)

Stars and Bars:

- Ways k people split n things
- Ways to add up k numbers to sum to n
- k undordered choices from set of n possibilities
- $\binom{\text{total} + (\text{sections - 1})}{\text{sections - 1}}$

$$\binom{n+k-1}{k-1} \tag{3}$$

Summary

First Rule (Product)

- k samples
- With replacement: n^k
- Without replacement: $\frac{n!}{(n-k)!}$

Second Rule (Division)

- When order doesn't matter (sometimes): can divide
- Without replacement (order doesn't matter): $\binom{n}{k} = \frac{n!}{(n-k)!k!}$ n choose k
  - You pick a different object every time. The total amount of orderings for your k objects is $k!$, so divide sample without replacement by $k!$ because order doesn't matter

One-to-one Rule

- Equal in number if one-to-one (Bijection)
- With replacement (order doesn't matter): $\binom{k+n-1}{n-1}$

CS70 - Lecture 15 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Gerald Zhang

Review Turing and Halt

- Turing(P) includes Halt(P,P)
- If halts (infinite loop), if doesn't halt, halt (diagonalization)
- Halt program exists ⇒ Turing program exists
- Turing("Turing") interpret program 'Turing' as text
  - Neither halts nor loops (diagonlized statement on itself)
  - Therefore, Turing program DNE
  - No Turing program ⇒ No Halt program (Contrapositive)

Review Stars and Bars

Wikipedia: Stars and Bars

- Choose n from k with replacement, order doesn't matter
- Stars and Bars Bijection (Counting rule)
- Place n (total) objects in k bins
- k bins are distinguishable, objects are not
- $k-1$ bars represent k bins
- Thm 1 (positive nums): Each bin must contain an object, there can only be ≤ 1 bar between each star. You must choose $k-1$ bars from the $n-1$ available positions.
- Thm 2 (non-negative nums): Bins can contain any no. of objects, there can be $\geq k-1$ bars between each star. You must choose $k-1$ bars from the $n+(k-1)$ available positions.

With Replacement/Order Doesn't Matter:

$$\text{Positive Groups} = \binom{n-1}{k-1} \tag{1}$$

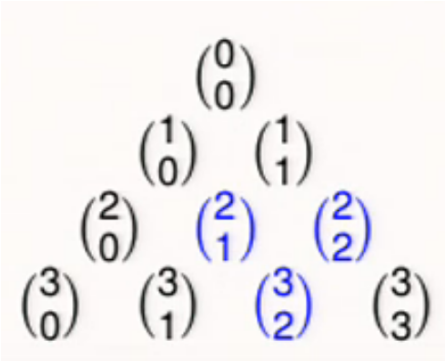
$$\text{Non-Negative Groups} = \binom{n+(k-1)}{k-1} \tag{2}$$

Combinatorial Proofs

- Define what the Left side counts and what the Right side counts, then equate them
- Ask same question for both sides and answer them using the correct approach corresponding to the side

- $\binom{n}{k} = \binom{n}{n-k}$
  - Left: Ways to choose k out of n items
  - Right: Ways to (not) choose $n-k$ out of n items, which is the same as choose k out of n items
- $\binom{n}{k} = \binom{n-1}{k-1} + \cdots + \binom{k-1}{k-1}$
  - Left: Ways to choose k out of n items
  - Right: Sum number of subsets that include the first i items and the number of subsets that do not include the first i items
- $2^n = \binom{n}{kn} + \binom{n}{n-1} + \cdots + \binom{n}{0}$: **Binomial Thm**
  - Sum of coefficients of an $(1+x)^n$ binomial (nth row in Pascal's Triangle)
  - Left: No. of subsets of n choices (element i is either in or out of the subset, 2 poss.)
  - Right: Sum of $\binom{n}{i}$ from i to n
    - $\binom{n}{i}$ ways to choose i elements of n choices

Pascal's Triangle



- Row n = coefficients of $(1+x)^n$
- Choose 2^n terms: 1 or x from $(1+x)$
  - Combine all terms corresponding to x^k
  - Coefficient of x^k is $\binom{n}{k}$: you choose k factors (products) that include x and there are n x's to choose from

Pascal's Rule:

- Left: No. of k subsets from $n+1$ choices
- Right: No. of subsets that choose the first item + No. of subsets that do not choose the first item = Left
  - $\binom{n}{k-1}$: No. of subsets of n that contain the first item. (Take away first item, left with n items and $k-1$ remaining choices)
  - $\binom{n}{k}$: No. of k subsets of n that do not contain the first item. (Take away first item, left with n items, but still need to make k choices.

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} \tag{3}$$

Simple Inclusion/Exclusion

- **Sum Rule: For disjoint sets** S and T : $|S \cup T| = |S| + |T|$
- **Inclusion/Exclusion Rule: For any** S and T : $|S \cup T = |S| + |T| - |S \cap T|$
  - Ex. No. of 10 digit phone numbers that have 7 as the first or second digit
  - S = first digit 7. $|S| = 10^9$
  - T = second digit 7. $|T| = 10^9$
  - $S \cap T$ = first and second digit. $|S \cap T| = 10^8$
  - $|S| + |T| - |S \cap T| = 10^9 + 10^9 - 10^8$

Probability Space

- Random Experiment: Define possible outcomes and likelihoods (percentages) have Statistical regularity
- Set of Ω outcomes: (Ex. $\Omega = \{H, T\}$)
  - Probabilities assigned to each outcome: $\Pr[H] = 0.5, \Pr[T] = 0.5$
  - Elements of Ω describes one outcome of the complete experiment
- Assign probability to each outcome $\Pr[A]$
  - Probabilities assigned to each outcome: Ex. $\Pr[H] = 0.5, \Pr[T] = 0.5$

- Ω = sample space (can be countable or uncountable)
- $\omega \in \Omega$ = sample point
- probability $\Pr[\omega]$ s.t. $0 \leq \Pr[\omega] \leq 1$ and $\sum_{\omega \in \Omega} \Pr[\omega] = 1$

Uniform Probability Space

- Each outcome ω is equally probable: $\Pr[\omega] = \frac{1}{\Omega}$ for all ω
- Ω must be finite

Non-Uniform Probability Space

- Each outcome ω is any $\Pr[\omega]$ s.t. $0 \leq \Pr[\omega] \leq 1$ and $\sum_{\omega \in \Omega} \Pr[\omega] = 1$

CS70 - Lecture 16 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Gerald Zhang

Set Notation Review

- Set A, Complement $\bar{A}$
- Union (In either: or): $A \cup B$
- Intersection (In both: and): $A \cap B$
- Difference (In A, not B) $A \setminus B$
- Symmetric Difference (In only one: xor) $A \Delta B$

Probability

- event E = subset of outcome: $E \subset \Omega$
- Any Sample Space: $\Pr[E] = \sum_{\omega \in E} \Pr[\omega]$
- Uniform Space: $\Pr[E] = \frac{|E|}{|\Omega|}$
- $p_n := \Pr[E_n] = \frac{|E_n|}{|\Omega|}$
  - $p_n := \frac{\binom{n}{k}}{|\omega|^n}$ if $E = n$ coin tosses with exactly k heads

Stirling Formula: (for large n)

- $n! \approx \sqrt{2\pi n}(\frac{n}{e})^n$
- $\Pr[E] = \frac{|E|}{|\Omega|}$
  - Can apply Stirling Formula because $|E|$ and $|\Omega|$ are defined by combinations (factorials)

Probability is Additive

- If events A and B are disjoint, then sum probabilities
- Non-disjoint sets, use Inclusion/exclusion property: $\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$
- **Union bound**: $\Pr[A_1 \cup \cdots \cup A_n] \leq \Pr[A_1] + \cdots + \Pr[A_n]$
- If $A_1, ..., A_N$ are a pairwise disjoint partition of Ω and $\cup_{m=1}^{N} A_m = \Omega$, then $\Pr[B] = \Pr[B \cap A_1] + \cdots + \Pr[B \cap A_N]$

Inclusion/Exclusion Property:
$\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$
Union Bound:
$\Pr[A_1 \cup \cdots \cup A_n] \leq \Pr[A_1] + \cdots + \Pr[A_n]$
Law of Total Probability:
If $A_1, ..., A_N$ are a pairwise disjoint partition of Ω and $\cup_{m=1}^{N} A_m = \Omega$ then,
$\Pr[B] = \Pr[B \cap A_1] + \cdots + \Pr[B \cap A_N]$

Conditional Probability

- Probability of A given B
- $\Pr[A|B] = \frac{\Pr[A \cap B]}{\Pr[B]}$

Product Rule

$$\Pr[A_1 \cap \cdots \cap A_n] = \Pr[A_1]\Pr[A_2|A_1] \cdots \Pr[A_n|A_1 \cap \cdots \cap A_{n-1}] \tag{1}$$

Total Probabability $\times$ Product Rule

$$\Pr[B] = \Pr[A_1]\Pr[B|A_1] + \cdots + \Pr[A_N]\Pr[B|A_N] \tag{2}$$

Causality vs. Correlation

- Events A and B are **positively correlated** if $\Pr[A \cap B] > \Pr[A]\Pr[B]$, but this does not imply causation
- Eliminate external/common causes to test causality

Bayes Rule

- Let m = number of situations where A and B occurred, and n = number of situations where $\bar{A}$ and B occurred.
- Therefore: $\Pr[A|B] = \frac{m}{m+n}$

Bayes Rule (Simplified using Law of Total Probability)

$$\Pr[A_n|B] = \frac{\Pr[A_n]\Pr[B|A_n]}{\sum_m \Pr[A_m]\Pr[B|A_m]} = \frac{\Pr[A_n]\Pr[B|A_n]}{\Pr[B]} \tag{3}$$

Independence

- Two events A and B are independent if $\Pr[A \cap B] = \Pr[A]\Pr[B]$
- Two events A and B are independent if and only if $\Pr[A|B] = \Pr[A]$
- $\Pr[A]$ decreases/increases given B

If A and B are **independent** sets:

$$\Pr[\bar{A} \cap \bar{B}] = 1 - A \cup B \tag{4}$$

$$\Pr[A \cap B] = \Pr[A] + \Pr[B] - \Pr[A \cup B] \tag{5}$$

CS70 - Lecture 17 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Gerald Zhang

Review

- Example: $B \subset A \Rightarrow A$ and B are positively correlated
  - $\Pr[A|B] = 1 > \Pr[A]$ and $\Pr[A \cap B] = \Pr[B] > \Pr[A]\Pr[B]$
- Example: $A \subset B = \emptyset \Rightarrow A$ and B are negatively correlated
  - $\Pr[A|B] = 0 < \Pr[A]$ and $\Pr[A \cap B] = 0 < \Pr[A]\Pr[B]$
- For uniformly distributed probability space Ω, $\Pr[A] = \frac{|A|}{|\Omega|}$

Probability of A given B:

$$\Pr[A|B] = \frac{\Pr[A \cap B]}{\Pr[B]} \tag{1}$$

Probability of A and B (intersection):

$$\Pr[A \cap B] = \Pr[B]\Pr[A|B] = \Pr[A]\Pr[B|A] \tag{2}$$

A and B are positively correlated if:

$$\Pr[A|B] > \Pr[A] \quad , \quad \Pr[A \cap B] > \Pr[A]\Pr[B] \tag{3}$$

A and B are negatively correlated if:

$$\Pr[A|B] < \Pr[A] \quad , \quad \Pr[A \cap B] < \Pr[A]\Pr[B] \tag{4}$$

A and B are independent iff:

$$\Pr[A|B] = \Pr[A] \quad , \quad \Pr[A \cap B] = \Pr[A]\Pr[B] \tag{5}$$

Find prior probability given some observation B (A given B)

1. Total probability of B given prior probabilities
   - **Law of Total probability**
   - $\Pr[B] = \Pr[A_1]\Pr[B|A_1] + \cdots + \Pr[A_n]\Pr[B|A_n]$
2. Find $\Pr[A|B]$
   - **Bayes Rule**
   - $\Pr[A|B] = \frac{\Pr[A]\Pr[B|A]}{\Pr[B]}$

Terms

- **Most likely A Posteriori (MAP) of** B: The A_m that gives the highest $\Pr[A_m]\Pr[B|A_m]$
- **Maximum Likelihood Estimate (MLE) of** B: The A_m that gives the highest $\Pr[B|A_m]$

Mutual Independence

- A subset of events $A_1, ..., A_k$ where $A_k, k \in J$ are **mutually independent** if the probability that they all occur is equal to the product of their individual probabilities

Mutual Independence
Definition

$$\Pr[\cap_{k\in K} A_k] = \prod_{k\in K} \Pr[A_k], \text{ for all finite } K \subseteq J \tag{6}$$

Theorem

- If the events $\{A_j, j \in J\}$ are mutually independent, and if K_n are pairwise disjoint finite subsets of J, then all the events $\cap_{k\in K_n} A_k$ are independent (same is true if we replace some of the A_k by $\bar{A}_k$

Collision Calculation
Let m = no. of elements, n= no. of bins, C = collision

$$\Pr[\bar{C}] \approx e^{(-\frac{m^2}{2n})} \tag{7}$$

When $m = 1.2\sqrt{n}$

$$\Pr[C] \approx \frac{1}{2} \tag{8}$$

Collision Derivation
If A_i = no collision when the ith ball is placed in a bin

$$\Pr[A_i | A_{i-1} \cap \cdots \cap A_1] = 1 - \frac{i-1}{n} \tag{9}$$

No collisions = $A_1 \cap \cdots \cap A_m$
Product Rule:

$$\Pr[A_1 \cap \cdots \cap A_m] = \Pr[A_1]\Pr[A_2|A_1]\cdots\Pr[A_m|A_1 \cap \cdots \cap A_{m-1}] \tag{10}$$

Apply to $\Pr[\bar{C}]$:

$$\Pr[\bar{C}] = (1 - \frac{1}{n})\cdots(1 - \frac{m-1}{n}) \tag{11}$$

Natural log of both sides:

$$\ln(\Pr[\bar{C}]) = \sum_{k=1}^{m-1} \ln(1 - \frac{k}{n}) \approx \sum_{k=1}^{m-1} \ln(-\frac{k}{n})^{*} = (-\frac{1}{n})(\frac{m(m-1)}{2}) \approx -\frac{m^2}{2n} \tag{12}$$

* Use property that $\ln(1-\varepsilon) \approx -\varepsilon$ for $|\varepsilon| << 1$
Gauss Summation: $1 + 2 + \cdots + m - 1 = \frac{m(m-1)}{2}$

Example: Checksums

- m = no. of files, b = no. of bits in the checksum, C = files share a checksum
- Find b s.t. $\Pr[C] \leq 10^{-3}$
  - * $\Pr[C] \approx 1 - e^{(-\frac{m^2}{2(2^b)})}$
  - * $b = \frac{\ln(-\frac{m^2}{2\ln(1-10^{-3})})}{\ln(2)} = 2.9\ln(m) + 9$
- $\therefore b \geq 2.9\ln(m) + 9$

Probability of Getting n_i out of n with m picks

- Define event of failure A_m (not success)
- Determine probability of failing on each iteration of m
  - $\Pr[A_i|A_{i-1} \cap \cdots \cap A_1] = 1 - \Pr[\bar{A}_i]$ for $i = \{1, ..., m\}$)
  - If not intuitive, try brute force and find a pattern for each $\Pr[A_i]$
- Use Product Rule to get $\Pr[A_m]$
  - $\Pr[A_1 \cap \cdots \cap A_m] = \Pr[A_1]\Pr[A_2|A_1] \cdots \Pr[A_m|A_1 \cap \cdots \cap A_{m-1}]$
  - If events are **independent** $\Pr[A_1 \cap \cdots \cap A_m] = \Pr[A_1]\Pr[A_2] \cdots \Pr[A_m|A_{m-1}]$
- Take natural log of both sides and simplify using the property that $\ln(1 - \varepsilon) \approx -\varepsilon$ for $|\varepsilon| << 1$
- Raise e to the power of both sides (e^n) to derive approximate solution for $\Pr[A_m]$
  - $\Pr[A_m] \approx e^{expression}$

Probability of Complete Collection

- Define event of failure of one iteration E_k
  - E_k for $k = \{1, ..., n\}$
  - Derive $\Pr[E_k]$ using method above: **Probability of Getting n_i out of n with m picks**
- find probability of failing any iteration (or/union)
  - $p := \Pr[E_1 \cup E_2 \cup \cdots \cup E_n]$
- Estimate p using Union Bound
  - $p := \Pr[E_1 \cup E_2 \cup \cdots \cup E_n] \leq \Pr[E_1] + \Pr[E_2] + \cdots + \Pr[E_n]$
- Plug in $\Pr[E_k]$ expression derived above to find Pr[failure of at least one iteration] $\leq expression$
- Use expression to derive minimum value of m to get a certeain $\Pr[miss]$ s.t. $\Pr[miss] \leq p$

CS70 - Lecture 18 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Gerald Zhang

Random Variables

- Random variable is a known, deterministic, function that maps outcome to a number (onto, but not necessarily one-to-one)
- Random Variable X for an experiment with sample space Ω is a function $X : \Omega \to \mathbb{R}$
  - X assigns $X(\omega) \in \mathbb{R}$ to each $\omega \in \Omega$
  - X is not random, nor a variable, it is called a random variable, because its outcome depends on the intial probability/experiment (varies from experiment to experiment)
  - After deriving X and knowing the initial probabilities, can determine the likelihood of each outcome of X
- $X^{-1}(A) = \{\omega | X(\omega) = A\}$: Inverse image of value A
  - Set of outcomes that map to A

Distribution

- The probability of X taking on a value A.
- Definition: The distribution of a random variable X is $\{(a, \Pr[X = a]) : a \in \mathbb{A}\}$ where $\mathbb{A}$ is the range of X
- $\Pr[X = A] = \Pr[X^{-1}(A)]$

Random Variable Method

- Examine the elements of the Sample Space (ex. $\{HHH, THH, HTH, TTH, HHT, THT, HTT, TTT\}$
- Define the Random Variables by given params ex. +1 on heads, -1 on tails $\Rightarrow \{3, 1, 1, -1, 1, -1, -1, -3\}$
- Determine the probability of any generalized event (ex. getting i heads and $n - i$ tails if the prob. of getting heads is p: $p^i(1-p)^{n-i}$)
- Multiply the above probability by the amount of times it occurs (summation) (ex. All ways to get i heads out of n flips: $\binom{n}{i}$)
- This determines the probability of each element in the Random Variable: the distribution of outcomes. (ex. Binomial Distribution)

Binomial Distribution

$$B(n, p) : \Pr[X = i] = \binom{n}{i} p^i (1-p)^{n-1}, i \in \{0, \cdots, n\} \tag{1}$$

- Flip n coins with probability p to get heads, Random var: No. of heads
- Ways to choose i heads out of n flips: $\binom{n}{i}$
- Determine probability of $\omega = i$ heads (probability of heads in any position is p): Pr of i heads and $n-1$ tails is p^i and $(1-p)^{n-i}$ respectively $\therefore \Pr[\omega] = p^i(1-p)^{n-i}$
- Probability of $X = i$ is the sum of all $\Pr[\omega]$ where ω contains i heads: $\binom{n}{i}$

Error Channel

- Apply Binomial Distribution
- Packet is corrupted with probability p
- Send $n + 2k$ packets
- Find probability of at most k corruptions
- $\sum_{i \leq k} \binom{n+2k}{i} p^i (1-p)^{n+2k-i}$: Sum gets total probability of all i corruptions s.t. $i \leq k$
- For RS Code, choose k s.t. the above probability is large

Combining Random Variables

- Let X and Y be two Random Vars in the same Probability space
- Then, $X + Y$ is an RV that assigns value $X(\omega) + Y(\omega)$ to ω
- General Case: $g(X, Y, Z)$ assigns value $g(X(\omega), Y(\omega), Z(\omega))$ to ω

Expectation

- $E[X] = \sum_a a \times \Pr[X = A] \approx \frac{X_1 + \cdots + X_n}{N}$
- Random variable X, X has a possible values (gains). Multiply each possible gain a by the probability of RV $X = a$ and sum over all RVs to get expected value.
- Average $= E(X)$ holds for uniform probability
- Expectation is linear

Expectation Theorem

$$E[X] = \sum_{\omega} X(\omega) \times \Pr[\omega] \tag{2}$$

- Sum of all products of X_i and the probability of getting that X_i (also called the mean by frequentist interpretation): **Law of Large Numbers**

Expectation Derivation Methods

- Two ways to compute the mean value
  - Given: Distribution of X (set of values a and their probabilities)
    - $E[X] = \sum_a a \times \Pr[X = A] \approx \frac{X_1 + \cdots + X_n}{N}$
  - Given: Probability Space
    - Sum over all ω's in probability space
    - $E[X] = \sum_{\omega} X(\omega) \times \Pr[\omega]$

Example of Both Derivation Methods

- Flip fair coin 3 times
- $\Omega = \{HHH, HHT, HTH, THH, HTT, THT, TTH, TTT\}$
- $X =$ number of H's $: \{3, 2, 2, 2, 1, 1, 1, 0\}$
- Method 1: $E[X] = \sum_a a \times \Pr[X = A] = 3(\frac{1}{8}) + 2(\frac{3}{8}) + 1(\frac{3}{8}) + 0(\frac{1}{8}) = \frac{3}{2}$
- Method 2: $E[X] = \sum_{\omega} X(\omega) \times \Pr[\omega] = (3 + 2 + 2 + 2 + 1 + 1 + 1 + 0)(\frac{1}{8})$