What Did We Learn in CS70?

Yining Liu

November 2, 2018

We started with learning about logic. You learned about that"If it rains, I'll bring an umbrella" is the same as "If I don't have an umbrella with me, then it's not raining", both of which are different from "If I have an umbrella, then it's raining". [1] You also wrapped your head around $\forall$, $\exists$, and learned that you can't move them freely whenever you want. [2] You became comfortable with negating a statement: stay in the same "universe" but flip the quantifiers; negate each clause, and flip the logic symbols (which are used to connect clauses).

After making sure you're comfortable with manipulating a statement, you learned about how to *prove* a statement. Direct proof? Proof by contradiction? Proof by contraposition? Proof by cases? Proof by induction? Wisely use one or a combination of proof techniques to convince others that a statement is true.

We spent a relatively long time on learning about induction. If you want to prove a statement $P(n)$ [3] is true for all $n \in \mathbb{N}$, do 1) prove $P(0)$ is true and 2) assume $P(n)$ then show $P(n+1)$ is true. Oh, *why does induction work*? Well ordering principle. In fact, for all induction proofs, you can formulate a similar proof using WOP: just put all "bad inputs" in a set S , find the smallest element n_0 in S , show that n_0 cannot be 0, and then show n_0 cannot be in the set by showing $P(n_0 - 1) \implies P(n_0)$. Get stuck when you want to prove $P(n)$? Try giving yourself more information by using strong induction, which, by the way, is equivalent to simple induction. [4]

We are (yet-to-be) computer scientists. We care about graphs. You learned about some fast ways to check some properties of a graph. Can I find a tour that visits all edges exactly once? [5] Well, check whether all vertices have even degree. After exploring duality of graphs, you learned about the famous Euler's formula, which tells you a nice relationship between v, f and e in a connected graph: $v + f = e + 2$. If someone come to you with a graph and ask: is my graph planar? You know after counting v and e, you can confidently give the answer "no" to some of the graphs. [6] [7] [8]

[1] $P \implies Q$ is the same as $\neg Q \implies \neg P$. They're not the same as $Q \implies P$.
[2] See discussion 1a, discussion 1b.
[3] It's a good habit to always explicitly write out $P(n)$.
[4] See discussion 1d.
[5] This is called a Eulerian Tour.
[6] K_5 and $K_{3,3}$ are nonplanar.
[7] planar $\implies e \leq 3v - 6$.
[8] Bipartite + planar $\implies e \leq 2v - 4$.

We also color graphs. [9] Bipartite graphs are always 2 colorable. If you want to color the vertices of a planar graph, 4 colors is always enough! You also learned about hypercubes, which has a nice recursive definition. Ready to use induction on graphs? You know that you should start with the graph in $P(n+1)$, modify the graph to the one in $P(n)$ so you can use your inductive hypothesis. Then, add back whatever you remove to get the graph in $P(n+1)$ again and do something to show the original statement still holds. You know that we should use this procedure to avoid potential built-up errors. [10]

We then moved on to modular arithmetic. It's like a sad little world where people there try to use finite number of integers to get a number system that works similar to ours. [11] Those people actually did a great job "translating" between their numbers and ours. They can add [12], subtract [13], and multiply [14]. In most of those worlds [15], even for non-zero numbers, we can't always divide[16]. In mod m, a number a only has a multiplicative inverse if it's coprime with m. How do I check if a is coprime with m (that is, $gcd(a, m) = 1$)? Run Euclid's Algorithm. Okay. $gcd(a, m)$ is 1! How do I get $a^{-1} \pmod{m}$? Run Extended Euclid's Algorithm. [17]

We took a tiny detour to learn about bijections. We know that f^{-1} exists if and only if f is a bijection. Why did we take this detour? You know that if $gcd(a, m) = 1$, $f(x) = ax \pmod{m}$ has an inverse so it's a bijection, which is then used to prove Fermat's Little Theorem. [18] [19] You can use FLT to quickly simplify some exponents in mod p if p is prime.

That was a lot about mod. *Why* do we learn about them? One important application is RSA. You met Alice, Bob and Eve. Using RSA, Alice and Bob pick two large primes p and q, an integer e, whose inverse in mod $(p-1)(q-1)$ is used as the decryption key d. Alice wants to encrypt a message m? Just do $m^e \pmod{N}$. Bob wants to decrypt a cipher c? Just do $c^d \pmod{N}$. How do you quickly calculate exponents as N is not prime here (so cannot use FLT)? Try repeated squaring. Also, thankfully, evil Eve cannot decrypt the message in a reasonable time because factoring N is hard.

When Alice sends her message to Bob, she breaks it up into several packets. Sometimes, even without Eve, Alice might not be able to reliably send her message to Bob because the channel might erase or modify some of her packets. Before learning about how to deal with those two type of errors, you equipped yourself with a tool - polynomials. You learned that there are two ways to represent a polynomial with degree d: either using $d+1$ coefficients,

[9] Take CS170 to learn *why* we care about coloring vertices.
[10] See homework 2.
[11] Do NOT think of mod as an operation.
[12] $a \equiv b \pmod{m}, c \equiv d \pmod{m} \implies a + b \equiv c + d(\pmod{m}$
[13] Subtraction is just adding additive inverse.
[14] $a \equiv b \pmod{m}, c \equiv d \pmod{m} \implies ab \equiv cd \pmod{m}$
[15] All non-zero numbers have a multiplicative inverse in mod p
[16] Division is multiplying by multiplicative inverse.
[17] See discussion 2c.
[18] First version: p prime, $a \not\equiv 0 \pmod{p} \implies a^{p-1} \equiv 1 \pmod{p}$;
[19] Second version: p prime $\implies a^p \equiv a \pmod{p}, \forall a$.

or $d+1$ points. Those two representations are equivalent in some sense: want to translate from points to coefficients? Use Lagrange Interpolation; want to translate from coefficients to points? Evaluate the polynomial at $d+1$ points.

You played around with polynomials by learning about secret sharing. If you want to share a secret such that only k officials getting together can reveal the secret, and even $k-1$ of them together would know *nothing*, you can use polynomials! Just plot your secret at $x=0$, randomly pick $k-1$ more points so that you can uniquely construct a degree k polynomial. Then, give each official one point. When k of them getting together, they can construct the original polynomial and get the secret.

Time to get back to our original goal of using polynomials to deal with erasure errors and general errors. If your channel erases k packets, then use your n packets to get a degree $n-1$ polynomial, evaluate your polynomial at k more points and send all $n+k$ packets. It's okay if the channel erases k of them - the receiver can still construct the original polynomial back using the received packets. What if the channel *modifies* k packets? Then you need to send $n+2k$ packets [20].

By the way, have you ever wondered about stuff like *"are all infinities the same"*? If so, you are pretty lucky, since you learned about countability which hopefully answered your doubts. Some sets [21] with infinite size are countable. Want to prove a set is countably infinite? Find a way to enumerate the elements; that is, find a bijection between your set and $\mathbb{N}$. Some sets [22] are uncountable. Want to prove a set is uncountable? Use diagonalization: assume you can enumerate the elements; put them in a table and construct a new element by flipping the entries in the diagonal.

Have you wondered about other stuff like *"are all programs computable"*? It turns out that the answer is "no". One of the uncounputable program is *TestHalt*, which takes a program P and an input x, and tells us whether P halts on x. How do we show whether a program is not computable? Don't worry - you just need to show that program is "harder" than *TestHalt* [23].

Then, we nicely made an transition into probability by learning about counting [24]. How do you count things? For most of the cases, you want to divide the things you want to count into smaller problems. In general, if your smaller problems are sub-tasks, then you want to multiply; if your smaller problems are different cases, then you want to add. Balls and bins? Stars and bars? They are just ways to model the problems. Also, have you noticed that you can use two ways to count the exact same thing? Congratulations. You probably just constructed a combinatorial proof.

Let's count some useful things. When you perform an experiment, the sam-

[20] See Discussion 3b

[21] Examples of countably infinite sets: $\mathbb{N}, \mathbb{Q}$, the set of bit strings with finite length

[22] Examples of uncountabe sets: $\mathbb{R}$, the set of bit strings with infinite length

[23] See Discussion 4a for examples. See my computability notes for more explanations. Essentially, you want to reduce *TestHalt* to that program by using that program to implement *TestHalt*.

[24] See Discussion 4b for more examples.

ple space is a set consists of outcomes, which are your direct observations in some sense. Then, you can use events [25] to group your outcomes. Since each outcome assocaites with a probablity, the probability of an event is just the summation of the probabilities of the outcomes it's made up from. Wait... how does that relate to counting? Luckily, when your outcomes have the same probabilities, you can use counting to find out the probability of an event [26].

We then learned about conditional probability. That is, what if you restrict the sample space by saying B happens? Then, you can use $\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}$. Confused about the difference between $P(A|B)$ and $P(A \cap B)$? $P(A|B)$ is saying "I already know B happens; what's the probability of A happening? And $P(A \cap B)$ is saying "I still know nothing; what's the probability of A and B happening at the same time?

Let's see why conditional probability is helpful. Sometimes, you want to compute $\Pr(B|A)$ but you know $\Pr(A|B)$, $\Pr(A|\neg B)$ and $\Pr(B)$. How do you "filp" the condition? First, you might want to use Bayne's Rule: $\Pr(B|A) = \frac{\Pr(A \cap B)}{\Pr(A)}$. Okay, $\Pr(A \cap B)$ can be calculated by $\Pr(A|B) \cdot \Pr(B)$, but what about $\Pr(A)$? We have some information about B so let's introduce it to our calculation. You can partition A by checking whether B happens: $\Pr(A) = \Pr(A \cap B) + \Pr(A \cap \neg B)$. Okay, we know $\Pr(A \cap B)$, but what about $\Pr(A \cap \neg B)$? Let's try something similar: $\Pr(A \cap \neg B) = \Pr(A|\neg B) \Pr(\neg B)$ [27]. And nice! We are done.

Some events do not depend on each other. That means, knowing one event happens doesn't affect the probability of the other event. [28] How do you check? Compute the probablity of each and check whether the product is the probability of the intersection. [29] What's nice about them? If you want to calculate probabiliy of an *intersection* of independent events, [30] you can just calculate the probability of each and then multiply them. By the way, a nice case for calculating the *union* of events if when the events are disjoint [31]: the probability of the union then would just be the sum of probability of individual events.

We notice that there are similarities between some events and their probablities, so we decide to use *random variables* and *distributions* to capture these connections. We started with learning about three famous random variables that take on discrete values.

The first is *Binomial random variable*. It denotes the number of success in a fixed number of independent trials (each trial has the same probability of success). What's the distribution of $X \sim \text{Binomial}(n, p)$? That is, what is $\Pr(X = x)$? There are $\binom{n}{x}$ configurations to get x success in n trials, and the

[25] An event is a subset of the sample space.

[26] If outcomes have the same probability, then $\Pr(A) = \frac{|A|}{|\omega|}$

[27] $\Pr(\neg B) = 1 - \Pr(B)$

[28] A and B are independent $\iff P(A|B) = P(A)$

[29] A and B are independent $\iff P(AB))$

[30] P(intersection): If not independent, use Product Rule.

[31] P(union): If not disjoint, use Inclusion-Exclusion Principle

probability for success for each configuration is $p^x(1-p)^{n-x}$ [32].

The second is *Geometric random variable*. It denotes the number of independent trials until we get the first success (each trial has the same probability of success). What's the distribution of $X \sim \text{Geometric}(p)$? That's the probability of the last trial is success and the previous trials are all failures [33].

The last one is *Poission random variable*. It denotes the number of occurance in a fixed time period, given the rate of occurance in that fixed time period. Want an intuitive way of seeing the weird PMF of Poisson distribution [34]? If you plot the PMF of a binomial RV with large n and small p, it approximately overlaps with the PMF of Poisson(np).

Indeed, distribution captures *everything* about a random variable. But sometimes, we want a "summary" of the random variable. You learned about expectation [35] (which tells you the weighted average), and variance [36] (which tells you the deviation). You also learned about how to *calculate* variance: a standard method is using expectation: $Var(X) = E[X^2] - (E[X])^2$. Okay, then, how do we calculate expectation? Thanks to linearity of expectation, you can break up your random variable into smaller ones [37], calculate the expectation of the smaller ones and then add them up. What if your smaller random variables are not independent? Doesn't matter - that's one of the beauty of linearity of expectation [38].

We also learned about the interaction between two (or more) random variables, and we use joint distribution to describe the interaction. Joint distribution captures everything about these two random variables! Using the joint distribution, you can recover the distribution of both random variable [39], or analyze the distribution of one RV conditioning on the value of the other RV [40]. In general, knowing marginal distributions isn't enough to determine the joint distribution; but if the random variables are independent, knowing marginal distributions is enough! [41]

PMF essentially captures *all* information about a random variable. What if we don't want such details? What if we only want the "weighted average"? That's *expectation*. What if we only want the "deviation"? That's *variance*. Similarly, joint PMF captures *all* information about two random variables. What if we only want a measure of linear relationship? That's *covariance*.

You learned about some different strategies to calculate expectation, in case

---

[32] $X \sim \text{Binomial}(n,p) \iff \Pr(X=x) = \binom{n}{x} p^x (1-p)^{n-x}$

[33] $X \sim \text{Geometric}(p) \iff Pr(X=x) = (1-p)^{x-1}p$

[34] $X \sim \text{Poisson}(\lambda) \iff \Pr(X=x) = e^{-\lambda}\frac{\lambda^x}{x!}$

[35] $E[X] = \sum_{\forall x} x \Pr(X=x)$

[36] $Var(X) = E[X - E[X]]$

[37] Common strategy: break up RV into a sum of indicators or a sum of geometric random variables

[38] $E(aX+Y) = aE(X) + E(Y)$. Independence is not required.

[39] Marginal distributions:
$\Pr(X=x) = \sum_{\forall y} \Pr(X=x, Y=y); \Pr(Y=y) = \sum_{\forall x} \Pr(X=x, Y=y)$

[40] Conditional distributions:
$\Pr(X=x|Y=y) = \frac{\Pr(X=x,Y=y)}{\Pr(Y=Y)}; \Pr(Y=y|X=x) = \frac{Pr(Y=y,X=x)}{X=x}$

[41] X, Y independent $\iff \Pr(X=x, Y=y) = \Pr(X=x)\Pr(Y=y), \forall x, y$

the expectation is hard to calculate directly from the formula. You can break up your random variable into the sum of smaller ones [42] and then use linearity of expectation. If the random variable takes on values $\{0, 1, \cdots\}$ and you know how to calculate the tail probabilities, then you can use sum up the tail probabilities to get the expectation by *tail sum formula*. Sometimes, you might think "it would've been so easy to calculate $E(X)$, if I know the value of Y", then maybe try taking the weighted average of conditional expectation.

We moved on the continuous random variables. Since the probability of them taking on an exact value is zero, we define *probability density function* instead of PMF for them. How do you get the probability of a continuous RV? Probability is embedded as the area under PDF. You can transfer most of the concepts in discrete RV to continuous RV: change $\sum$ to $\int$, and change $p_X(x)$ to $f_X(x)dx$.

You learned about two famous continuous random variables. The first is *Exponential Random Variables*. It's used to model the length of time for an event to happen, and it's a continuous analogue of geometric random variables. Like geometric random variables, they have memoryless property and their tail probabilities are pretty easy to calculate. The second is *Normal Random Variables*. They are random variables whose distribution follows a bell curve. Unfortunately, their CDF doesn't have a closed form. However, we have a nice table for the CDF of standard normal random variables! You can transform any normal random variable to the standard one, and then use that table. Do you want another motivation on learning about normal distributions? From WLLN, we know that sample average converges to population mean. CLT tells us something stronger: if you have a lot of i.i.d. random variables, no matter what their distribution is, the sum of them converges to normal distribution.

Sometimes, you only want an upper bound for your probablity. If you know the expectation, Markov Inequality gives you an upper bound for the tail probability. [43] If you know the variance, Chebychev's inequality gives you an (generally better) upper bound for the probability of extreme values [44].

So far, we've mostly only investigated about the distribution of a finite number of random variables. We're closing this course by exploring about *Markov Chain*: a sequence of random variables, each of which only depends on the one before it. Fortunately, we don't need to explicitly give you the PMF for all of them. If someone tell you what the transition probability and the initial distribution, you will be able to tell them the distribution of any of the random variable in the sequence.

You learned about a special distribution (called the invariant distribution). Once you reach this distribution, your markov chain "stops changing". You were also introduced the terms irreducible ("you can go to anywhere, from anywhere") and aperiodic ("all states have period 1"). Why do we care about these properties? It turns out that although all markov chains have at least one invariant distribution, irreducible markov chain only has *one*. If your markov chain

[42] For example, indicators or geometric random variables (Coupon Collector's Problem)
[43] Markov Inequality requires nonnegative random variables.
[44] Chebychev's inequality can be applied to any random variable.

also happens to be aperiodic, then you would get something even stronger: no matter what your initial distribution is, the distribution would always converge to π.

Look at everything you've learned! I'm very proud of you all. I hope you enjoyed this class as much as I did. Eat well; take good rest; and good luck on your final!