EECS 376 Discussion 11

Sec 27: Th 5:30-6:30 DOW 1017

IA: Eric Khiu

Slide deck available at course drive/Discussion/Slides/Eric Khiu

Starter: Matching Pennies

- Consider a game with two players Alice and Bob
- Each player has a penny and choose heads or tails
  - Alice wins the round if both choose the same outcome
  - Bob wins the round if both choose different outcome

		Alice	
		H	T
Bob	H	Alice wins	Bob wins
	T	Bob wins	Alice wins

- They will play the game for 10 rounds the final winner is whoever wins the most rounds
- Consider the following algorithms:
  - Here, RAND(S) is a function that output a random element in set S

```
ALG1 (roundNum):
    if roundNum is odd then return H
    else return T
```

```
ALG2 (roundNum):
    num ← RAND({0,1})
    if num is odd then return H
    else return T
```

Discuss: If you were Alice, which algorithm would you choose and why?

Unit 4: Randomness in Computation

Motivation: Randomness in computation

- The algorithms we have seen thus far have been deterministic
  - Execute the same steps each time they are run and produce the same result
- If we use deterministic algorithm in Matching Pennies, the opponent would be able to observe the program's strategy once and defeat it every single time thereafter
  - How to prevent the opponent from predicting our moves? Make moves randomly!
- In this unit, we consider how randomness can be applied to computation
- We will start with reviewing/ introducing some tools to analyze randomness

Agenda

- Tools for analyzing randomized algorithms
- Markov’s inequality
- Modular arithmetic review (if time)

Tools for Analyzing Randomness

Warning: This section contains a lot of math

Course notes

Expected Values

- Let X be a discrete random variable (RV) over the set of events Ω, each with some probability in range $[0,1]$
- The **expected value** of X is

$$E[X] = \sum_{\omega \in \Omega} \omega \cdot \Pr[X = \omega]$$

- Example: Consider a fair 6-sided die with RV D being the result of the roll.

$$\Pr[D = 1] = \Pr[D = 2] = \cdots = \Pr[D = 6] = \frac{1}{6}$$

$$E[D] = 1\left(\frac{1}{6}\right) + 2\left(\frac{1}{6}\right) + \cdots + 6\left(\frac{1}{6}\right) = \frac{7}{2}$$

Linearity of Expectations

- Let X_1 and X_2 be two RVs and $X = c_1X_1 + c_2X_2$, then
$$E[X] = c_1E[X_1] + c_2E[X_2]$$
- More generally, if we have RVs $X_1, \dots, X_n$ and $X = c_1X_1 + \cdots + c_nX_n$, then
$$E[X] = c_1E[X_1] + \cdots + c_nE[X_n] = \sum_{i=1}^{n} c_iE[X_i]$$
- **Exercise:** Let X_1 be the result of a fair coin toss where $X_1 = 1$ if heads and $X_1 = 0$ if tails; X_2 be the results of a fair six-sided die roll. What is the expected value of $X = X_1 + X_2$?
  - $E[X_1] = 0\left(\frac{1}{2}\right) + 1\left(\frac{1}{2}\right) = \frac{1}{2}$, $E[X_2] = \frac{7}{2}$ from previous
  - By linearity of expectation, $E[X] = \frac{1}{2} + \frac{7}{2} = 4$

Indicator Random Variable

- An **indicator RV** for an event A is defined as follows:

$$\mathbb{1}_A = [\![A]\!] = \begin{cases} 1 & \text{if } A \text{ happens} \\ 0 & \text{otherwise} \end{cases}$$

- Consider an event A that happens with probability $\Pr[A]$. Let X be an indicator random variable for A. What is $E[X]$?

$$E[X] = 1 \cdot \Pr[X = 1] + 0 \cdot \Pr[X = 0] = \Pr[A]$$

- If X is a discrete RV, it is sometimes useful to write $X = X_1 + \cdots + X_n$ to compute $E[X]$

Discuss: Intuitively, why do you think this is the case?

- Linearity of expectation! $E[X] = E[X_1] + E[X_2] + \cdots + E[X_n]$

Example: Are you a *peak*?

- Take integers $1, \dots, n$ and permutate them randomly as a sequence $a_1, \dots, a_n$. We say a_i is a *peak* if it is greater than all previous numbers, i.e., $a_i > a_j$ for all $j < i$. For example:

  $$\underline{2}, 1, \underline{3}, \underline{5}, 4 \rightarrow \text{three peaks}$$

- Let X be the number of peaks in the sequence. Find $E[X]$. You may leave your answer as a sum without simplifying it.
  - Let X_i be an indicator RV such that $X_i = 1$ if a_i is a peak, 0 otherwise
  - **Obs:** $\Pr[X_1 = 1] = 1$ (no previous), $\Pr[X_2 = 1] = 1/2$ (either $a_2 > a_1$ or $a_2 < a_1$)
  - In general, a_i is a peak $\Rightarrow a_i = \max\{a_1, \dots, a_i\}$, since all i numbers are distinct and only one max, so $\Pr[X_i = 1] = \Pr[a_i \text{ is } \textit{that} \max] = 1/i$
  - $E[X] = E[X_1] + \cdots + E[X_n] = \Pr[X_1 = 1] + \cdots + \Pr[X_n = 1] = \sum_{i=1}^{n} \frac{1}{i}$

Exercise: Increasing Subarray

- Let A be a array of length n of a random permutation of n distinct integer. Compute the expected number of increasing subarrays in A of length k.
  - Hint: First define an indicator RV that consider whether a particular subarray of length k is increasing, then determine that probability
  - Let $X_i = 1$ if $A[i, \ldots, i+k-1]$ is increasing and 0 otherwise
  - Since we only consider subarrays of length k, set $X_i = 0$ for $i = n-k+2, \ldots, n$
  - For any array of length k, since all k numbers are distinct, we $k!$ permutations, but only one is increasing, so $\Pr[X_i = 1] = \Pr[A[i, \ldots, i+k-1] \text{ is } \textit{that} \text{ increasing permutation}] = 1/k!$
  - $E[X] = E[X_1] + \cdots + E[X_{n-k+1}] = \sum_{i=1}^{n-k+1} \frac{1}{k!} = \frac{n-k+1}{k!}$

Recap: Approximation Algorithms

- We can define how *good* an approximation is in terms of an approximation ratio α
  - Let $val(y)$ be a function that maps the output of a function to some value
  - Let OPT be the value of an optimal solution for some search problem
- An approximate solution y is said to be an **α-approximation** if

$$\alpha \cdot OPT \leq val(y) \text{ for maximization problem}$$

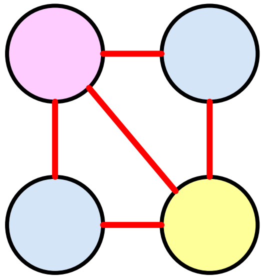
$$val(y) \leq \alpha \cdot OPT \text{ for minimization problem}$$

Discuss: Can we prove that the output of a randomized algorithm is an α-approximation?

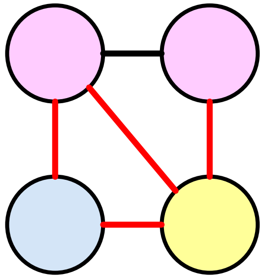
- Yes, but only in expectation-sometimes we got unlucky/ lucky and exit the bound!
  - Use $E[val(y)]$ instead of $val(y)$

Example: 3-painting

- In an undirected graph, a *3-painting* is an assignment of one of three colors to each vertex. (Adjacent vertices do not *necessarily* need to have different colors). Given a 2-painting of an undirected graph, and edge is called *colorful* if its endpoints are assigned different colors.



#colorful edges = 5



#colorful edges = 4

Example: 3-painting

- In an undirected graph, a *3-painting* is an assignment of one of three colors to each vertex. (Adjacent vertices do not *necessarily* need to have different colors). Given a 2-painting of an undirected graph, and edge is called *colorful* if its endpoints are assigned different colors.
- Consider the following algorithm:

```
PAINTING(G=(V,E)):
 for v in V:
     num ← RAND({1,2,3}) // uniformly choose between {1,2,3} with prob. 1/3 each
     if num = 1 then v.color ← pink
     else if num = 2 then v.color ← blue
     else v.color ← yellow
```

- Prove that PAINTING is 2/3 approximation in expectation.
  - Hint: First compute $E[val(y)]$, then prove the bound

Example: 3-painting

```
PAINTING(G=(V,E)):
    for v in V:
        num ← RAND({1,2,3}) // uniformly choose between {1,2,3} with prob. 1/3 each
        if num = 1 then v.color ← pink
        else if num = 2 then v.color ← blue
        else v.color ← yellow
```

- Step 1: Compute $E[val(y)]$
  - For each $e \in E$, let X_e be an indicator RV such that $X_e = 1$ if e is colorful and 0 otherwise
  - For each e, there are $3 \cdot 3 = 9$ possible paintings, 3 of them have same colors on both ends (6 of them have different colors), so $\Pr[X_e = 1] = \frac{6}{9} = \frac{2}{3}$
  - $E[X] = \sum_{e \in E} E[X_e] = \sum_{e \in E} \Pr[X_e = 1] = \frac{2}{3}|E|$

Example: 3-painting

```
PAINTING(G=(V,E)):
    for v in V:
        num ← RAND({1,2,3}) // uniformly choose between {1,2,3} with prob. 1/3 each
        if num = 1 then v.color ← pink
        else if num = 2 then v.color ← blue
        else v.color ← yellow
```

- Step 2: Prove bound
  - Now we have $E[val(y)] = \frac{2}{3}|E|$
  - Let OPT be the optimum number of colorful edges. By definition, $OPT \leq |E|$
  - Therefore, $E[val(y)] = \frac{2}{3}|E| \geq \frac{2}{3}OPT$, as desired.

TL; DPA

- We reviewed/ introduced tools to analyze randomness: expected values, linearity of expectations, and indicator RV
- It is sometimes useful to express a discrete RV as a sum of indicator RV when computing expectations
- For randomized algorithm, use $E[val(y)]$ to prove approximation in expectation

Markov Inequality

Warning: This section also contains a lot of math

[Course notes]

Starter: Search Algo Optimization

- Suppose you are optimizing a search algorithm for a large, constantly updating database system. The user can't wait for more than 1 second in general.
- You have the following two options

Option A	Option B
• Average search time: 0.05s • Potentially take more than 2s on a search during high-demand periods	• Average search time: 0.15s • Rarely take more than 0.6s on any search even under heady load

Discuss: Which one would you choose and why? What additional information you think will help you make the decision?

- $\Pr[A \text{ takes more than } 1\text{s}]$ and $\Pr[B \text{ takes more than } 1\text{s}]$
- **Obs:** They are both probabilities that the RV deviates from the expectation by some amount

Markov Inequality

- **Motivation:** Find an upper bound on the probability that a random variable X deviates from its expected value by some amount
- **Markov's Inequality:** Let X be a positive RV and $a > 0$, then

$$\Pr[X \geq a] \leq \frac{E[X]}{a}$$

- Rearranging, we get

$$\Pr\big[X \geq a \cdot E[X]\big] \leq \frac{1}{a}$$

$$\Pr[X \geq a] \leq \frac{E[X]}{a}$$

$$\Pr\left[X \geq a \cdot E[X]\right] \leq \frac{1}{a}$$

Example: Hash Table

- Suppose we have a hash table of size n^2 and a hash function h that chooses the mapping address uniformly at random from $0, \ldots, n^2 - 1$.
- Let $S = \{s_1, \ldots, s_n\}$ be the set of inserted elements and X be the RV indicating the number of collisions after performing n insertion.
- Find an upper bound on the probability that there is at least one collision ($h(s_i) = h(s_j)$) after inserting n distinct elements. (You may use $\frac{n-1}{2n} < \frac{1}{2}$ for any $n \in \mathbb{N}$)
  - First, compute $E[X]$

$$E[X] = \sum_{\substack{\text{all pairs } (i,j) \\ i \neq j}} \Pr\left[h(s_i) = h\left(s_j\right)\right] = \sum_{\substack{\text{all pairs } (i,j) \\ i \neq j}} \frac{1}{n^2} = \binom{n}{2} \cdot \frac{1}{n^2} = \frac{n-1}{2n}$$

  - Using Markov's inequality

$$\Pr[X \geq 1] \leq \frac{E[X]}{1} = \frac{n-1}{2n} < \frac{1}{2}$$

$$\Pr[X \geq a] \leq \frac{E[X]}{a}$$
$$\Pr\left[X \geq a \cdot E[X]\right] \leq \frac{1}{a}$$

Search Algo Optimization Revisit

Option A	**Option B**
• Average search time: 0.05s • Potentially take more than 2s on a search during high-demand periods	• Average search time: 0.15s • Rarely take more than 0.6s on any search even under heady load

- Let A be the search time using option A and B be the search time using option B. Using Markov's inequality and $a = 1$, we have

$$\Pr[A \geq 1] \leq 0.05 \quad \text{and} \quad \Pr[B \geq 1] \leq 0.15$$

Discuss: Does this result change your decision?

- The upper bounds of the chance of option A taking at least 1 second is lower than that of option B- maybe A is better?
- WAIT: We haven't considered the "*rarely* take more than 0.6s"! Who knows $\Pr[B \geq 1]$ is actually 0.0001?
- **Takeaway:** Markov's inequality is a weak bound, but still applicable to many cases

“Reverse” Markov Inequality

- We can also find the lower bound on the probability that a RV X deviates from its expected value by some amount, if we know some upper bound for X
- If X is positive RV that is never larger than B and $a < B$, then
$$\Pr[X > a] \geq \frac{E[X] - a}{B - a}$$
- **Example:** Suppose we have a biased coin were $\Pr[H] = 0.3$. Find a lower bound on the probability that there are strictly more than 10 heads after 100 tosses.
  - Let X be the number of heads after 100 tosses
  - We have $E[X] = 0.3 \cdot 100 = 30$ and $B = 100$, so
$$\Pr[X > 10] \geq \frac{30 - 10}{100 - 10} = \frac{2}{9}$$

TL; DPA

- Markov’s inequality gives upper bound on the probability that a positive RV deviates from its expected value by some amount
- It is a weak bound, but applicable in many cases
- “Reverse” Markov’s inequality gives a lower bound

Modular Arithmetic Review

Warning: This section still contains a lot of math

Course notes

Modular Arithmetic Review

- Let, a, b, n be integers
- **Definition:** $a \bmod n$ is the remainder of a when divided by n
  - $a \bmod n$ is a unique value in $\mathbb{Z}_n = \{0, \dots, n-1\}$
- **Definition:** a and b are congruent modulo n, written as $a \equiv b (\text{mod } n)$ if
  - $a \bmod n = b \bmod \text{n}$, or equivalently,
  - $\exists k \in \mathbb{Z}$ such that $a = b + kn$, or equivalently
  - $a - b$ is a multiple of n
- **Modular Arithmetic:** Suppose $a \equiv b (\text{mod } n)$, $c \in \mathbb{Z}$
  - Addition: $a + c \equiv b + c \ (\text{mod } n)$
  - Multiplication: $ac \equiv bc \ (\text{mod } n)$

Division in $\mathbb{Z}_n$

- **Definition:** Let $a \in \mathbb{Z}$. $a^{-1} \in \mathbb{Z}$ is a **multiplicative inverse** of a in modulo n such that
$$a^{-1} \cdot a \equiv 1 \pmod{n}$$
  - Note: We typically standardize a^{-1} to be in $\mathbb{Z}_n$
- In modular arithmetic, dividing by a is the same as multiplying by a^{-1}
- WARNING: Division is not always possible, as a does not always have an inverse
  - For example: 2 has no inverse in $\mathbb{Z}_4 = \{0,1,2,3\}$
$$0 \cdot 2 \equiv 2 \cdot 2 \equiv 0 \pmod{4}, 1 \cdot 2 \equiv 3 \cdot 2 \equiv 2 \pmod{4}$$
- **Theorem:** An integer a has a multiplication inverse in mod n iff $\gcd(a, n) = 1$
  - **Corollary:** For all $a \neq 0 \in \mathbb{Z}_p$, where p is prime, there is a multiplicative inverse of a in modulo p. -This is key in cryptography!

Finding Multiplicative Inverse: Intuition

- Suppose we want to find multiplicative inverse of 4 in mod 7
  - By inspection, $\gcd(4,7) = 1$, so 4 has multiplicative inverse in mod 7
- By definition, we want some $b \in \mathbb{Z}$ such that
$$4b \equiv 1 \pmod 7$$
- By definition of modular congruence, $\exists k \in \mathbb{Z}$ such that
$$4b = 1 + 7k$$
- Rearranging,
$$4b - 7k = 1 = \gcd(4,7)$$
- **Obs:** b and k are coefficients of 4 and 7 in the linear combination of their gcd
- We have seen this in **Extended Euclid Algorithm**!

Extended Euclid Algorithm

function EXTENDEDEUCLID(x, y)
  if $y = 0$ **then**
    return $(x, 1, 0)$
  else
    Write $x = qy + r$ for an integer q, where $0 \le r < y$
    $(g, a', b') \leftarrow$ EXTENDEDEUCLID(y, r)
    $a \leftarrow b'$
    $b \leftarrow a' - b'q$
    return (g, a, b)

- Example: Find the multiplicative inverse of 4 in mod 7

x	y	q	r	g	$a \leftarrow b'$	$b \leftarrow a' - b'q$
7	4					

Extended Euclid Algorithm

```
function ExtendedEuclid(x, y)
    if y = 0 then
        return (x, 1, 0)
    else
        Write x = qy + r for an integer q, where 0 ≤ r < y
        (g, a', b') ← ExtendedEuclid(y, r)
        a ← b'
        b ← a' − b'q
        return (g, a, b)
```

- Example: Find the multiplicative inverse of 4 in mod 7

x	y	q	r	g	$a \leftarrow b'$	$b \leftarrow a' - b'q$
7	4	1	3			
4	3	1	1			
3	1	3	0			
1	0	-	-			

Extended Euclid Algorithm

```
function ExtendedEuclid(x, y)
    if y = 0 then
        return (x, 1, 0)
    else
        Write x = qy + r for an integer q, where 0 ≤ r < y
        (g, a′, b′) ← ExtendedEuclid(y, r)
        a ← b′
        b ← a′ − b′q
        return (g, a, b)
```

- Example: Find the multiplicative inverse of 4 in mod 7

x	y	q	r	g	$a \leftarrow b'$	$b \leftarrow a' - b'q$
7	4	1	3	1	-1	1-(-1)(1)=2
4	3	1	1	1	1	0-1(1)=-1
3	1	3	0	1	0	1-0(3)=1
1	0	-	-	1	1	0

- Observe that $4(2) - 7(-1) = 15 \equiv 1 (\text{mod } 7)$
- In fact, $4^{-1} \bmod 7 = 2$
  - Check $2 \cdot 4 = 8 \equiv 1\ (\text{mod } 7)$

Exercise

function $\textsc{ExtendedEuclid}(x, y)$
  if $y = 0$ **then**
    return $(x, 1, 0)$
  else
    Write $x = qy + r$ for an integer q, where $0 \le r < y$
    $(g, a', b') \leftarrow \textsc{ExtendedEuclid}(y, r)$
    $a \leftarrow b'$
    $b \leftarrow a' - b'q$
    return (g, a, b)

- Find $13^{-1} \bmod 21$

x	y	$q \leftarrow \lfloor x/y \rfloor$	$r \leftarrow x/y - q$	g	$a \leftarrow b'$	$b \leftarrow a' - b'q$
21	13					

Exercise

function $\textsc{ExtendedEuclid}(x, y)$
  if $y = 0$ **then**
    return $(x, 1, 0)$
  else
    Write $x = qy + r$ for an integer q, where $0 \le r < y$
    $(g, a', b') \leftarrow \textsc{ExtendedEuclid}(y, r)$
    $a \leftarrow b'$
    $b \leftarrow a' - b'q$
    return (g, a, b)

- Find $13^{-1} \bmod 21$

$-8 \bmod 21 = 13$

x	y	$q \leftarrow \lfloor x/y \rfloor$	$r \leftarrow x/y - q$	g	$a \leftarrow b'$	$b \leftarrow a' - b'q$
21	13	1	8	1	5	-3-5(1)=-8
13	8	1	5	1	-3	2-(-3)(1)=5
8	5	1	3	1	2	-1-(2)(1)=-3
5	3	1	2	1	-1	1-(-1)(1)=2
3	2	1	1	1	1	0-1(1)=-1
2	1	2	0	1	0	1-0(2)=1
1	0	-	-	1	1	0